



Co-funded by
the European Union



ECCC
EUROPEAN CYBERSECURITY
COMPETENCE CENTRE

DFIR Guide



SOCCER

**WE PLAY FOR CYBER
IN ACADEMIC SECTOR**

The project is co-financed by the Polish Ministry of Science and Higher Education
under program *Co-financed International Projects*.

This publication was supported by the SOCCER project, funded under Grant
Agreement No. 101128073, with the support of the European Cybersecurity
Competence Center (ECCC).

COORDINATOR



AGH UNIVERSITY
OF KRAKOW

CONSORTIUM MEMBERS



JAGIELLONIAN UNIVERSITY
IN KRAKÓW



UNIVERSITY OF AGRICULTURE
IN KRAKOW

MUNI



cesnet
.....



Tomas Bata University



UNIVERSITY
OF TARTU



Mykolas Romeris
University

List of authors

DFIR Guide	Teamwork of the SOCCER Consortium Partners, editing by Pavol Sokol
AGH University of Krakow	Marcin Kopeć, Łukasz Faber, Paweł Kopeć
Masaryk University	Milan Čermák, Daniela Belajová, Rudolf Lukáč, Martin Laštovička
University of Tartu	Pelle Jakovits, Tommy Tomson, Gular Samadova
Pavol Jozef Šafárik University in Košice	Pavol Sokol, Tomáš Bajtoš, Monika Rapavá, Zuzana Hannelová, Henrieta Paločková, Michal Šafranko, Filip Dvorský, Jakub Mohler
Tomas Bata University in Zlín	David Malaník, Nicol Daňková, Ladislav Dorotík, Petr Žáček, Lukáš Králík

SOC CER Project

2025

Table of contents

CHAPTER 1 INCIDENT HANDLING AND RESPONSE

1.1.	Definition of basic terms and abbreviations	9
1.2.	Security of information processing	11
1.3.	Definition and classification of security incidents	12
1.4.	Security incident resolution phases	15
1.5.	Specific provisions concerning the protection of personal data	25
1.6.	Incident report template	26
1.7.	Manual for incident report template	30

CHAPTER 2 SPECIFIC PROCEDURES FOR SECURITY INCIDENTS

2.1.	Specific procedure – Data breach	36
2.2.	Specific procedure – Malware	40
2.3.	Specific procedure – Ransomware	57
2.4.	Specific procedure – Social engineering	60
2.5.	Specific procedure – Linux server compromise	72

CHAPTER 3 DIGITAL FORENSICS

3.1.	Initial phases of digital forensics	80
3.2.	Identification of digital evidence source	82
3.3.	Digital evidence acquisition	85
3.4.	Digital evidence acquisition – Virtual machines	89
3.5.	Digital evidence acquisition – KAPE tool	91
3.6.	Collaboration with Law Enforcement Agencies	102
3.7.	Digital forensic report	103

CHAPTER 4 LIVE FORENSICS

4.1.	Live forensics – Windows operating system	107
4.2.	Live forensics – Linux operating system	110
4.3	Live forensics – MacOS operating system	113

CHAPTER 5 DIGITAL ARTIFACTS ANALYSIS

5.1.	Analysis of Windows system artifacts – System artifacts	119
5.2.	Analysis of Windows system artifacts – Process executions	126
5.3.	Analysis of Windows system artifacts – File-related artifacts	137
5.4.	Analysis of Windows system artifacts – Web browser artifact analysis	144
5.5.	Analysis of Linux system artifacts	148
5.6.	Analysis of MacOS system artifacts	173
5.7.	Analysis of mobile artifacts	179
5.8.	Memory artifacts analysis	186
5.9	Network forensic artifact analysis	196
5.10	Digital Forensic and incident response on Microsoft Entra ID	222

Digital forensics and incident response (DFIR) guides, methodologies and best practices

Introduction

This material serves as a supplement to the document referred to as the Toolbox. The Toolbox provides fundamental guidelines for handling security incidents and digital forensic analysis. This document focuses on specific aspects of security incident responses, offering a detailed overview of methodologies and specific procedures.

The document begins by outlining the process of handling security incidents, including definitions, incident classification, resolution phases, and relevant legislative aspects. Special attention is given to incident reporting, escalation, and post-incident activities, including report preparation.

Next, the document provides specific procedures for handling different security incidents, such as data breaches, malware, ransomware, social engineering, and Linux server compromises. Each scenario includes recommendations for detection, isolation, resolution, and documentation of incidents.

A significant portion of the material is dedicated to digital forensic analysis, including the identification, acquisition, and preservation of digital evidence. The document provides a detailed examination of digital evidence sources, such as computers, storage devices, network infrastructures, and virtual environments.

The document also covers forensic tools, describing reliable software solutions for acquiring digital evidence, including KAPE and other specialized forensic tools. A dedicated section covers collaboration with law enforcement agencies, outlining the request handling process and legal considerations.

Another key topic discussed is live forensic analysis, which examines the acquisition of evidence from Windows, Linux, and macOS systems in real-time. This section offers a comprehensive overview of methods for extracting data from memory, running processes, and network communication.

Finally, the document focuses on analyzing digital artifacts, which is crucial for understanding system activities. It examines artifacts from Windows, Linux, and macOS operating systems, along with mobile devices, volatile memory, and network traffic.

The document offers practical guidelines, methodologies, and best practices for investigating security incidents and conducting digital forensic analysis. Developed as part of the SOCCER project, it aims to support security analysts in academic institutions in effectively managing security incidents and properly handling digital evidence.

1. Incident handling and response

This material provides guidance on how to approach the process of managing and responding to security incidents within the security teams of academic institutions. It serves as a guide for developing an internal governing policy for security incident response within an academic organization.

The guidance material has been developed as a draft security incident handling procedure in accordance with NIST SP 800-61 Rev. 2, Computer Security Incident Handling Guide¹, while considering the European legal framework, particularly the NIS Directive², the NIS2 Directive³, and the General Data Protection Regulation (hereinafter referred to as "GDPR")⁴.

The content of the methodological material is a general framework for dealing with security incidents, including individual phases and steps. The material does not address specific procedures for dealing with specific types of security incidents. These partial procedures (e.g. for handling malware, social engineering) should be developed independently by the academic organization.

1.1. Definition of basic terms and abbreviations

This methodological material will use the abbreviations and definitions of terms related to information security, cybersecurity, and personal data protection, as outlined below. It is recommended that these abbreviations and definitions be included in the internal regulations of the academic organization as part of the overall security incident handling procedure.

Meaning of abbreviations:

- **CSIRT** – Computer security incident response team,
- **SOC** – security operation centre,
- **DPO** – data protection officer
- **NIST** – National Institute of Standards and Technology,
- **GDPR** – General Data Protection Regulation,
- **NIS** – Network and Information Systems Directive,
- **NIS2** – Second Network and Information Systems Directive,
- **PII** – Personally Identifiable Information,
- **DoS** – Denial of Service,
- **DDoS** – Distributed Denial of Service,
- **APT** – Advanced Persistent Threat,
- **TLP** – Traffic Light Protocol,
- **ISO** – International Organization for Standardization,
- **IEC** – International Electrotechnical Commission,
- **ENISA** – European Union Agency for Cybersecurity,
- **EUROPOL** – European Union Agency for Law Enforcement Cooperation,
- **LE** – Law Enforcement.

Definitions:

- **Asset** is anything of value to the academic organization.
- **Attack** is a deliberate attempt to exploit a vulnerability in a system/asset in order to gain unauthorized privileges or to damage/destroy that asset, or any other asset within the system or organization.

- **Availability** is the ability to be accessible and usable upon request by an authorized entity.
- **Breach** is the loss of control, compromise, unauthorized disclosure, unauthorized acquisition, or any similar event in which someone other than an authorized user accesses, or potentially accesses, personally identifiable information, or an authorized user accesses personally identifiable information for an unauthorized purpose.
- **Compromising** is the loss of trust, exposure to threats, or the potential for suspicion.
- **Computer network** is an electronic communication network in which data is transmitted by means of fixed or wireless communication links.
- **Confidentiality** is the property that ensures information is not disclosed or made accessible to unauthorized individuals.
- **Data subject** is any natural person whose personal data is processed.
- **Denial of service** is the disruption of an information system's operation by depleting its resources, e.g. running out of memory capacity or overwhelming the information system.
- **Digital evidence** is any information of evidentiary value stored or transmitted in digital form.
- **Event** is any observable change in the behavior of a system, environment, process, or individual.
- **Impact** is the negative consequence(s) of an incident on the academic organization, its employees, and students, e.g. data loss, data corruption, data misuse, data leakage, disruption of information system services, internet connection failure, misuse of system resources for criminal activities, damage to the organization's reputation, violation of legal regulations, threats to citizens' rights, etc.
- **Incident reporter** is any external or internal entity that reports a cybersecurity incident.
- **Information asset** is anything of value to an organization, including, but not limited to, another organization, individual, computing device, information technology (IT) system, IT network, IT circuit, software (both installed and physical instances), virtual computing platforms (commonly found in cloud and virtualized computing), and related hardware (e.g. locks, cabinets, keyboards).
- **Information system** is an application, service, technical device, or any other element that processes information.
- **Information system of a security team** is a ticketing and support system designed to record, process, and manage information related to incidents, incident reports, resolved cases, and other incident-related information.
- **Integrity** is the property of ensuring the accuracy and completeness of assets.
- **Personally Identifiable Information (PII)** is any information that relates to an identified or identifiable natural person.
- **Risk** is the likelihood that a vulnerability in the system will affect the authentication, availability, authenticity, integrity, or confidentiality of the data processed or transmitted, along with the severity of the consequences resulting from the intentional or unintentional exploitation of such a vulnerability.
- **Security event** is an identified occurrence related to the state of an information asset (e.g. unsuccessful user login) that may indicate a potential breach of information security policy, a failure of safeguards, or a previously unknown situation that could be security-relevant. Not all security events are classified as information security incidents.
- **Security incident** is a security event, or series of unexpected or unwanted events, that has been determined to impact the organization, prompting a need for response and recovery. This is due to a significant likelihood of compromising business operations and threatening information security.

- **Security incident management** is a set of processes for detecting, reporting, assessing, responding to, handling, and learning from information security incidents.
- **Security incident response** includes actions taken to analyze, detect, defend against, and respond to an information security incident, including measures to protect and restore the normal operational conditions of an information system and the information it stores.
- **Terminal equipment** is an electronic communication device or a technical component that enables communication and is designed for direct or indirect connection to the connection points of a computer network.
- **Threat** is an objectively existing potential capability to directly or indirectly compromise a system, the information it processes, or other assets of an organization.
- **Vulnerability** is a property, condition, or circumstance that allows a specific threat to be realized. For example, connecting an unprotected computer to the Internet creates a vulnerability that enables a hacker attack, while an outdated virus database is a vulnerability that allows a computer to be infected with malicious software.

1.2. Security of information processing

Information sharing is a crucial activity when managing security incidents. The Traffic Light Protocol, currently in version 2.0 (hereafter referred to as “TLP”), is used within the security community to enable the sharing of potentially sensitive information and promote more effective collaboration. Information is shared from an information source to one or more recipients. TLP is a set of four labels (red, amber, green, clear) used to indicate sharing boundaries that should be followed by recipients.⁵

Other approaches to information processing security may also apply within an academic organization, particularly the implementation of data protection legislation. This legislation mandates the security of personal data processing, which may be part of the information exchanged.

Another example is the management of information security within an organization. According to ISO 27001:2022 Annex A 5.1.2, information must be classified based on various factors, including legal requirements, value, criticality, and sensitivity to unauthorized disclosure or modification⁶. This classification should be designed to reflect the organization's unique business activities, ensuring it does not hinder or complicate them.

Draft provisions for the internal regulations of an academic organization

The internal regulations of an academic organization, particularly those governing the protection of personal data, shall apply to any information obtained during the handling of a security incident.

Individuals involved in the resolution of a security incident are required to follow the Traffic Light Protocol, version 2.0 (hereinafter referred to as “TLP”), when communicating and exchanging information related to the security incident.

The TLP uses four colors to indicate different levels of sensitivity of information and the appropriate methods for sharing that information, namely:

- **TLP: RED** – non-disclosable information that may only be accessed by authorized individuals and cannot be redistributed. It is typically used for information that would pose a significant risk to the confidentiality, reputation, and operations of the organizations involved. This information must not be discussed in the presence of third parties.

- **TLP: AMBER** – information that may only be disclosed to those directly involved in the exchange of information, and to individuals within the organization and its structure on a 'need to know' basis.
- **TLP: AMBER+STRICT** – used to indicate information that can only be shared within the recipient organization. This designation applies to sensitive information that must be shared effectively among authorized recipients, but where there is a risk of compromising privacy, reputation, or operations if it is leaked.
- **TLP: GREEN** – information that can be shared with other organizations or individuals within the broader community or sector, but should not be disseminated through publicly accessible communication channels, such as websites. Unless otherwise specified, it is assumed that the information can be shared within the information and cybersecurity community.
- **TLP: CLEAR** – information intended for public dissemination, with no restrictions on its distribution. However, copyright must be respected in its use.

1.3. Definition and classification of security incidents

To address security incidents within any organization, including academic organizations, it is essential to define what constitutes a **security incident**. There are various definitions within the field of information security. ISO 27001:2022 defines a security incident as *a single or a series of unwanted or unexpected information security events that have a significant probability of compromising business operations and threatening information security*.⁷

The definition of a security incident can also be found in relevant legislation. According to Article 6(6) of the NIS2 Directive, an incident is defined as *an event compromising the availability, authenticity, integrity or confidentiality of stored, transmitted or processed data or of the services offered by, or accessible via, network and information systems*.

Academic organizations should specify the scope and types of security incidents. In particular, it is important to determine the scope of incidents that the organization aims to address, or to ensure that the legal framework clearly outlines the types of security incidents that the organization is obliged to address. An example that is not entirely clear is copyright infringement. While such infringements are indeed violations and the academic organization is obliged to address them, it is questionable whether the security team should handle them as part of their activities.

In practice, however, we do not solely rely on the concept of a security incident. It is crucial to distinguish between different types of security incidents. This enables those handling the incident to choose the most appropriate course of action. Otherwise, one might mistakenly treat a fraudulent email message containing a malicious link the same way as a malware infection that has encrypted a device and is spreading within the organization's computer network. This distinction between different types of security incidents is referred to as a **security incident taxonomy**.

Currently, there is no uniform approach to classifying individual security incidents, and it is not possible to determine a universally 'best' or 'correct' classification. Organizations typically create or adopt a taxonomy of security incidents based on their specific needs and expectations⁸. Since different organizations have varying expectations, they often develop their own internal security incident classification. On the other hand, there are several initiatives that attempt to unify these taxonomies. This unification is especially

important due to the need for mutual reporting of security incidents between organizations, including the exchange of information regarding these incidents. If collaborating organizations use different taxonomies, they might be addressing the same security incident but labelling it differently. For example, an attack containing malware designed to download another malware (Trojan) might be classified as a malware-Trojan in one taxonomy, and as a malware-infection in another. Currently, the following taxonomies of security incidents are most commonly used:

- eCSIRT.net mkIV⁹,
- CIRCL.LU taxonomy¹⁰,
- Common Taxonomy for Law Enforcement and CSIRTs (Common Taxonomy for LE and CSIRTs)¹¹.

Within this document, the classification of security incidents is based on the Incident Classification / Incident Taxonomy according to CSIRT.net, with adaptations.

At the same time, it is important to consider the classification of security incidents based on the scope and severity of their impact. This approach can significantly assist in the triage of security incidents, helping the security team of an academic organization to determine whether it has sufficient staff and technical resources to address multiple security incidents simultaneously.

Draft provisions for the internal regulations of an academic organization

We divide security incidents into nine main categories:

- **Unwanted content** – especially spam and harassment, this category also encompasses emails or websites containing discriminatory or discrediting content, pornography, promoting violence, etc.
- **Malicious code** – software intentionally inserted into a system with harmful intent. This includes Trojan horses, viruses, worms, spyware, ransomware, etc.
- **Information gathering** – including activities such as scanning, eavesdropping, and social engineering.
- **Penetration attempts** – specifically attempts to compromise systems and services by exploiting vulnerabilities, repeated unsuccessful login attempts, and attempts to penetrate in previously unknown ways.
- **Penetration** – specifically the successful compromise of a system, application, or user accounts. It can occur remotely or through unauthorized local access. Involvement in a botnet also falls under this category.
- **Unavailability** – primarily attacks aimed at disrupting the availability of systems (DoS and DDoS), as well as disruptions caused by local activities such as power interruptions (sabotage).
- **Threats to information security** – specifically the misuse of information and systems, or attacks that intercept transmitted information. For example, a successful compromise can also endanger information security.
- **Fraud** – specifically the sale and installation of unlicensed software and other copyrighted material, identity spoofing attacks, phishing, or the misuse of resources for unauthorized purposes.
- **Other** – security incidents that do not fit into any of the previous categories.

Security incidents are classified based on the scope and severity of impact as follows:

- **Low severity** – minimal impact on the operations of the academic organization, particularly regarding the confidentiality, integrity, and availability of its information systems and the data of the academic organization and its users,
- **Moderate severity** – moderate impact on the operations of the academic organization, especially on the confidentiality, integrity, and availability of its information systems and the data of the academic organization and its users

- (threatening a small part of the academic organization or the provision of support services),
- **High severity** – negative impact on the operations of the academic organization, particularly on the confidentiality, integrity, and the availability of its information systems and the data of the academic organization and its users (threatening a significant part of the academic organization or the provision of essential services),
 - **Critical severity** – significant negative impact on the operations of the academic organization, particularly on the confidentiality, integrity, and availability of its information systems and the data of the academic organization and its users (threatening the entire academic organization or the provision of essential services).

Table 1 – Example of security incident severity classification

Severity	Definition	Categories
Critical	An incident affecting critical systems, information, or services with a severe impact on the organization's operations, integrity, availability, or confidentiality.	• Ransomware affecting critical infrastructure
	• Denial of Service (DoS/DDoS) attack disrupting essential services	• Penetration (successful system compromise, botnet involvement)
	• Compromised critical assets (e.g. domain controllers, financial systems)	• Advanced Persistent Threat (APT) attacks
	• Active internal hacking (privilege escalation, lateral movement)	• Active external hacking (perimeter defense breach)
	• Malware outbreaks (e.g. destructive worms, ransomware infections)	• Threats to information security (interception, data exfiltration)
High	An incident affecting non-critical yet significant systems or information, causing operational delays or moderate security concerns.	• Unauthorized access to sensitive information
	• Internal hacking attempts (non-active exploitation)	• External hacking attempts (non-active, probing)
	• Major data breach affecting sensitive but non-critical data	• Unlawful activity (fraud, financial crime, unauthorized transactions)
	• Phishing attacks targeting privileged accounts	• Compromised non-critical assets (workstations, user accounts)
	• Penetration attempts (exploitation of vulnerabilities, repeated login failures)	• Information gathering (scanning, eavesdropping, social engineering)
Medium	An incident with potential security risks, policy violations, or requiring forensic investigations, but without immediate operational impact.	• Policy violations (e.g. unauthorized software installations, data misuse)

	Insider threats requiring investigation	Phishing campaigns targeting a wide user base
	System errors or load issues impacting non-critical operations	Unauthorized access attempts requiring monitoring
	Fraud (identity spoofing, unlicensed software use)	Unavailability due to localized issues (power interruptions, sabotage)
Low	Minor security issues or service disruptions with no significant impact on operations.	Spam and minor phishing emails
	Minor misconfigurations with no immediate security implications	Unwanted content (spam, harassment, inappropriate material)
	System issues causing minor inconvenience	Routine security alerts requiring minimal follow-up
	Other (security incidents that do not fall into the previous categories)	

1.4. Security incident resolution phases

According to Article 6(8) of the NIS2 Directive, incident handling refers to any actions and procedures designed to prevent, detect, analyze, contain, respond to, and recover from an incident.

Security incident response is a complex process that should be divided into several phases. Each phase may involve various activities, depending primarily on the type, scope, and severity of the security incident.

This document refers to the manual of the National Institute of Standards and Technology (NIST), *800-61 Revision 2 – The NIST Computer Security Incident Handling Guide*. According to NIST 800-61-r2¹², the handling of a security incident involves the following phases (Figure 1): Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned.



Figure 1 – Phases of security incident resolution according to NIST 800-61 r2.

For the purposes of academic organizations, the solution phases according to NIST 800-61 r2 have been slightly modified and adjusted. A flowchart of the security incident resolution process is shown in Figure 2.

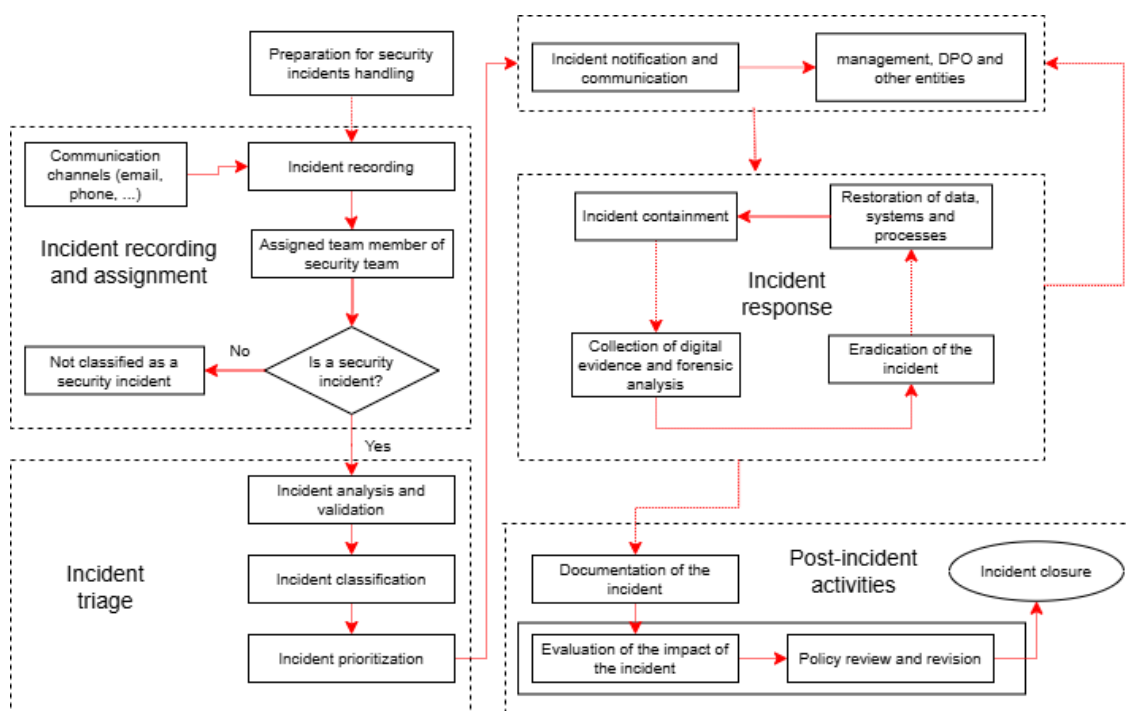


Figure 2 – Proposed phases of security incident resolution

Table 2 shows the difference between the incident resolution phases based on NIST 800-61 r2 and the proposed phases of security incident resolution.

Table 2 – Comparison of the incident resolution phases based on NIST 800-61 r2 and the proposed phases of security incident resolution

NIST 800-61-r2	Proposed phases
Preparation	Preparation
Detection & analysis	Incident recording and assignment
	Incident triage
	Incident notification and communication
Containment, eradication, recovery	Incident response
Post-incident activity	Post-incident activities

Draft provisions for the internal regulations of an academic organization

The process of dealing with security incidents can involve the following phases:

- preparation,
- recording,
- triage,
- notification,

- response,
- post-security incident activities and closure,
- review of working practices and suggestions for improvement.

Each phase may consist of a number of activities, which can vary depending on the type, scope, and severity of the security incident.

In addition to the general process for handling security incidents, the organization may issue a Rector's Guideline that outlines the procedure for specific categories of incidents.

Preparing for security incidents

Preparing for security incidents is crucial for ensuring an adequate response. This phase focuses on establishing the right procedures and tools before a security incident occurs.

Preparing for security incidents involves acquiring adequate technical and other resources, establishing processes, and training the security team of the academic institution. This phase also includes preparing specific steps for each phase, checklists, etc.

The phase of preparing for security incidents includes the following activities and tasks:

1. Developing and updating the incident response policy and plan, which outline the scope, objectives, and approach of the incident response process.
2. Establishing and training the incident response team, which includes the incident response lead, security team members, and the ad hoc personnel as needed.
3. Implementing and testing incident response tools and resources that support the process, such as incident response software and hardware, procedures, checklists, templates, as well as contact lists and agreements.
4. Implementing and testing security controls and measures that protect information systems and data from incidents, such as security policies, standards, and guidelines, security awareness and education programmes, and security monitoring and auditing systems.
5. Defining conditions that serve as guidelines for reporting obligations based on local and EU laws or regulations (e.g. GDPR, NIS2 Directive). These conditions are stored in detailed playbooks.

Draft provisions for the internal regulations of an academic organization

Preparing for security incidents primarily consists of the following points:

1. Establishing workflows for handling specific categories of cybersecurity incidents, which includes defining the incident response policy and plans that outline the scope, objectives, and approach to the incident response process.
2. Verifying, practising, and updating work procedures through regular testing and implementation of necessary security measures to protect information systems and data (e.g. security policies, standards, guidelines, audits, and security monitoring).
3. Training all relevant personnel, including the incident response team (comprising the incident response lead, members of the security team, and ad hoc members), with a focus on effectively handling cybersecurity incidents.
4. Establishing and maintaining the technical security (hardware and software) necessary to handle various categories of cybersecurity incidents, including the implementation and testing of incident response tools and resources (e.g. software, hardware, checklists, templates, and contact lists).
5. Defining conditions for reporting obligations in compliance with local and European legal regulations (e.g. GDPR, NIS2), ensuring that these conditions are thoroughly documented in playbooks.

Initiation of resolution and recording of security incidents

This phase is dedicated to initiating the resolution of a security incident. It begins when the security team receives information about a potential security incident, either from their own activities (e.g. network traffic monitoring) or through a security incident report.

Reporting a security incident is typically done via email, a form, or a phone call. For CSIRT teams, at a minimum, a report must be received via email, and in emergency cases, a phone call is required. If the report is accepted by phone, a member of the security team must conduct an interview, asking a set of predefined questions. An example of these questions can be found on the Slovak Government CSIRT.SK website.¹³

The form is recommended for use if the academic institution's security team utilizes an automated tool to process reports. However, it should be considered that the reporter of a security incident may not have technical expertise. In some cases, they may be discouraged from reporting a security incident because of a complicated way of reporting or the requirement to fill in extensive data that they may not have the answer to. Therefore, it is recommended to simplify the process initially and, at this and the subsequent stages, request additional information about the security incident from the reporter or other sources.

When a security incident is reported, a request (ticket) is created and assigned to a member of the academic institution's security team. The assigned incident handler will investigate the issue and, if it qualifies as a security incident, they will proceed with the necessary actions. If the issue does not qualify as a security incident, the request will be closed. If a technical incident is identified, the request will be escalated to local technical support.

It is essential for the internal regulations of an academic institution to mandate that the users of its IT infrastructure and information systems report security incidents. Additionally, each user should be adequately informed on the proper procedures for reporting security incidents.

Draft provisions for the internal regulations of an academic organization

Security incidents can be initiated based on:

- end-user reporting,
- proactive identification by the academic organization's security team, DPO, or local IT administrator.

Security incidents can be reported via:

- the email address incident@xxx.yz
- a telephone call to the number XXX.

Every employee, student, and guest of an academic organization is required to report any incidents or suspected security incidents, including potential data breaches, through any of the methods mentioned above.

When an incident is reported, it is assessed to determine whether it constitutes a security incident. If confirmed, including personal data breaches, it will be handled according to these Operating Regulations. If identified as a technical issue, the incident will be escalated to the appropriate IT administrator through the academic organization's helpdesk system. Otherwise, the entire process will be stopped.

Incident reports and security incidents are logged in the academic organization's internal security team ticketing system. This system should record all communications with the security team. Security incidents involving personal breaches are handled by the DPO (as a data breach) and the security team (as a security incident).

The security incident is assigned to the appropriate member of the academic organization's security team based on the department's work schedule (hereafter referred to as the 'security incident responder'). The internal security team ticketing system should log all communications with the security team. Incident resolution responses should preferably be sent from the internal security team ticketing system rather than from the email addresses of the security incident responders. This ensures that all communication is properly recorded, which is also crucial in case of substitutability within the security team.

In particular, the following information should be collected when initiating the resolution of a security incident:

- the estimated start time of the security incident,
- the time and manner in which the security incident was detected,
- a description of the incident's course and scope,
- a description of any countermeasures taken,
- the name and contact details of a person who can provide further information,
- any other relevant information.

In the event of a security incident where a personal data breach is believed to have occurred, the following information shall be collected in addition to the information listed above:

- a description of the categories of data subjects affected by the breach,
- the approximate number of data subjects concerned by the breach,
- a description of the categories of records affected by the breach.

Where it is not possible to provide all the information related to a security incident at once, the information may be provided in stages, without undue delay.

Security incident triage

The triage of a security incident involves further analysis and validation of the incident. During this phase, the category and priority of the security incident is determined. The incident handler analyses the security incident and examines details such as the type of attack, the severity of the incident, the intended target of the attack, impact on the organization, method of dissemination, and potential vulnerabilities that have been or could be exploited. This information will help the academic organization's security team in the subsequent stages of resolving the security incident.

Draft provisions for the internal regulations of an academic organization

The purpose of the security incident triage is to determine the incident's category and severity rating.

As a rule, the incident handler within the triage performs the following steps:

1. Add information about the security incident,
2. Assess the scope and severity of the security incident based on available information,
3. Classify the security incident by type into one of the predefined categories,
4. Evaluate whether the incident falls into a known category with an existing workflow,
5. Prioritize security incidents based on their category, severity, and the academic organization's available resources for resolution.

Security incident notification

In the security incident notification phase, relevant information about the security incident is communicated to various stakeholders, including the academic organization's management, third-party vendors, and other entities (e.g. law enforcement and state cybersecurity and privacy organizations).

For security incidents with high or critical severity, the incident handler must report them to the organization's management immediately after confirmation and verification. For other incidents, it is not necessary to inform the organization's management immediately. Certain authorizations should be explicitly defined in the academic organization's internal regulations to ensure a timely response to urgent activities.

Draft provisions for the internal regulations of an academic organization

The handled security incident is reported to:

- the Director of the Security Team for incidents of at least medium severity,
- the management of the academic organization for incidents of at least high severity,
- the academic organization's crisis management team for incidents of at least high severity (if the academic organization has a crisis management team),
- the PR department of an academic organization if it is likely to be necessary to communicate information towards the public and the media.

If a security incident results in a personal data breach, the DPO must be notified without delay. The incident handler and other involved personnel shall provide the necessary assistance to that person. This may also include notifying the affected data subjects, as required by the GDPR.

If it is likely that the public, a large number of users (especially employees and students) of the academic organization, or the media, which become interested in the incident, need to be informed about the security incident and its resolution, the spokesperson of the academic organization's PR department shall also be informed to the extent necessary.

If legal issues are likely to arise during the resolution of a security incident, the incident handler shall contact the legal department of the academic organization.

The handling of a major security incident shall be coordinated by the crisis management team. Information about a serious security incident and its resolution shall be prepared by the crisis management team and communicated through available communication channels.

The organizational components of the academic organization shall follow the crisis management guidelines at each step of the incident response, including communication with staff, students, the public, and the media.

An academic institution is responsible, under national cybersecurity legislation (following the implementation of the NIS 2 directive), for reporting serious cybersecurity incidents. Serious cybersecurity incidents are those considered to have high or critical severity (Chapter 2).

Reporting a serious cybersecurity incident is carried out through the procedure outlined in the legal regulation (e.g. via email or an information system):

- without undue delay, and no later than 24 hours after its detection, an early warning is reported, specifying whether the serious cybersecurity incident could

have been caused by unlawful conduct or whether it may have a cross-border impact,

- without undue delay, and no later than 72 hours after its detection, a notification of a serious cybersecurity incident is reported. This notification updates and supplements the information from the early warning, particularly providing the initial assessment of the cybersecurity incident, its severity, and consequences. In the case of an operator of a basic service who is a provider of trusted services, the notification must be submitted without undue delay, but no later than 24 hours after its detection,
- no later than one month after reporting the notification pursuant to the previous point, a final report is submitted. This report contains, in particular, a detailed description of the serious cybersecurity incident, including its severity and consequences, the type of cyber threat or root cause of the cybersecurity incident, the measures in place and ongoing, and the cross-border impact, if applicable.

General and specific procedures for handling security incidents

This guidance material is intended to assist in the development of a generic procedure for handling security incidents within academic organizations. It is highly recommended to develop specific procedures for each type of security incident that the security team encounters on a relatively routine basis. If a specific resolution workflow is developed for the category of security incident being addressed, the incident handler must follow that procedure.

Security Incident Handling Procedures are stored and managed in the academic organization's security team information system. It is advisable to keep one printed copy of this procedure in case the security team's information system is affected by a security incident.

Draft provisions for the internal regulations of an academic organization

If a resolution workflow is developed for the category of security incident being addressed, the incident handler must follow that workflow.

If, during the course of resolving a security incident according to the workflow, the incident handler determines that:

- the security incident has been misclassified,
- the security incident is of high or critical severity,
- the incident handler shall determine the correct category or severity.

If, during the course of resolving a security incident according to the workflow, the incident handler determines that the resolution procedure is inappropriate for the security incident being addressed, they shall inform the head of the academic organization's security team about the incident. The head will then decide on the next course of action.

Security incident response

Security incident response is a time-consuming phase in the incident response process. From a procedural standpoint, it follows the notification of a security incident and typically concludes with the resolution of the incident. Given the importance and scope of this phase, a separate chapter has been dedicated to it.

The specificity of this phase of incident handling lies primarily in its variation depending on the type of security incident. It is important to establish a general framework for the different phases within the general procedure, and to detail the activities and specificities of each phase for a particular type of security incident in the specific

procedures. As shown in the security incident handling flowchart in Fig. 2, before starting the general procedure, security team members are required to prioritize the specific procedure, if such a procedure exists within the organization.

The response to a security incident generally consists of the following activities:

- containment of the security incident,
- digital trace collection and digital forensics,
- eradication of the security incident,
- recovery from the security incident.

It is important that the activities can run in parallel. In some cases, the resolution of a security incident may be divided into several parts, with recovery already underway on one part of the IT infrastructure, while digital traces are still being collected on another.

Containment of Security Incidents

During the containment phase of a security incident, the organization attempts to **mitigate (limit) the impact of the security incident** by stopping its progression and preparing for full recovery. This phase often involves cycling back to detection and analysis. For example, security teams may need to determine whether other devices within the network are infected with malicious code (malware).

For this phase, it is important to define the authority and responsibilities of the entities within the academic organization that will participate in resolving security incidents. A key consideration is determining who has the authority to decide, and for which security incidents, to disconnect a segment of the academic organization's computer network or restrict access to its information system.

Draft provisions for the internal regulations of an academic organization

Containment of a security incident is a necessary security incident response activity that prevents the security incident from spreading and reduces an impact.

Gathering digital evidence and digital forensics

During this phase, digital and other relevant evidence is collected. Subsequently, digital forensic analysis is conducted to identify details of the attack method, exploited security vulnerabilities, and compromised network devices and information systems. This information enables the security team to improve containment measures and eliminate the initial attack vector (e.g. by identifying exploited vulnerabilities).

Digital forensics does not need to be performed for every security incident. Academic organizations should determine when this activity is necessary, considering their internal capabilities. This activity can also be outsourced to an external security team.

Draft provisions for the internal regulations of an academic organization

Digital evidence collection is a necessary security incident response activity in which relevant digital evidence is identified and collected.

During this phase, it is essential to collect digital evidence in a manner that enables the identification of critical information about the security incident or, if necessary, to submit the information to law enforcement authorities.

In cases of digital evidence recovery, the incident resolver must maintain a record of all actions taken with digital traces and evidence media as part of the chain of custody.

In the event of a high-severity security incident, a digital forensic analysis will also be performed. Digital forensics may also be conducted for less serious security incidents if

the security incident responder determines that the information obtained from this analysis could help clarify the details of the incident.

Eradication of a security incident

In the eradication phase of a security incident, the **root cause of the security incident is eliminated** and all attack vectors (the pathways an attacker uses to achieve their goal within the organization) are closed. This is done to prevent similar security incidents from occurring in the future. Remediation methods may include fixing security vulnerabilities (bugs), replacing malfunctioning devices, and implementing enhanced security mechanisms.

Draft provisions for the internal regulations of an academic organization

Security incident eradication is a crucial security incident response activity aimed at identifying the cause, vulnerability, or reason that led to the occurrence of the security incident.

During the eradication phase, it is mandatory to implement appropriate security steps and measures, specifically:

1. Removal of identified security vulnerabilities,
2. Installation of current updates,
3. Reconfiguration of affected services,
4. Deactivation of compromised accounts.

Recovery from a security incident

Once the causes of a security incident have been remediated, the academic organization's IT department is responsible for **recovering** affected systems, services, resources, and data. It is also responsible for minimizing the duration of disruption to the organization's services or operations caused by the incident. For this reason, it is crucial to restore affected devices, applications, and systems as quickly as possible, either through replacement or quick fixes. During this phase, the security team collaborates with the IT department. Depending on the internal structure of the academic organization, this phase may be carried out solely by the IT department. This phase is typically executed in two steps:

- **service recovery** – according to the organization's contingency plans (e.g. restoring data in an academic information system when the hardware supporting the information system is damaged),
- **validation of the organization's equipment, information systems, and computer network** – testing and verifying the functionality of these systems.

Draft provisions for the internal regulations of an academic organization

Security incident recovery is the activity of restoring data, information systems, and processes within an academic organization that have been affected or impacted by a security incident.

Only data and information systems that are unlikely to be further impacted by the continuation of the current security incident or the occurrence of a new security incident may be recovered.

Some phases, such as identification, containment, or eradication, may need to be repeated depending on the nature and complexity of the security incident. For example, if new vulnerabilities are discovered during the eradication phase, the identification and eradication processes must be repeated. Another example involves severe security

incidents that affect multiple organizational units or various systems, where individual steps may need to be performed multiple times to ensure a thorough resolution of the security incident.

Post-security incident activities and security incident closure

Once the security incident response has been completed, the academic organization's security team will finalize the information related to that security incident. In particular, the security team's ticketing system shall be updated with details on how the security incident was resolved, the resulting impact of the security incident (if identifiable), and a brief description of the resolution process. The description of the security incident should include the attack vector or any identified parts of the incident.

The post-incident activity involves the following key activities and tasks:

- **Incident documentation:** Collecting and recording incident-related data and artifacts to ensure the accuracy, completeness, and consistency of incident information. Proper documentation supports incident reporting, analysis, and serves as a reference for future improvement.
- **Incident reporting:** Communicating and presenting incident-related data and findings to ensure timely and accurate information sharing among the incident response team, relevant stakeholders, and local authorities. This also ensures compliance with legal and regulatory obligations (e.g. GDPR, NIS2 directive).
- **Incident analysis:** Reviewing and evaluating incident data and artifacts to determine root causes, lessons learned, and best practices. This process helps assess the performance, effectiveness, and efficiency of the incident response efforts.
- **Incident improvement:** Implementing and monitoring corrective and preventive actions to address identified weaknesses and enhance overall security resilience. This includes refining processes, updating policies, and strengthening the organization's incident response capabilities.

If the security team communicates with other organizational units or institutions, it will provide adequate information to these entities.

For significant security incidents (e.g. security incident of high or critical severity) it is appropriate to produce a security incident report. This report can then serve as a basis for criminal investigations or for addressing data breaches.

Once the resolution of a security incident has been concluded, the adequacy of the applied procedure, tools, individual steps, KPIs, and other factors must be reviewed. It is not necessary to analyze and evaluate security incidents of low and medium severity, or those that are frequently recurring. The aim is to identify those aspects of the security incident response that could not be applied or need to be adjusted based on the lessons learned. The results from this phase provide input for the security incident preparation phase.

Draft provisions for the internal regulations of an academic organization

Security incidents for which a resolution workflow has been developed are closed in accordance with that workflow. Other security incidents shall be closed in the system in which they are recorded.

In particular, the following must be stated when closing a security incident:

- how it was resolved,
- the resulting impact of the security incident,
- a brief description of the handling of the security incident.

In the case of security incidents of high and critical severity, a security incident report shall be produced.

The incident handler is responsible for closing the security incident.

If requested by the reporter, the reporter of the security incident shall be informed once the security incident has been concluded.

All successful and unsuccessful activities that took place during the resolution of a security incident need to be evaluated after the incident has been resolved to assess the adequacy and effectiveness of the procedures used, and to identify new procedures that need to be developed. It is not necessary to analyze and evaluate security incidents of low and medium severity, or those that occur frequently.

1.5. Specific provisions concerning the protection of personal data

Several types of security incidents also qualify as data breaches, particularly when they compromise the confidentiality, integrity, or availability of an academic organization's assets that contain personal data. EU personal data legislation, specifically the GDPR, applies to all EU Member States.

The academic organization is required to document each security incident involving a personal data breach, including the relevant facts, its consequences, and the remediation measures taken.

An important aspect is the need to report security breaches within 72 hours of the academic organization becoming aware of the incident. The report must be submitted to the authority responsible for data protection oversight in the respective country.

Draft provisions for the internal regulations of an academic organization

The DPO of the academic organization or an authorized representative shall, upon written notification, prepare supporting documents and report the personal data breach without undue delay, preferably within 72 hours of becoming aware of it, to the authority responsible for data protection oversight in the respective country, unless the breach is unlikely to pose a risk to the rights and freedoms of natural persons.

If the notification has not been submitted to the Authority within 72 hours, a justification for the delay must accompany it.

The supporting material for the above notification shall include, at a minimum:

- the date and time of the discovery of the security incident involving the personal data breach,
- the date and time when the security incident involving the personal data breach began,
- the date and time when the security incident involving the personal data breach ended,
- a description of the nature of the personal data breach,
- a description of the categories of data subjects affected by the infringement,
- the approximate number of individuals impacted by the infringement,
- a description of the categories of records affected by the breach,
- the approximate number of records affected by the infringement,
- a description of the likely consequences of the personal data breach,
- a description of the measures taken or proposed to address the personal data breach, including, where necessary, any measures to mitigate potential adverse effects.

Where it is not possible to provide the information related to the notification of a personal data breach at the same time, the information may be provided in several stages without further undue delay.

The academic organization's security team, in collaboration with the DPO, is required to document each security incident involving a data breach, including the facts associated with the breach, its consequences, and the remediation measures taken.

1.6. Incident report template

[Team Logo]

[Title]

CSIRT-202x-Incident-001

TLP:AMBER / **TLP:RED**

Title of analysis	Security incident report (name / designation of the incident)
Analysis ID	# CSIRT-202x-Security-001
Incident Type ¹	Incident_type ¹⁴
Incident severity	Low/Medium/High
Prepared by	Department or Faculty / Cyber Security Incident Response Team Address Contact e-mail, website
Version	1
Date of preparation	01.01.202x
TLP	TLP:AMBER / TLP:RED

Managerial summary

Security Incident [name / designation of incident] occurred ...

[detailed description]

This report includes a list of Indicators of Compromise (Appendix A) and several security measures that, as part of proactive organizational activities, could help prevent similar security incidents or mitigate their impact on the organization.

Incident description

A detailed view of how the incident occurred, when it happened, and the resulting consequences.

Security incident response

As part of the security incident response, an investigation was conducted on the compromised device. This included inspecting the device, identifying the method of compromise, as well as analyzing the attacker's actions. Further information on the security incident response is provided in the following sections:

- security incident resolution timeline,
- methodology,
- limitations,
- analysis of digital traces and other information.

Timeline of the security incident

Several key events were identified as part of the security incident response. This section (Table 3) provides a timeline of the security incident, detailing the activities of the entities involved in its resolution. Relevant email communications can be found in Appendix B.

Table 3 – Timeline of the security incident response

Timestamp (CEST)	Event
01.01.2024 at 00:00	Description of event

Methodology

The procedure for handling a security incident follows the guidelines outlined in **NIST** Special Publication 800-61 Revision 2 – Computer Security Incident Handling Guide. The procedure will be adjusted based on the CSIRT's level of involvement in the security incident response. Incident handling includes conducting basic digital forensics, open-source intelligence (OSINT) analysis, and the analysis of malicious code behavior (if applicable). For OSINT, tools such as SHODAN, OTX Endpoint Security, AbuseIPDB, and other publicly available services (e.g. GitHub) were used. The course of the digital forensic analysis depends on the method (postmortem or live analysis), the operating system, as well as the extent of the compromise (application, server, etc.)

Tools commonly included in the Linux/Windows operating systems were used to analyze the compromised device. Specifically, operating system logs, application logs, file system logs, processes, network connections, and other relevant forensic artifacts (such as scheduled tasks) were examined.

Limitations

No significant constraints were identified in handling this security incident that would hinder the security incident response or digital forensic investigation. Similarly, no extensive evasion techniques by the attacker (e.g. deletion of application logs) were observed that would have complicated the analysis, such as unsecured operational memory or deleted logs.

Analysis of digital evidence and other information

OSINT (open source) analysis

Basic information about the analyzed device

File system analysis

Logs

Lateral movement

Privilege escalation and persistence

Notes and other relevant information

No other relevant information was identified during the resolution of the security incident that would impact the organization's security.

Conclusion

As part of the resolution of the security incident, the necessary actions were taken to halt its progression, determine the cause of the incident, and restore the affected system. Additionally, suitable patches and countermeasures were also implemented.

Recommendations

During the investigation of the security incident, several factors were identified that contributed to its occurrence and could potentially impact the organization's security, particularly in terms of integrity, confidentiality, and availability. Considering this particular incident, the security team recommends implementing the following security measures to help prevent similar incidents or reduce their impact on the organization.

Attachments

Attachment A: Indicators of Compromise (IoCs)

Several Indicators of Compromise (IoCs) were identified during the resolution of the security incident, which are listed in the following tables by IoC type. Table 4 lists the IP addresses along with the compromise phase of the device (connection, malware download, communication with the C&C server), the country of origin of the IP address,

its reputation according to the Cisco Talos service¹⁵, information on whether it is a mobile network or a proxy according to the ip.api¹⁶ service, and whether the IP address is listed in the AbuseIPDB¹⁷ reputation portal.

Table 4 – List of IP addresses as IoCs

IP address	Phase	Country	WEB reputation	Mobile	Proxy	AbuseIPDB	...

At the same time, several file names and hashes were identified as IoCs during the resolution of the security incident. Table 5 lists the relevant names and their associated digital fingerprints (hashes), along with the corresponding malware names according to VirusTotal¹⁸.

Table 5 – List of file names and hashes as IoCs

File name	Hash (SHA256)	Name (ESET-NOD32)	Name (Microsoft)	...

Attachment B: Email communication

Attachment C: Content of relevant files

Attachment D: Sources used

- [1] Incident Classification / Incident Taxonomy according to eCSIRT.net – adapted, available at <https://www.trusted-introducer.org/Incident-Classification-Taxonomy.pdf>
- [2] NIST Special Publication 800-61 r2 – Computer Security Incident Handling Guide [online], available at <https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-61r2.pdf>
- [3] Service Shodan [online], available at <https://www.shodan.io/>
- [4] Service AbuseIPDB [online], available at <https://www.abuseipdb.com>
- [5] Service Talos [online], available at <https://www.talosintelligence.com/>
- [6] Service ip-api [online], available at <https://ip-api.com/>
- [7] Service VirusTotal [online], available at <https://www.virustotal.com/>

Attachment E: Chain of custody (digital footprint tracking)

Table 6 – Chain of custody

1.7. Manual for incident report template

General information

The identification of a security incident report is determined by the serial number (for the current year). The name of the security incident is assigned by the organization but should be specific and unambiguous (e.g. based on the type of incident and the system involved). The general description of the incident should include the priority or severity of the incident, if known. It is appropriate to use the TLP (Traffic Light Protocol) to indicate the mode of reporting a security incident. As this is a security incident report, it is suitable to use Amber, Amber + Strict, or Red levels. These levels sufficiently restrict access to the document.

Number:	1	File type:		Recovered file:		Size:	
Case description:							
File name:							
Location of digital footprint acquisition:							
Notes:							
Timestamps							
Access:			Modify:			Change:	
Hash		MD5					
		SHA1					
Timestamp		Action executed with digital trace					

Managerial summary

This section should provide a brief overview of the security incident, its severity, and its impact on the organization. It should present management with clear and easily understandable information about what occurred and the potential consequences.

Incident description

A description of the security incident, including its type, along with the key facts and events that took place during the incident. This section should provide a detailed account of how the incident occurred and which systems or resources were affected. It should also outline the consequences resulting from the incident.

Security incident response

This section outlines the steps taken to resolve the security incident as well as the implemented actions. It also details the measures taken to limit damage and restore normal operations. This section should also document any communication with external institutions and authorities that were informed about the incident.

Further details on the handling of the security incident are provided in the following sections:

- Timeline of the security incident resolution,
- Methodology,
- Limitations,
- Analysis of digital traces and other information.

Timeline of the security incident resolution

A chronological record of the events related to the security incident, from its discovery to its resolution. This timeline should clearly outline how the events unfolded, and the duration of the resolution process.

Table 7 – Timeline of the security incident response (manual)

Timestamp (CEST)	Event
01.01.2024 at 00:00	Event description

Methodology

A description of the methods and procedures used to analyze and resolve the security incident. This section should provide a detailed view of how the teams identified, analyzed, and addressed the incident. It should also include the guidelines that were followed during the resolution of the security incident.

Limitations

This section identifies the limitations or challenges encountered during the resolution of the security incident. It should outline any constraints that may have impacted the success of the resolution process.

Analysis of digital traces and other information

A detailed analysis of digital traces, data, and other relevant information collected during the incident. This section should include technical details and the team's findings regarding the origin and scope of the incident. The analysis can be categorized as follows:

- OSINT (open source) analysis
- Basic information about the analyzed device
- File system analysis
- Logs
- Lateral movement
- Privilege escalation and persistence
- Timeline of incident response

Notes and other relevant information

Any additional notes, observations, or information relevant to understanding or resolving the incident should be included in this section.

Conclusion:

A summary of the key findings and lessons learned from the analysis of the incident. This section should provide closure and a clear understanding of what occurred during the incident, how it was resolved, and its implications.

Recommendations

This section provides specific recommendations for enhancing security and reducing the risk of similar incidents in the future. It may include both technical and organizational measures.

Attachments

Attachment A – Indicators of Compromise (IoCs)

Several Indicators of Compromise (IoCs) were identified during the resolution of the security incident, which are listed in the following tables by IoC type. Table 8 lists the IP addresses along with the compromise phase of the device (connection, malware download, communication with the C&C server), the country of origin of the IP address, its reputation according to the Cisco Talos service, information on whether it is a mobile network or a proxy according to the ip.api service, and whether the IP address is listed in the AbuseIPDB reputation portal.

Table 8 – List of IP addresses as IoCs (manual)

IP address	Phase	Country	WEB reputation	Mobile	Proxy	AbuseIPDB	...

At the same time, several file names and hashes were identified as IoCs during the resolution of the security incident. Table 9 lists the relevant names and their associated digital fingerprints (hashes), along with the malware name according to VirusTotal.

File name	Hash (SHA256)	Name (ESET-NOD32)	Name (Microsoft)	...

Table 9 – List of file names and hashes as IoCs (manual)

Attachment B – Email communication

This section should include all emails associated with the participants involved in the security incident.

Attachment C – Contents of relevant files

This section should include all contents of relevant files associated with the participants in the security incident.

Attachment D – Sources

This section should list all relevant resources used during the security incident response (e.g. online services, online databases, guidelines, etc.).

Attachment E – Chain of custody

Table 10– Chain of custody (manual)

Number:	1	File type:	Recovered file:	Size:
Case description:				
File name:				

Location of digital trace:					
Notes:					
Timestamps					
Access:		Modify:		Change:	
Digital footprint (hash)	MD5				
	SHA1				
Timestamp	Action executed with digital trace				

References

- [1] CICHONSKI P. et al.: Computer security incident handling guide. NIST Special Publication, 2012, 800.61: 1-147 [online] [13.3.2024], available at <https://csrc.nist.gov/pubs/sp/800/61/r2/final>
- [2] CIRCL.LU: CIRCL Taxonomy – Schemes of Classification in Incident Response and Detection, [online] [13.3.2024], available at <https://www.circl.lu/pub/taxonomy/>
- [3] Directive (EU) No. 2016/1148 of the European Parliament and of the Council of 6 July 2016 concerning measures for a high common level of security of network and information systems across the Union
- [4] Directive (EU) No. 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No. 910/2014 and Directive (EU) No. 2018/1972, and repealing Directive (EU) No. 2016/1148 (NIS 2 Directive)
- [5] ENISA: Reference Incident Classification Taxonomy, Task Force Status and Way Forward, January 2018, [online] [13.3.2024], available at https://www.enisa.europa.eu/publications/reference-incident-classification-taxonomy/at_download/fullReport
- [6] EUROPOL: Common Taxonomy for Law Enforcement and The National Network of CSIRTs, [online] [13.3.2024], available at https://www.europol.europa.eu/cms/sites/default/files/documents/common_taxonomy_for_law_enforcement_and_csirts_v1.3.pdf
- [7] FIRST: The Traffic Light Protocol (TLP) – Standards Definitions and Usage Guidance, August 2022 [online] [13.3.2024], available at <https://www.first.org/tlp>
- [8] ISO/IEC 27001:2022 – Information security, cybersecurity and privacy protection – Information security management systems
- [9] Regulation (EU) No. 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive No. 95/46/EC (GDPR)
- [10] STIKVOORT, D.: Incident Classification / Incident Taxonomy according to eCSIRT.net – adapted, [online] [13.3.2024], available at <https://www.trusted-introducer.org/Incident-Classification-Taxonomy.pdf>

2. Specific procedures for security incidents

The previous chapter outlined the general approach to handling security incidents, covering the key phases. However, different security incidents require specific procedures tailored to their unique characteristics and impact. This chapter focuses on detailed response protocols for various security threats, providing structured guidelines to effectively identify, contain, investigate, and remediate incidents.

The chapter begins with data breaches and addresses severity assessment, risk evaluation, notification processes, and post-incident reviews. Next, the chapter covers malware incidents, outlining the identification process, isolation of compromised devices, evidence collection, and forensic analysis techniques. A dedicated section discusses ransomware attacks, emphasizing rapid response measures, data recovery strategies, and forensic investigation techniques. The chapter also explores social engineering attacks, including phishing, smishing (SMS phishing), and vishing (voice phishing). Finally, the chapter provides a detailed procedure for handling Linux server compromises, outlining incident identification, forensic investigation, and response steps to mitigate unauthorized access and system exploitation.

2.1. Specific procedure – Data breach

Identifying the security incident phase

A Data breach is any incident or security breach that results in accidental or unlawful destruction, unauthorized retention, misuse, loss, alteration, disclosure of, or access to data that is transmitted, stored, or otherwise processed by the organization, its employees, contractors, or service providers, including broader privacy violations.

It is assumed that the "Incident Triage" and "Incident Recording and Registration" steps have been completed at this stage. This means the incident has been identified as a security incident, it has been recorded and assigned an investigator. The process proceeds to the "Incident Response" phase now.

According to the classification of security incidents (hereafter referred to as SI), leakage of personal data falls into the category of "Information Security Threat" and should be distinguished from "Unauthorized Access to Information", "Unauthorized Alteration of Information", and "Other" in the context of the solution. From a legal perspective, leaking Business Intelligence data constitutes a criminal offence (similarly to compromising email accounts), and is classified as "Unauthorized access to a computer system".

How to respond to data breaches

This data breach response plan defines key terms, establishes procedures and delineates clear lines of authority for the organization's staff in the event of a confirmed or suspected data breach. The plan has been designed to enable the organization to promptly contain, assess, and respond to data breaches, minimizing potential harm to affected individuals.

When responding to a breach or suspected data breach, there are five essential steps to consider:

- **STEP 1: Assess severity**
- **STEP 2: Contain the breach**
- **STEP 3: Evaluate the risks associated with the breach**
- **STEP 4: Notify**
- **STEP 5: Review and report the incident**

Severity assessment

The procedure for responding to data breach incidents may vary based on their severity. Data breaches should be dealt with on a case-by-case basis by assessing the risks involved and determining the appropriate course of action based on that assessment. When deciding whether to escalate a data breach, the responsible person should consider the following:

- Are multiple individuals impacted by the confirmed or suspected breach?
- Is there, or could there be, a real risk of serious harm to the affected individuals?
- Does the confirmed or suspected breach suggest a systemic issue within the organization's processes or procedures?
- Could the confirmed or suspected breach attract media or stakeholder attention?
- The type of personal information involved, particularly if sensitive data has been affected.
- Whether any safeguards, such as encryption, would prevent the unauthorized party from using the information.
- The potential harm that could result from the breach (Breach risk assessment).
- The nature of the damage that could result from the leak (e.g. identity theft, financial loss, etc).

This process also involves validating systems (checking whether these are existing accounts) and conducting a preliminary assessment. If the incident is determined to be a false positive, it should be closed accordingly. Even if the responsible person decides not to escalate the case, or if the incident is confirmed to be a false positive, the following steps must still be carried out:

1. Inform the reporter of the findings obtained during the incident verification.
2. Record the incident in the internal incident management tool, including a description of the actual or suspected violation and the actions taken by the responsible person or employee to address the issue, along with the outcomes of the actions taken.
3. Have the responsible person confirm that no further action is required.

Breach containment

During the containment phase of a data breach, immediate actions are taken to limit the extent of the breach and prevent further damage. Key steps include:

- **Identify and stop the breach:** Quickly determine the source of the breach and halt any ongoing data exposure. This may involve disconnecting affected systems from the network, revoking access privileges or shutting down compromised accounts.
- **Secure vulnerable systems:** Ensure that all systems, networks, and devices affected by the breach are secured. Apply patches, update software or change passwords as needed to prevent further unauthorized access.
- **Preserve evidence:** Maintain logs, records, and other evidence related to the breach to facilitate a thorough investigation at a later time.

- **Communicate internally:** Notify the supervisor that this is an ongoing incident and record the time the incident started. In the event of a critical or serious security incident, a crisis management team consisting of designated personnel should be activated.

Note: If this is an active or ongoing security incident and the attacker has active access to the system, shutting down the system may have further consequences. In this case, it is necessary to consider possible impacts resulting from the shutdown. Depending on the case, the choice of solutions may vary (for example, in disk encryption, disconnecting the external network will not help). A personal data leak may not mandate the system shutdown. In any case, it is important that the person leading the IR process be consulted and the management notified of the situation.

Evaluate the risks associated with the breach

- **Assess the scope:** Evaluate the extent of the breach, including identifying which systems and data have been compromised, assess the impact on the organization.
- **Collect information about the breach, such as:**
 - the date, time, duration and location of the breach
 - the type of personal information affected by the breach
 - how the breach was discovered and who discovered the breach
 - the cause and extent of the breach
 - a list of affected individuals, or possibly affected individuals
 - the risk of serious harm to the affected individuals
 - the risk of other harms
- **Maintain thorough records** of the suspected breach and the actions of the Response Team, including steps taken to address the situation and the decisions made.
- **Establish** the cause and extent of the breach.

Notification

Identify persons to be informed about the breach, both within and outside the organization at this initial stage.

It is required by law that all persons concerned must be informed of the breach with a clear, simple description of the nature (see: Annexes, section 5.1).

It is also necessary to contact the individual who reported the personal data leak (incident reporter). The information shared should include:

- description of the findings to date,
- measures taken and their impact,
- cooperation with the CSIRT team in determining the next steps to be taken and the implementation of further measures, if any.

Having been credibly notified of an incident, the organization is required to report the findings to the Data Protection Authority (DPA) within 72 hours. Communication with the DPA should take place using a dedicated form, indicating the scope of the breach, the number of individuals affected, incident resolution, future plans to prevent repeat incidents, etc.

- If the exact number of affected persons is unknown or cannot be determined, the number of users on the affected system should be reported.
- In terms of the nature of the personal data protection breach, both confidentiality and integrity may have been compromised (data may have been altered).

- When determining the time period of the security incident, it is necessary to provide (if available) logs, such as server, memory, or application logs that can help pinpoint the time of the incident. If the exact date and time of the system compromise are unknown, the last known the most recent log entry confirming the system compromise should be used. Logs need to be retained for post-incident documentation purposes.

Note: It is essential to assess whether other parties, such as the police, the Data Protection Authority, or other relevant agencies or organizations should be notified, or if there are specific contractual provisions or Memoranda of Understanding that require that specific entities be informed about the breach. Both the responsible organization and the timeframe for reporting an incident may vary depending on the country in which the incident is being handled.

In the event of media interest, the policy is not to communicate directly with the media; all questions should be forwarded to the Press Office.

Review and report incident

At this stage:

- The CSIRT team, in collaboration with the person in charge of personal data within the organization, should perform a complete data analysis to assess the impact and implications for future operations. The results should be communicated to supervisors based on the incident's severity. Examples of potential causes include system vulnerabilities or leaked admin data.
- The incident should be logged in the internal ticketing system.
- Security Incident Report should be created for incidents of high and critical severity that threaten part of or the entire organization. The report should include a complete history of the incident and details of its resolution. Additional information about the incident should also be disseminated, internally - for employees/members of the organization, or externally - for the media. A joint meeting of CSIRT members who were actively involved in the incident resolution and the administrators of the affected system should also be held to review the overall incident management process and to improve steps to be taken in case of a repeated incident in the future.
- All systems should be tested for security vulnerabilities and all current workflows should be reviewed accordingly.

Attachments

Attachment A

Model message notifying the user of a suspected leak of personal data:

TLP:AMBER

Dear XXX,

We would like to alert you to a possible leak of your login credentials (email address, password), which can happen when using public resources available to the organization. The leaked data related to a specific (xxx) service can be accessed via the following link: – xxx.txt. As this appears to be a probable leak, we would kindly ask you to identify the leaked password and associate it with a corresponding service.

Please ensure that the (new) password is sufficiently long, complex and different from any passwords you are already using for other services. This is a necessary precaution as

there is high probability that the leaked login data could be misused for illegitimate purposes.

If you have any questions, please do not hesitate to contact us.

Sincerely,

TLP:AMBER

Note: The highlighted part of the text should be replaced with a unique link in the folder for external sharing on the repository. A separate folder needs to be created for each data leak at that location.

References

- [1] (Data breach response plan. Available at: [https://data.nsw.gov.au/sites/default/files/inline-files/Data Breach Response Plan.pdf](https://data.nsw.gov.au/sites/default/files/inline-files/Data%20Breach%20Response%20Plan.pdf) (Accessed: 5 August 2024).
- [2] Oaic (2024) Part 2∷ preparing a data breach response plan, OAIC. Available at: <https://www.oaic.gov.au/privacy/privacy-guidance-for-organisations-and-government-agencies/preventing-preparing-for-and-responding-to-data-breaches/data-breach-preparation-and-response/part-2-preparing-a-data-breach-response-plan> (Accessed: 5 August 2024).
- [3] Computer Security Incident Handling Guide. Available at: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf> (Accessed: 5 August 2024).
- [4] Data breach playbook | cyberalberta. Available at: <https://cyberalberta.ca/system/files/data-breach-playbook.pdf> (Accessed: 5 August 2024).

2.2. Specific Procedure – Malware

Identifying the phases of a security incident

According to the diagram outlining the proposed phases of security incident resolution (Chapter 1), as shown in Figure 2 the process is currently in the "Incident Response" phase. It is assumed that the phases "Incident Recording and Registration" and "Incident Triage" have already been completed.

According to the Security Incident Classification, malware is categorized as "Malicious code" and can be further classified as "Information gathering", "Threats to information security", or "Other".

From a legal perspective, malware should be regarded as a criminal offence (similarly to compromising email accounts) and classified as "Unauthorized access to a computer system".

Quick procedure

The following steps outline a simplified approach to malware incident response, ensuring swift containment, analysis, and resolution:

- Incident logging
- Identification of affected devices and determination of the extent of compromise
- Thwarting the spread of malware by isolating affected devices

- Collecting relevant evidence (e.g., forensic analysis, external cooperation)
- Cleaning all affected systems
- Performing static analysis
 - Checking basic file information
 - Identifying the file type
 - File hashes
 - Deep code analysis
 - OSINT research and analysis
- Performing dynamic analysis
- Removing the malware
- Documentation of findings / Final incident logging
- Creation of detection rules

Malware resolution procedure

When dealing with malicious code incidents, it is necessary to isolate the affected devices in order to rule out the possibility that the malicious code will spread further. Other essential actions include retrieving suspicious files using forensic analysis and performing a thorough examination of the sample using static and dynamic analyses to remove the malicious code and recover the devices.

Device isolation can take various forms. The most common ones include disconnecting the device from the Internet and associated networks, blocking suspicious URLs and IP addresses, or even shutting down the device. It is important to note, however, that shutting down a device can compromise forensic evidence and it should therefore be done with caution.

In analyses (static or dynamic), various elements, such as timestamps, file type, hash of the malicious code sample and other details of the file are analyzed, based on a given malicious code sample. The analysis also covers the behavior of the sample itself on the device, the processes created or infected along with the persistence mechanisms established.

If, after analyzing the sample, the Forensic Analysis Team is unable to clearly identify the type of malicious code, the report will instead detail its capabilities.

A summary of the entire resolution process is shown in Figure 1.

Types of malware

Malware (malicious software or code) has been created to damage a computer system or intercept its data. There are many different types of malicious code, each with its own goals and methods. The most common types of malicious code include [4],[5],[6]:

- **Viruses:** self-replicating programs that spread by infecting other files. They can corrupt files, steal data, or disrupt the functioning of a system.
- **Worms:** Self-replicating programs that spread through a network. They can overwhelm network traffic or damage systems.
- **Trojans:** Programs that pretend to be useful or harmless but contain hidden malicious code. They can steal data, install other malware or damage systems.
- **Ransomware:** A type of malware that encrypts a victim's files and demands a ransom for the files to be decrypted. In many cases, ransomware also exfiltrates data, stealing sensitive information before encryption to increase leverage over the victim.
- **Spyware:** Software that tracks a victim's activities and steals personal information, such as passwords or credit card numbers.
- **Adware:** Software that displays unwanted advertising content to the victim.

- **Rootkit:** A type of malicious code that gains administrative privileges on a system. It can hide files, change system configuration and install other malicious code.

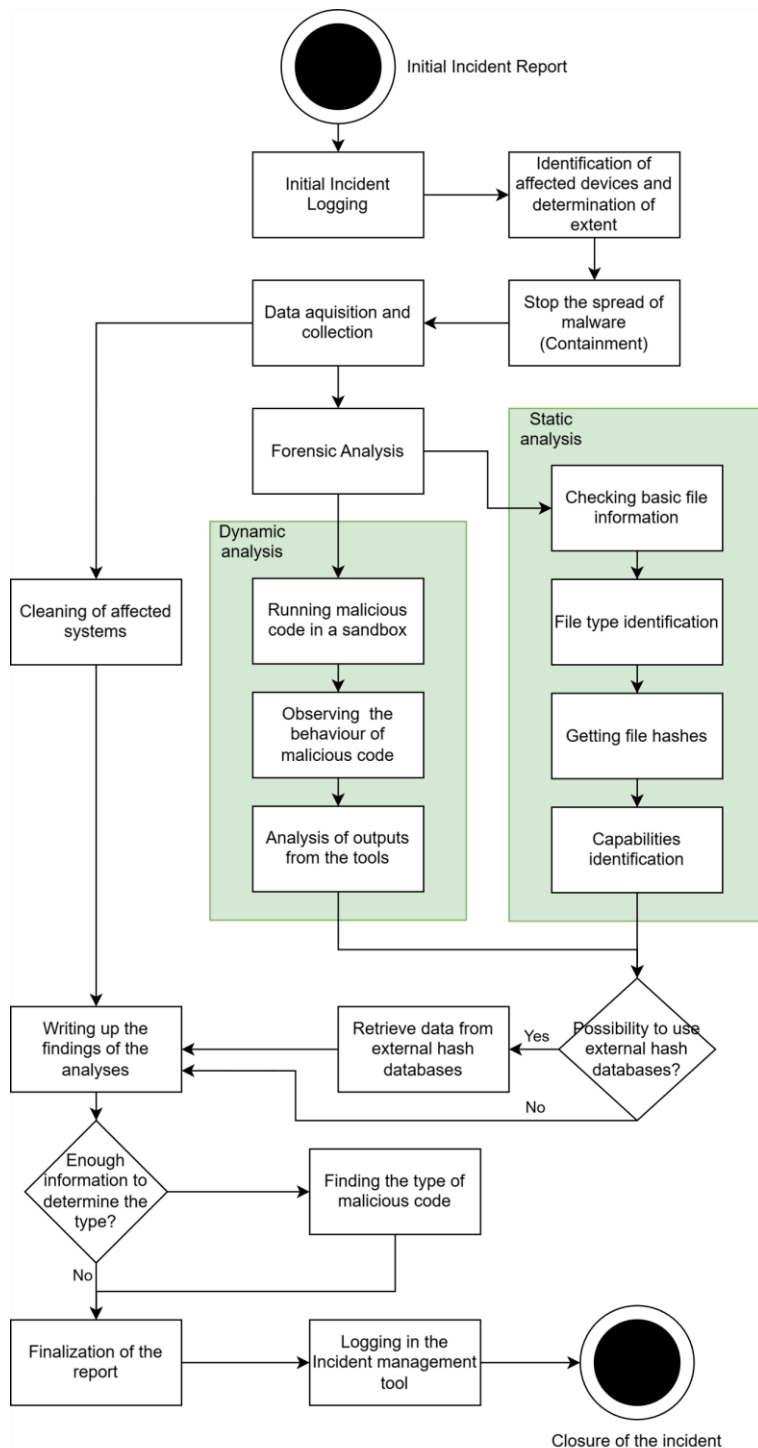


Figure 3: Incident handling procedure [1][2]

Incident logging

Incidents are recorded in the Incident Management tool by creating a new case. The Forensic Analysis Team, or a person responsible for logging the progress of the analysis, adds notes, fills in missing tags, etc. throughout the investigation to ensure the proper logging of the performed tasks. The information is collected so that the team can, at any point during the investigation, better understand the scope and nature of the incident as well as the actions taken.

The tags include malicious code type (e.g. ransomware), name of the machine on which the malicious code was detected and a general "malicious code" tag, used if at this stage of the investigation the actual code type is not yet known. In this case, the Analysis Team will proceed without adding any specific tag. The description can outline the different steps taken at this stage of the incident.

Identification and isolation of compromised devices

The goal of this stage is to identify compromised devices and isolate them using internal tools (such as Microsoft EDR and ESET Antivir). The Analysis Team monitors various types of logs obtained from individual tools to detect and identify compromised devices or ranges.

The next step is to isolate any compromised devices to prevent the malware from spreading.

Types of isolation:

- **Shutting down the compromised devices** – this action requires caution as it may result in the loss of forensic evidence. Shutdown should only be considered if the specific circumstances and the significance of the services provided by the device permit it.
- **Quarantine/isolation of the device** – placing the device in an isolated network segment without access to any sensitive data, critical systems or other uninfected devices.
- **Shutting down Internet access** – in the case of network segmentation, separation of affected segments from unaffected segments may be considered as a viable option.
- Blocking suspicious URLs and IP addresses
- **Blocking remote access applications and VPNs** – terminating access to compromised devices using RDP and VPNs.
- **Airplane mode** – with mobile devices, airplane mode can block any communication with the compromised device.

Collecting evidence

Collecting evidence is an important step, ensuring that the organization can analyze incidents and assess their extent, identify attack vectors and support forensic analysis. It is essential to choose a proper method of collecting evidence as improper collection can compromise its integrity, admissibility in court and overall reliability. Therefore, we recommend using widely accepted and reputable tools, such as Eric Zimmerman solutions, FTK Imager etc.

- Obtaining relevant information and data. Relevant data includes logs, screenshots, network traffic logs, disk images, communications related to the attack, and other evidence that can be helpful when investigating the situation. The following tools are recommended for collecting files from an infected machine:

- Disk Images – AccessData FTK Imager (<https://accessdata-ftp-imager.software.informer.com/3.1/>)
 - Memory Dumps – Volatility (<https://github.com/volatilityfoundation/volatility>)
 - Windows logs – Windows Event viewer.
 - Windows files:
 - Pagefile.sys – Virtual memory management file
 - Hibernfil.sys – A file used to store the system state during hibernation
 - Eric Zimmerman tools (<https://ericzimmerman.github.io/#!index.md>)
 - Note: The usability of the specific tools may vary depending on the operating system involved, as some tools are more effective or compatible with particular environments.
- Collecting data from outside the organization – cooperation with the Internet Service Provider for relevant network traffic data and further insights.
 - Forensic analysis of the compromised systems, in accordance with internal guidelines for forensic analysis.

Static analysis

Static analysis is performed without running the file and aims to find out as much information about the file itself as possible. The procedure varies depending on the file type, as each type has its own specific characteristics. The already identified file types include those from the Microsoft Office family (.doc, .docx, .xlsx,) PDF files, .exe-type executable files, and many others, each requiring specific analytical procedures.

Checking basic information about the file is one of the key stages of the analysis. It involves collecting essential information about the file such as size, date of creation and modification, the author/creator and the digital signature of the malware. Any inconsistencies in the file data should raise suspicion and lead to further investigation.

With Windows devices you can extract file properties using PowerShell commands, such as Get-ItemProperty and Get-AuthenticodeSignature.

```
PS C:\> Get-ItemProperty .\kape.exe | Format-List

Directory: C:\

Name           : kape.exe
Length         : 7003832
CreationTime    : 03-Jan-23 11:47:12
LastWriteTime   : 03-Jan-23 11:47:12
LastAccessTime  : 30-Jan-25 08:55:30
Mode           : -a----
LinkType       :
Target         : {}
VersionInfo    : File:           C:\kape.exe
                  InternalName:  kape.exe
                  OriginalFilename: kape.exe
                  FileVersion:   1.3.0.2
                  FileDescription: kape
                  Product:       kape
                  ProductVersion: 1.3.0.2
                  Debug:         False
                  Patched:       False
                  PreRelease:    False
                  PrivateBuild:  False
                  SpecialBuild:  False
                  Language:      Language Neutral
```

Figure 4: Windows File properties extracted with PowerShell command.

```
PS C:\> Get-AuthenticodeSignature .\kape.exe

Directory: C:\

SignerCertificate          Status          Path
-----
DBB201F3C09629E82254C140E8B8043EFB46B0FB Valid          kape.exe
```

Figure 5: Windows File Digital signature extracted with PowerShell command.

As can be seen in Figures 4 and 5, crucial information can be extracted with a simple PowerShell command or using a GUI to access file properties.

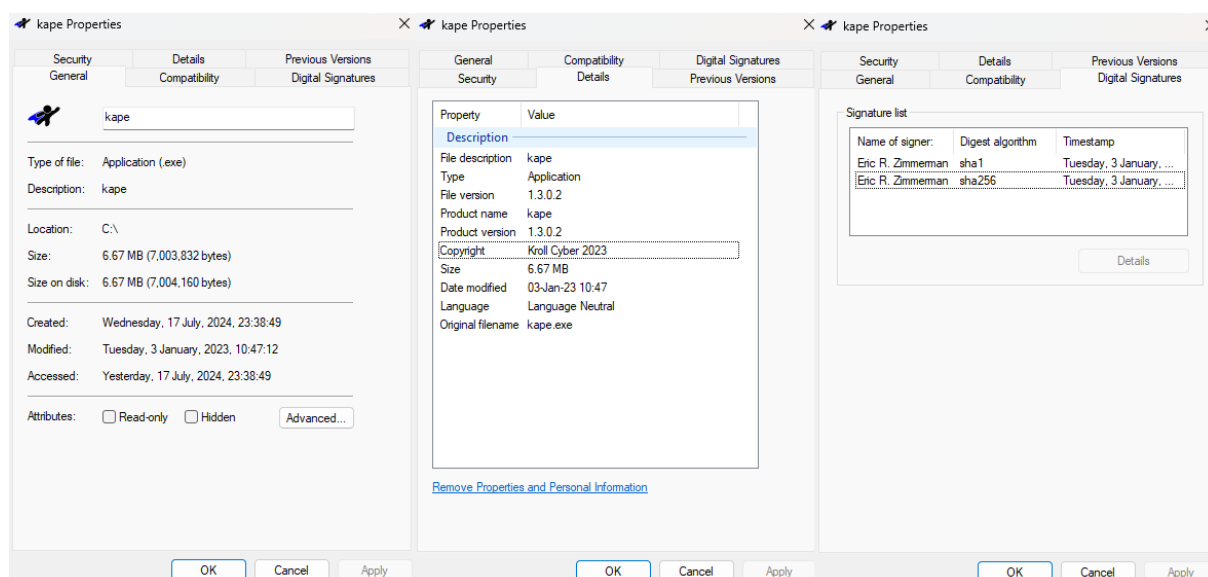


Figure 6: Windows file properties in GUI

In Windows, file properties can be retrieved by right-clicking on the file and selecting "Properties". The properties window displays information about the file type, size, creation and modification date, etc. as seen in Figure 4. However, using PowerShell commands is easier and faster as it provides all relevant data at the user's disposal.

Machines running Linux provide access to the `stat` tool, which can generate a list of basic properties and details about the file, such as file size, User ID, Group ID, creation and modification date.

```
root@desktop:/# stat kape.exe
  File: kape.exe
  Size: 7003832      Blocks: 13680      IO Block: 4096   regular file
Device: 820h/2080d Inode: 1249       Links: 1
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2024-07-17 23:54:02.152006354 +0200
Modify: 2024-07-17 23:53:44.182011538 +0200
Change: 2024-07-17 23:53:44.182011538 +0200
 Birth: 2024-07-17 23:53:44.172011539 +0200
root@desktop:/#
```

Figure 7: Linux file properties

The Analysis Team should document the basic information and characteristics of the suspicious file in order to facilitate further analysis, identify additional indicators of compromise and record relevant information on the file in the final report.

File type identification

Identifying the file type is essential for understanding how the file may be used and for selecting appropriate tools for further analysis. In general, files can fall into one of the following categories:

- **ASCII files** – these are the simplest, plain text (.txt, .html, .xml) files. ASCII files have no structure and their content can be read using a basic text editor.
- **Structured files** – these files have a binary structure that varies by file type (e.g., .pdf, .exe). The first bytes in the hexadecimal interpretation of a file define a header that specifies the file type. Note that not all files have a full header, extension and footer (e.g. portable executables or PEs).

File type identification

- **Linux file command** – this is the recommended option; use: file filename to determine the file type.
- **Based on file extension and icon** - this method is less safe as both the extension and the icon can be easily changed without affecting the file's structure or content.
- **Manual identification** - by searching and comparing the file's header (or footer).

It is advisable to use built-in tools such as Linux file commands since they can help the Analysis Team to access information about particular files faster and easier than other methods. Therefore, Linux file commands or equivalent tools are recommended for file type identification.

Examples of selected file headers:

4D 5A - Windows/DOS executable file - **.exe**

25 50 44 46 - Adobe Portable Document Format - **.pdf**

50 4B 03 04 14 00 06 00 - Microsoft Office Open XML Format Document - **.docx, .xlsx, .pptx**

In order to access the header of a file and to view the file in a hexadecimal form, you can use CyberChef – or, specifically, its Detect File Type recipe as well as a tool called "Hex Viewer". The CyberChef option is preferred, **if the nature of the case** permits the use of an external online tool. However, the "Hex Viewer" option is recommended when dealing with highly sensitive information.

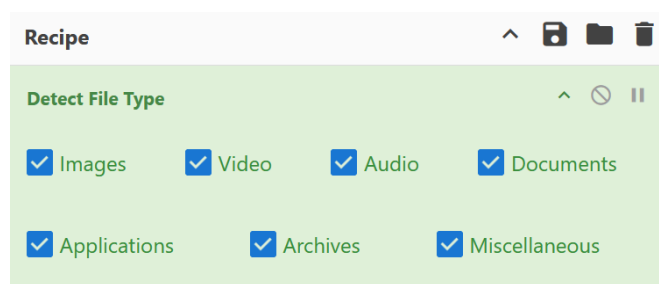


Figure 8: CyberChef Detect File Type Recipe.

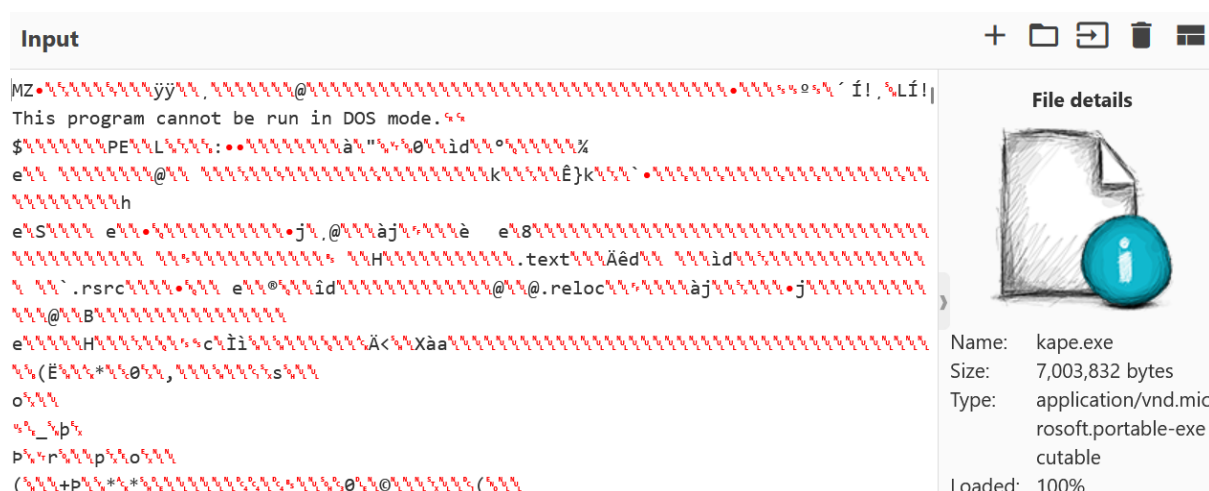


Figure 9: CyberChef Open File as Input.

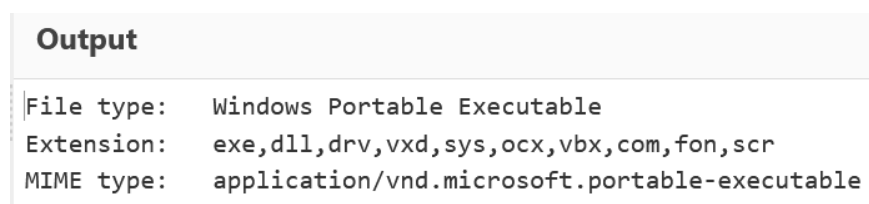


Figure 10: CyberChef Detect File Type Output.

To determine the file type using CyberChef, the "Detect File Type" recipe can be selected, as shown in Figure 8. In the next step, proceed to *Open File as Input* and select the file for analysis (Figure 9). The resulting output shows the detected file type along with a list of possible extensions associated with it (Figure 10).

If the circumstances do not allow the use of external online tools, a *Hex Viewer* can be employed instead. An example of such a tool is PSpad, which automatically detects the binary structure of the examined file and displays it in hexadecimal format. The Forensic Analysis Team must then determine the file type and the [header list](#) for the file type.

The Analysis Team records the file type determined in order to establish the subsequent steps in the examination of the suspicious file.

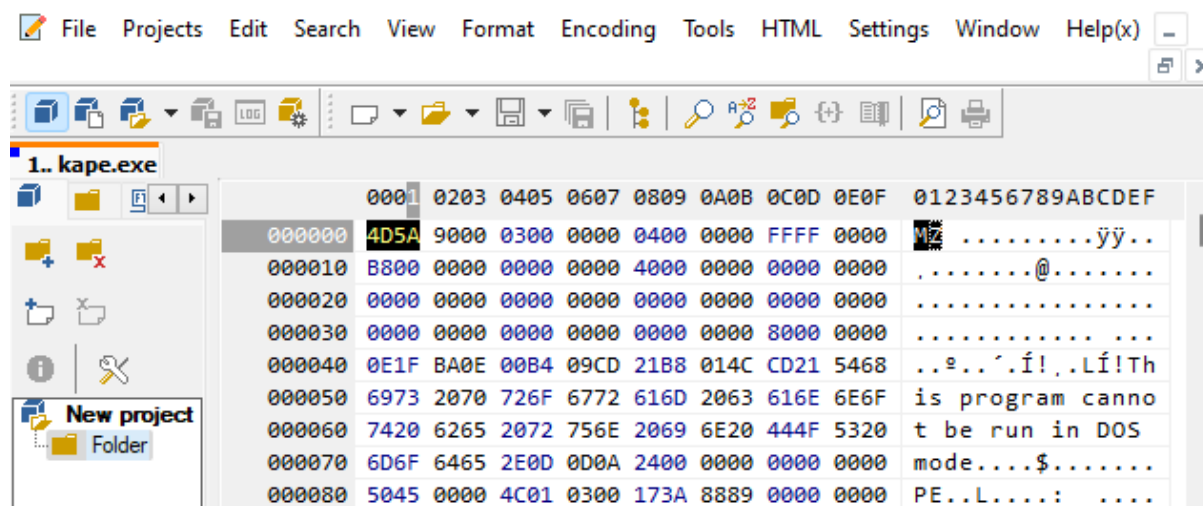


Figure 11: Identifying the file type using PSpad (.exe)

File hash

Hashing is the process of generating a fixed-length string, known as a file hash. The process can be applied to any file regardless of its type or size. The hash serves as a signature for the file.

This signature can be used to check the hash in databases of known malware hashes. Classifying a sample using its hash helps determine whether it belongs to a well-known malware family or to an Advanced Persistent Threat (APT) group.

Note: No two different files should generate the same hash value. However, some hashing functions might produce the same hash under certain conditions, a phenomenon known as hashing collision.

The most commonly used hashing functions in malware analysis are:

SHA256 – output size: **256 bit**, number of hexadecimal values: 64

SHA512 – output size: **512 bit**, number of hexadecimal values: 128

Other hashing functions can be used, if necessary. Note, however, that hashing functions like **SHA1** and **MD5** are vulnerable to collisions and should not be applied alone. Therefore, we recommend the use of more reliable alternatives: SHA-256 or SHA-512. If SHA-1 or MD5 must be used, they should be combined with stronger hashing functions, so that two hashes are generated for each investigated file.

Other hashing functions used in malware analysis include Fuzzy Hash, Import Hash and Section Hash.

Useful hashing tools:

- Linux CLI commands: `sha512sum [filename]`, `sha256 [filename]`
- Windows CMD commands: [Get-FileHash](#) with the `-Algorithm` switch to select the hashing algorithm
- HashMyFiles (https://www.nirsoft.net/utils/hash_my_files.html)
- Hasher (<https://ericzimmerman.github.io/#!index.md>)

After running the sample through the hash function, if the nature of the case allows it, the resulting hash value can be used to search the external databases, such as VirusTotal, Tria.ge to determine whether it has been previously identified. If a match is found, it provides an insight into how to deal with the sample, what artifacts are associated with it, what indicators to look out for, and how to successfully remove malware from the infected systems. The hash value can be used to locate other potentially compromised systems, serving as an indicator of compromise (IOC).

It is important to note that even a single-bit change in a file will result in a completely different hash value. This presents a challenge, especially with malware, as the sample can be modified for a specific target without changing its functionality. However, from a hashing perspective, it appears to be an entirely different file, unrelated to the original.

Malware that is capable of modifying itself is known as polymorphic malware or polymorphic code. In computer science, polymorphic code uses a polymorphic engine to mutate while preserving the original algorithm - that is: the code changes each time it is executed, but the function of the code (its semantics) remains unchanged. Accordingly, each instance will produce a different hash value (signature). Therefore, the recommended option for polymorphic malware is Fuzzy Hashing, which provides a percentage match between two files. SSDEEP is a CLI tool used for malware analysis and fuzzy hashing checks. (<https://ssdeep-project.github.io/ssdeep/index.html>)

```
PS C:\> Get-FileHash -Algorithm SHA256 .\kape.exe
```

Algorithm	Hash	Path
SHA256	6167472179D0B5B028560DCC84EA1A2E3CB2D7128DD18E4E9278263B86A4318B	C:\kape...

Figure 12: SHA256 hash of a file obtained via PowerShell cmdlet.

```
root@desktop:/# sha256sum kape.exe
6167472179d0b5b028560dcc84ea1a2e3cb2d7128dd18e4e9278263b86a4318b kape.exe
root@desktop:/# |
```

Figure 13: SHA256 hash of a file obtained using the Bash command.

Analysts can use the above-mentioned tools to obtain the hash of a file by using solutions such as PowerShell command on a Windows machine (Figure 12) or Linux CLI commands on a Linux distribution (Figure 13).

The analyst reports the suspicious file hash as a unique file identifier. This allows the analyst to determine whether the file has been subsequently altered and, if the nature of the case allows, to launch a subsequent analysis using external databases, such as VirusTotal, Tria.ge and others.

Disassembly

Dissassembly is a process of converting binary code into a human-readable format. This process allows the Analysis Team to examine the operation of the program and identify its various components and functions.

While manual disassembly provides a detailed insight into the analyzed sample, it is also complex and time consuming to perform. Automated disassembly tools can assist in identifying suspicious functionalities and behaviors of the analyzed sample. For example, tools like [CAPA](#) can automatically recognize patterns and capabilities associated with malicious activity and therefore speed up the analysis.

```
PS C:\> .\capa.exe .\kape.exe

md5      7e09af85742522ab31748c8992839ae6
sha1     a3b5cdfb814acf9e924e1cb29d4d2def537e8b90
sha256   6167472179d0b5b028560dcc84ea1a2e3cb2d7128dd18e4e9278263b86a4318b
analysis static
os       any
format   dotnet
arch     any
path     C:/kape.exe
```

Figure 14: Example use of CAPA on test file.

Capability	Namespace
self_delete (2 matches)	anti-analysis/anti-forensic/self-deletion
receive_data	communication
resolve_DNS	communication/dns
find_data_using_regex_in_.NET (8 matches)	data-manipulation/regex
contains_PDB_path	executable/pe/pdb
access_.NET_resource (2 matches)	executable/resource
embed_dependencies_as_resources_using_Fody/Costura	executable/resource
accept_command_line_arguments	host-interaction/cli
manipulate_console_buffer (15 matches)	host-interaction/console
get_common_file_path	host-interaction/file-system
get_file_version_info	host-interaction/file-system/meta
get_hostname	host-interaction/os/hostname

Figure 15: Capabilities of a test file revealed by CAPA

As shown in Figures 14 and 15, a simple command was executed, allowing the CAPA tool to analyze the test file and identify its capabilities. This significantly reduced the time required for analysis, compared to manual disassembly.

Disassembly can be performed using the IDAFree tool. IDAFree is a free version of the IDA Pro disassembler that offers a wide range of functionalities for binary analysis.

To begin the analysis, the Analysis Team launches IDA Free, selects the "New" button at startup to parse a new file and then selects the suspicious file for parsing.

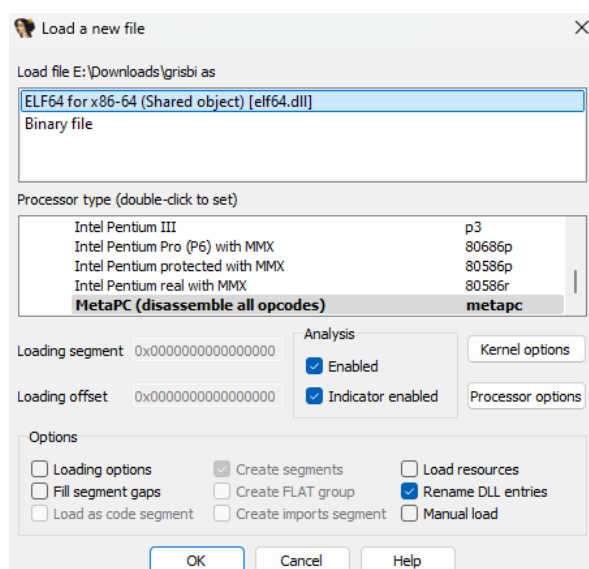


Figure 16: File selection confirmation in IDA Free window

As can be seen in Figure 16, after pressing OK the Analysis Team proceeds with file analysis, focusing on non-standard functions and code blocks that may indicate malicious activity. This may include closer investigation of files with suspicious names, using libraries offering unexpected functionalities (encryption libraries allowing the editing of .xml files, etc.), and other non-standard functionalities and components.

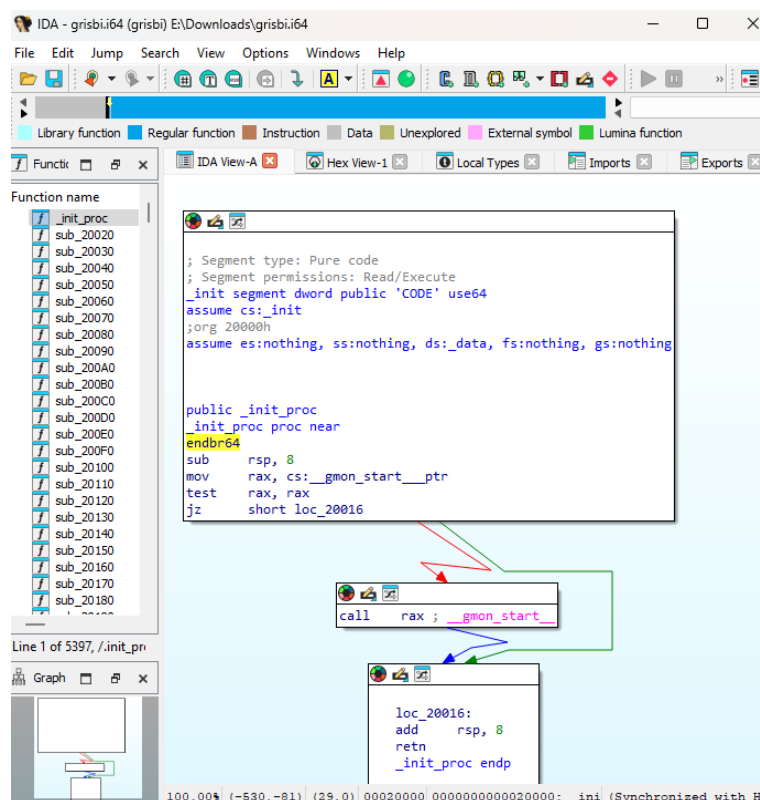


Figure 17: Preview of a newly opened file in IDA Free.

The IDA Free tool also allows the Analysis Team to search for suspicious strings within a given file. Such a search can be enabled through the Strings window.

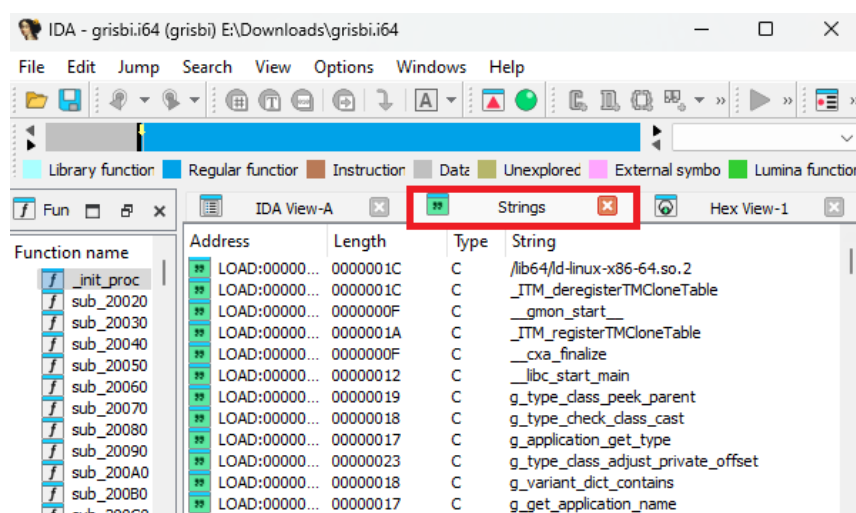


Figure 18: "Strings" window in IDA Free.

At the end the analyst will log the findings, like type of malware and etc. into the Incident management tool.

OSINT research and analysis

The analyst can use OSINT tools, like Google Dorking, external hash databases, threat intelligence platforms to gather information about samples, provided the nature of the case allows the use of external tools.

Some of the OSINT methods include:

- Google Dorking – Using advanced search operators to find exposed information.
- External Hash Databases – Checking hash values in databases, such as VirusTotal, Hybrid Analysis.
- Threat Intelligence Platforms – MITRE ATT&CK and others
- Domain and IP Analysis – tools like WHOIS lookup, Shodan, AbuseIPDB.
- Social media and Dark Web Monitoring – Searching platforms like Facebook, Twitter, etc.

By effectively utilizing OSINT methods, the Analysis Team can further their investigation and add contextual insights to the analysis of specific samples, which can help the team understand the scope of the incident.

The team of analysts log the findings into the incident management tool to keep the track of the collected information related to the sample and the incident.

Dynamic analysis

Dynamic analysis involves running malware in an isolated environment (sandbox) and monitoring its behavior, including changes in the file system, registries, process creation. The analysts execute the malicious code only in an isolated environment, typically within a virtual machine.

Dynamic analysis can be automated using external tools like ANY.RUN or Triage. However, it is important to note that the nature and the scope of the incident may render the use of external tools like ANY.RUN or Triage impossible, particularly when handling sensitive data. Nevertheless, these tools can considerably speed up the process of dynamic analysis.

Dynamic analysis can be divided into 4 phases. The first phase involves the initial setup of the virtual machine along with uploading the necessary tools and creating a backup (snapshot). During the second phase the system is configured, the tools are launched and the malicious code sample is uploaded to the virtual machine. In the third phase, the team of analysts runs the malicious code, closely monitoring its activity and behavior. Phase 3 concludes with individual outputs being collected from the analytical tools. In the fourth and final phase, the collected data are analyzed and the findings are documented for further references and reporting.

Phase 1

The analyst needs a virtual machine that is ready to analyze the malicious code. A snapshot of the virtual machine is necessary in case the analyst needs to revert to the original state after the completed analysis or if certain steps need to be repeated. Additionally, the Analysis Team should ensure that they have all tools necessary for the dynamic analysis of the malicious code sample and for uploading or installing the tools onto the virtual machine where the analysis will be performed.

List of tools:

- [Autoruns](#)
- [Process Explorer](#)
- [ListDlls](#)
- [WireShark](#)

Phase 2

The Analysis Team launches Process Explorer and saves the current list of processes to a .txt file by selecting "File" and then "Save As", to choose a save location and then store the file outside the virtual machine. The Process Explorer remains active.

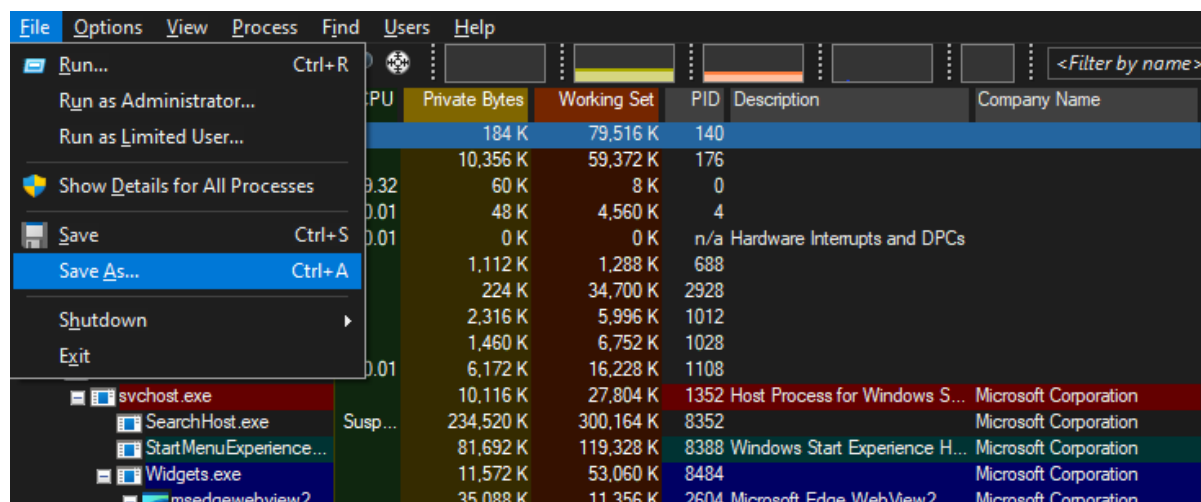


Figure 19: Saving the output from Process Explorer.

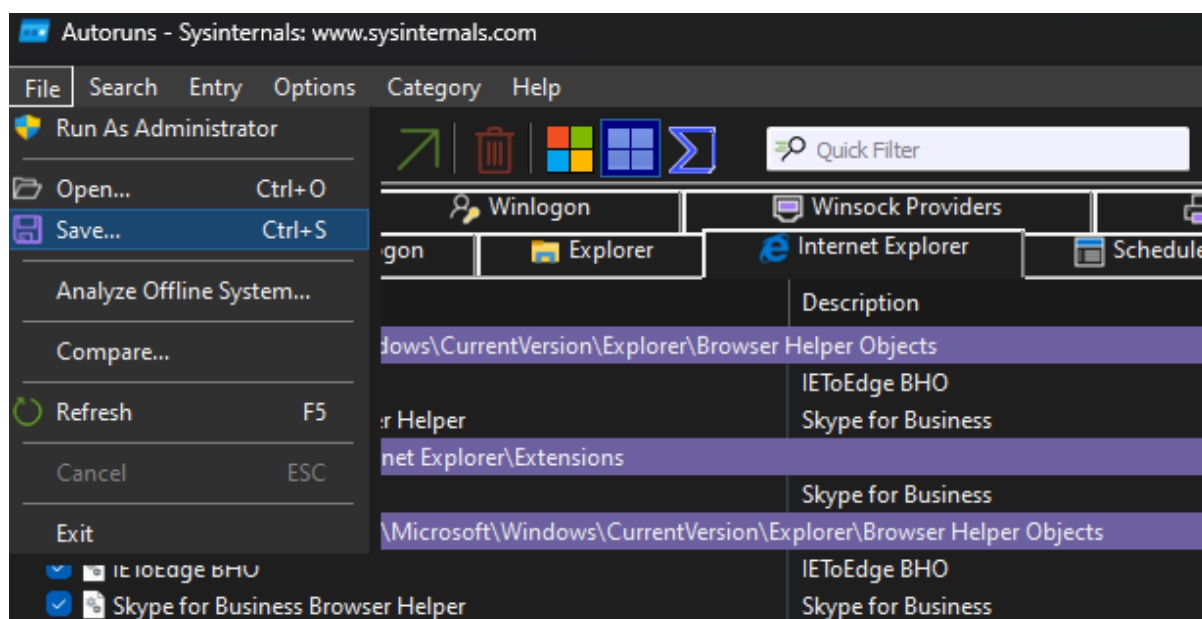


Figure 20: Saving the output from the Autoruns tool.

The **Autoruns** tool is launched and, following the same procedure as previously, the output is saved by selecting "File" → "Save..." (Figure 19). The file is then saved and stored outside the **virtual machine** (Figure 20).

.dll Files are obtained using the listdlls tool, where a text file for the output of the tool is created using Powershell commands. As a next step, the listdlls.exe tool is run and the obtained output is sent to the newly created file. Finally, the file is securely stored outside the virtual machine.

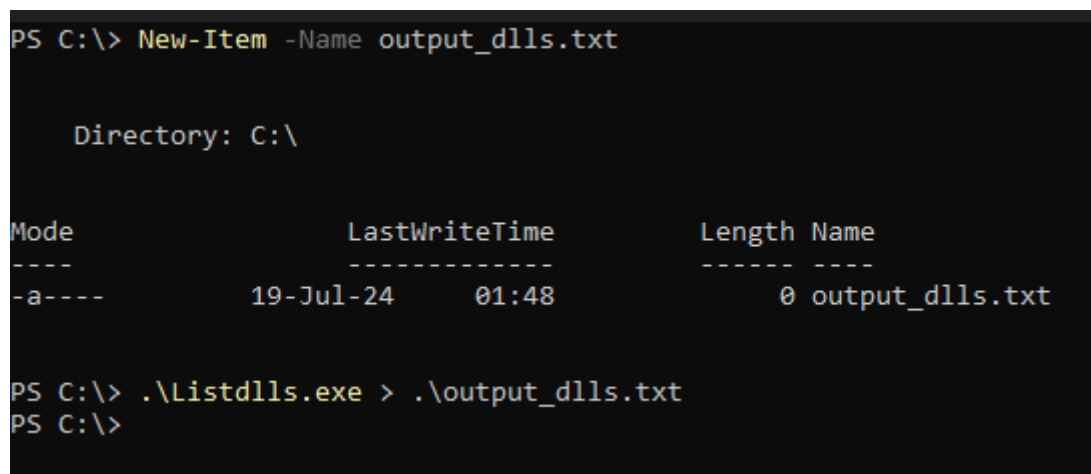


Figure 21: Saving the output from the ListDlls tool.

If the virtual machine used for the analysis is connected to the Internet, the analyst will launch Wireshark and select the interface with Internet access. This selection initiates the recording of the device's network communication.

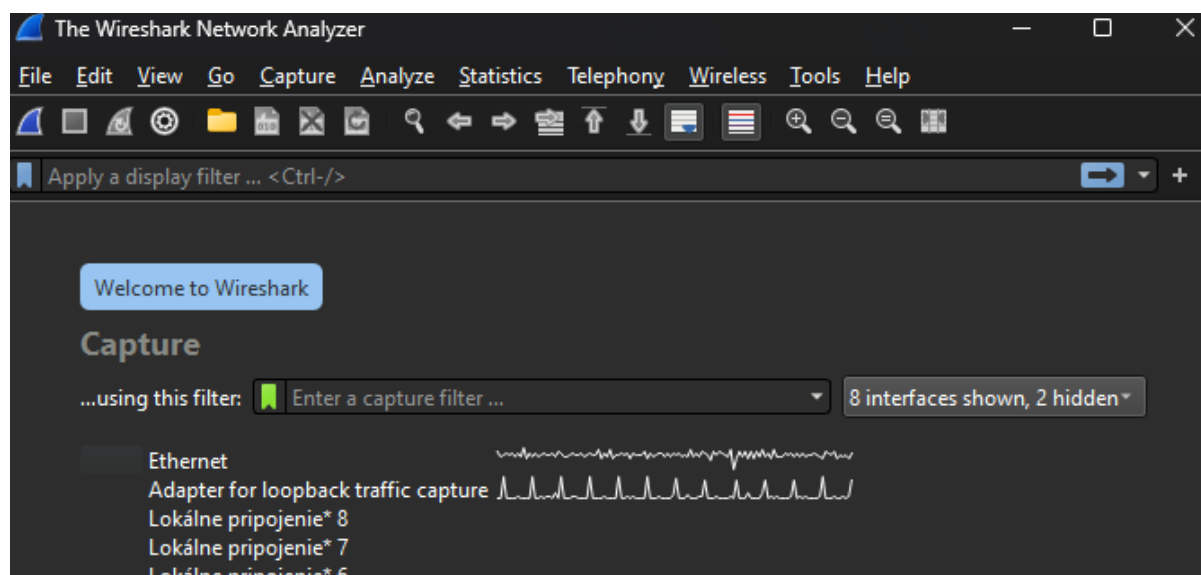


Figure 22: Network communication monitored with WireShark.

At the end of this phase, the malicious code is uploaded to the virtual machine and the process proceeds to the next phase.

Phase 3

In this phase, the previously used tools are executed alongside the malicious code itself. Throughout the process, the virtual machine is monitored for any unusual activity, with particular attention given to the emergence of suspicious processes in Process Explorer.

In some cases, this phase helps determine the type of malicious code. For example, when analyzing ransomware, symptoms such as increased CPU usage, file encryption followed by the creation of a ransom note may be observed within the virtual machine.

At the end of this phase, outputs from Autoruns, Process Explorer, and ListDLLs are saved, following the same procedure as in Phase 2. Additionally, network communication monitoring is stopped, and the output is saved by selecting "File" → "Save As...". Finally, all collected files are securely stored outside the virtual machine.

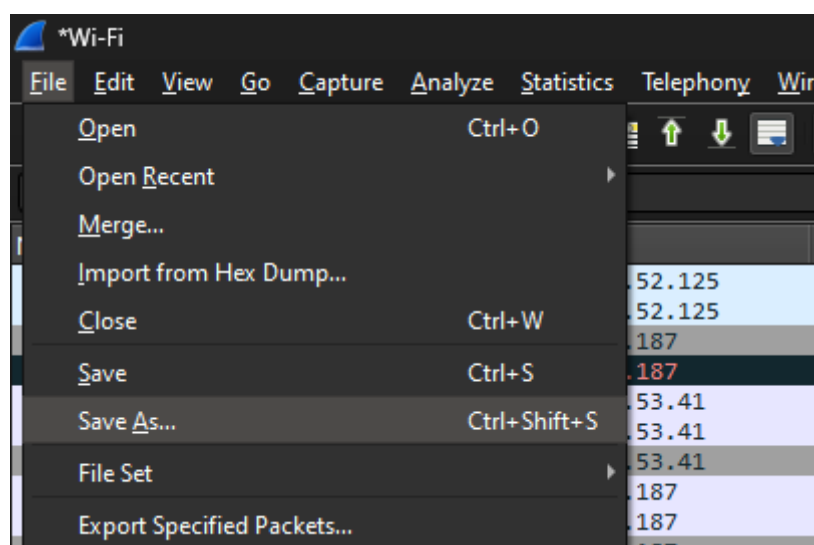


Figure 23: Saving a completed network communication trace using the WireShark tool.

Phase 4

In this phase, the results obtained from the tools and observations of the malicious code's behavior during Phase 3 are analyzed. A comparison is made between the outputs from Phase 3 and those from Phase 2 to identify any possible inconsistencies.

The Autoruns tool outputs are compared with the newly added programs that run at boot/startup, allowing for detection of traces of persistence created by the malicious code.

In Process Explorer, special attention is given to newly started processes that deviate from the standard behavior. The Analysis Team looks for processes that are either unusual and unexpected system processes or others that appear to be normal system processes but reference resources outside the system kernel.

The Wireshark output is reviewed to identify suspicious target IP addresses, which can be further analyzed using databases such as AbuseIPDB. All findings are documented to be later included in the final analysis report.

Finally, the detected leads and system-related information are used to determine the type of malicious code involved. Among the factors that must be considered at this stage is the extent to which each lead affects the outcome of the analysis and what behavior of the malicious code was observed during the dynamic analysis.

Removing malware

This section focuses on the process of removing malware from compromised devices and ranges. The initial step involves identifying an appropriate method for removing malware. In identifying methods, findings from static and dynamic analysis can be used (determining the type, behavior, etc. for specific malware). Additionally, available anti-malware solutions can be leveraged to perform malware removal. Examples of such solutions include:

- Microsoft Defender
- Eset Protect
- Malwerbytes
- Bitdefender

We recommend deleting malicious binary and text files and removing the corresponding values from the registry.

If the previous step has failed, the procedure involves reinstalling standalone systems with a clean OS and wiping infected storages before restoring them from trusted backups. If compromised email accounts were identified during the incident, any login credentials relating to those accounts should be reset.

If a monitoring system is in place, the network should be analyzed for signatures and indicators of compromise to prevent repeat attacks. Any detected signatures, domains, IP addresses and other indicators of compromise will be blocked at the network level. Signatures in anti-malware solutions will also be updated to further block malware.

Documentation of findings / Final incident logging

The findings of the analysis are documented in a formal analysis report using internal malicious code analysis templates. The report includes the timeframe of the analysis, a breakdown of the sequence of steps performed during the analysis along with the data, information, indicators of compromise and type of malicious code analyzed. The report also contains notes made during the static and dynamic analysis of the malicious code.

Additionally, analysts use the incident management tool to record progress by filling in the missing tags (if there any) and reporting any actions taken together with the findings of the analysis. If the incident has the 'on-going' status at the time of report completion, it remains open, with all relevant data and any actions taken being added to the case.

Create detection rule/rules

As the final step of the procedure, we will create detection rules specific to the type of attack encountered. A careful analysis of attack patterns can improve security rules, enabling proactive detection of similar threats in the future.

Key detection rules:

- File and Hash based indicators – generating alerts for downloads of the same named/hash file.
- SIEM and EDR rule adjustment – updating SIEM and EDR systems with refined detection rules.
- Enhancing detection of unauthorized modifications to registry keys, DLL hijacking, etc.
- Rules based on outbound traffic – identifying known IP addresses of malware sources, Command-and-Control IP addresses, etc.

By continuously improving your detection rules and capabilities you can better respond to possible threats and reduce the likelihood of future successful attacks.

References

- [1] Incident Response Playbook [online] [2025-01-30] Available from https://www.incidentresponse.com/minisites/workflows/download/Malware_Outbreak.pdf
- [2] Malware Infection Playbook [online] [2025-01-30] Available from <https://isomer-user-content.by.gov.sg/36/e87c0b79-135a-4f91-9a39-8d59fd081b6b/Playbook-for-Malware-Infection.pdf>
- [3] Malware Security 101 Microsoft [online] [2025-01-30] Available from <https://www.microsoft.com/en-us/security/business/security-101/what-is-malware>
- [4] Malicious Code Cybersecurity 101 CrowdStrike [online] [2025-01-30] Available from <https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/malicious-code/>
- [5] Malicious Code Cybersecurity 101 SentinelOne [online] [2025-01-30] Available from <https://www.sentinelone.com/cybersecurity-101/cybersecurity/malicious-code/>

2.3. Specific Procedure – Ransomware

Identifying the security incident phase

In line with the general incident handling and response guidelines, the process is currently in the Incident Response phase. It is therefore assumed that the previous phases: Incident Recording and Registration and Incident Triage have already been completed (Chapter 1).

According to the classification of security incidents (hereafter referred to as "SI"), ransomware attacks fall under the category of "Information Security Threat" and are distinguished from "Unauthorized Access to Information", "Unauthorized Modification of Information", or "Other".

From a legal perspective, a ransomware attack constitutes a criminal offence (similarly to compromising e-mail accounts) that falls under the category of "Unauthorized access to a computer system".

Quick steps

- Identify affected devices and determine the extent of the problem
- Stop the spread of malware (encryption) by isolating affected devices from the Internet or other subnets
- Block remote access (RDP, VPN)
- Find initial vector (forensic analysis)
- Clean all infected systems
- Check backups:
 - If okay, mitigate vulnerabilities (initial vector)
 - If not, start "from scratch"
- Change all login credentials
- Log the incident

Note: If unauthorized access or data loss is detected, follow the Data Breach guideline (Chapter 2.1).

How to deal with ransomware

The resolution of a ransomware incident is divided into 4 phases:

- **Containment** - This phase focuses on stopping the spread of ransomware within the organization and includes identifying and isolating infected systems to prevent further spread.
- **Gathering** - This phase gathers all available information about the attack such as log collection, network activity analysis, securing evidence, etc.
- **Eradication** - The goal of eradication is to clean up all affected systems. This includes removing malicious code, fixing vulnerabilities, etc.
- **Data, systems and process recovery** - This phase involves restoring the infrastructure to normal operations by verifying backups, applying them or reinstalling.
- **Future prevention and lessons learned** – This final phase focuses on analyzing the incident to identify weaknesses within the organization. It involves conducting a post-incident review to update security measures, increase awareness and improve incident response plans.

Containment

- Identify all affected devices (SHA-1 process name, executables, similar URLs or IP connections on the network):
 - Report the ransom note displayed on the user's screen
 - Find the ransom note in the file system
 - Report the unavailability of user files
 - Identify the large number of edited files on the network
 - After finding a specific device, check if other devices are also communicating with the same IP addresses (such as C2 server)
- Stop the spread of malware (encryption):
 - Shut down the affected systems. Communication with the Crisis Management Team or the organization's management is critical here.
 - With segmented networks, separate affected segments from unaffected ones by disconnecting Internet access.
- Block URLs and IP addresses that have been used for communication. In the case of an email attack, block the email address.
- Block VPN and remote access applications. Prevent attackers from accessing systems through VPNs and RDP applications, SSH, reverse shell, TOR or command and control center.

- Inform users about the unavailability of isolated services and devices.

Gathering

Collect evidence. Record all relevant information including logs, network traffic, screenshots and all communications related to the attack and incident resolution. The following tools may be used for this purpose:

- Disk images– AccessData FTK Imager (<https://accessdata-ftk-imager.software.informer.com/3.1/>)
- Memory dump – Volatility (<https://github.com/volatilityfoundation/volatility>)
- Windows logs – Windows Event viewer
- Eric Zimmerman Tools (<https://ericzimmerman.github.io/#!index.md>)

Note: Tools used for evidence collection may vary depending on the specific characteristics of individual systems. It is essential to choose tools compatible with the affected systems and capable of accurately capturing all relevant data.

Conduct forensic analysis of the collected data. The outcome of the analysis should serve as input for incident eradication.

Eradication

- Identify ways to remove malware.
- Perform the removal process using appropriate anti-malware tools:
 - Microsoft Defender
 - ESET PROTECT
 - Malwarbytes
 - Bitdefender
- Delete malicious binaries or corresponding registry values.
- Reinstall standalone systems with a clean operating system before restoring data from trusted backups.
- Change the details of any compromised accounts. This includes, at a minimum, changing login credentials (login and password).
- Monitor signatures and indicators of compromise on the network using system monitoring (if available) to prevent repeated attacks.
- Block detected signatures, domains, IP addresses, and other indicators of compromise on the network.
- Update signatures in anti-malware solutions to ensure malware remains blocked.

Data, systems and process recovery

- Check backup systems for malware.
- If backups are infected, do not use them. This requires reinstallation of the systems ("start from the scratch").
- If the backups have not been compromised, it is necessary to detect the presence of vulnerabilities or the initial vector of the attacker. If found, they need to be mitigated. Otherwise, the backups can be used safely.
- Change all login credentials.

Future prevention and lessons learned

- Conduct a post-incident review to analyze weaknesses and document improvements.
- Update security measures based on lessons learnt.
- Train employees to raise security awareness.
- Revise and update the incident response plan and remove any omissions.

- Revise and update backups policy.
- Share IOCs to relevant platforms.

The goal of this step is to restore all essential systems and resume normal system operations. To safely deploy and restore the system and data from backups, it is necessary to check the backup systems (if available). If the backup system has also been compromised in the attack and no data is available for restoration or if it is suspected to have been infected, a clean installation must be performed. If the backups for restoring systems and data are unaffected, you can proceed with restoration. Also, all passwords must be changed during this step.

When restoring systems and data, it is necessary to contact the administrators of the systems in question.

Incident logging

Incidents are registered in the incident management tool. According to the classification of security incidents, ransomware attacks are categorized as "Information Security Threats", with distinction made between "Unauthorized Access to Information", "Unauthorized Modification of Information", and "Other". Adding tags can help to indicate the type of incident (ransomware) and provide additional information such as IP addresses, affected departments, etc. Basic details about the receipt and handling of the incident should be included in the description.

Once these steps have been taken and the incident has been resolved, any additional findings should be documented before closing the incident.

References

- [1] #StopRansomware guide: CISA (no date) Cybersecurity and Infrastructure Security Agency CISA. Available at: <https://www.cisa.gov/stopransomware/ransomware-guide> (Accessed: 5 August 2024).
- [2] National Cyber Security Centrum (2022) Ransomware incident response plan, Publication | National Cyber Security Centre. Available at: <https://english.ncsc.nl/publications/publications/2022/augustus/2/incident-response-plan-ransomware> (Accessed: 5 August 2024).
- [3] Home (no date) The No More Ransom Project. Available at: <https://www.nomoreransom.org/> (Accessed: 5 August 2024).
- [4] (No date) GSVSOC_CIRT-Playbook-battle-cards/GSPBC-1000 - impact - data ... Available at: https://github.com/guardsight/gsvsoc_cirt-playbook-battle-cards/blob/master/GSPBC-1000-Impact-Data-Encrypted-For-Impact-Ransomware.pdf (Accessed: 5 August 2024).

2.4. Specific procedure – Social engineering

Identifying the security incident phase

The most typical methods of detecting social engineering are user reports and the direct receipt of an unwanted message by a member of the security team.

It is assumed that the "Incident Triage" and "Incident Recording and Registration" steps have already been completed. This means the incident has been evaluated as a security incident, recorded and assigned an investigator. The process is now in the "Incident Response" phase.

Determination of the incident category

Social engineering involves a variety of techniques used by attackers to manipulate individuals into divulging sensitive information or granting unauthorized access, which include phishing, baiting, vishing, smishing, and more. Physical techniques like tailgating and dumpster diving should not be overlooked either.

This section focuses specifically on phishing (delivered via e-mail), vishing and smishing, providing details of how to respond to such incidents, including the analysis of links, attachments, e-mail headers and phone numbers. While many other attack types are similar to phishing, differing primarily in delivery method and intent, incident handling and response remain largely the same.

The reported incident must be categorized based on the initial findings. While it may fall under more than one category, the incident should be recorded according to the taxonomy by choosing the category based on its most prominent feature (additional categories can be included in the incident description). A list of possible categories and subcategories is provided in Table 11. Here all possible categories for e-mail incidents should be included, based on the taxonomy chosen by your organization. Providing a description will facilitate decision-making.

Table 11 - Incident categories of social engineering

Category	Subcategory	Description
Information gathering	Social engineering	Scams that exploit trust and relationships between people to obtain sensitive information.
Unwanted content	Spam	. Message delivered to the user without their consent, and it is one of a larger set of messages sharing the same content.
	Violence	Unwanted content promoting violence.
	[add your organization's subcategories]	[add description]
Fraud	Identity impersonation	Attacks in which one entity illegitimately impersonates another person for benefit.
	Phishing	Fraudulent emails, messages, or websites designed to trick users into revealing sensitive information.
	[add your organization's subcategories]	[add description]

Phishing e-mails

Summary of actions

- Message content control.

- Message sender verification (including certificate verification).
- Verify links and attachments.
- Secure and analyze e-mail headers.
- Categorize e-mails.
- Contact the reporting party with response.
- Notify the sending organization.
- Document and close the incident.

Phishing e-mail solution procedure

To address phishing, additional information should normally be obtained through the analysis of the suspicious e-mail message. Readily available information should be checked in the first place. This includes details obtained from the forwarded fraudulent e-mail such as contents, sender's address, URL links and attachments. If the result is inconclusive, the analysis proceeds by retrieving the header of the original message and analyzing it according to [Chapter 4.2.3 Email header analysis](#). A summary of the entire resolution process is presented in Figure 24.

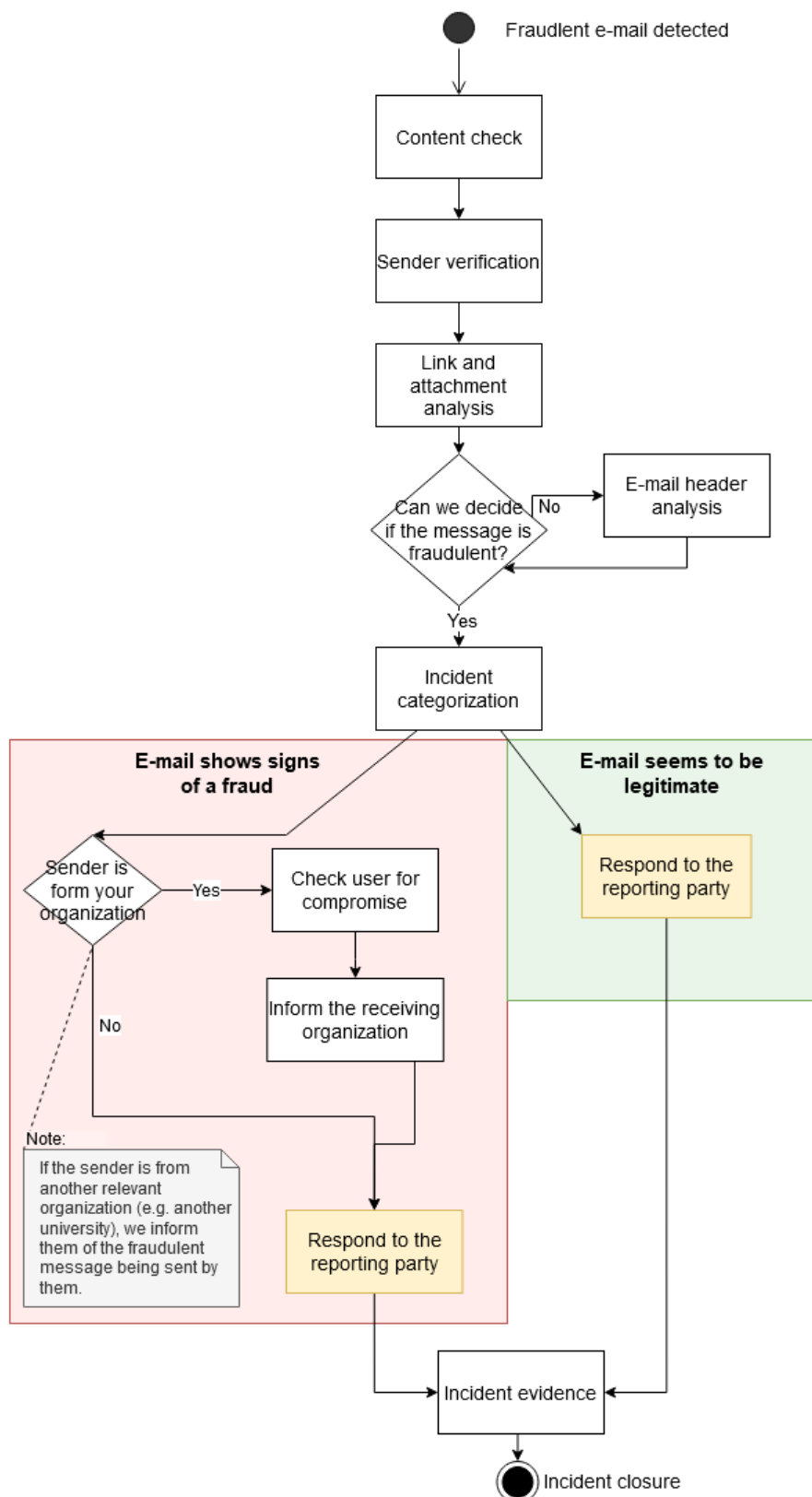


Figure 24: E-mail incident resolution process

Basics of e-mail analysis

The basic investigation focuses on the body of the forwarded message.

Content verification

This primarily concerns the style, grammar, and intent behind the message. The following aspects should be considered:

- **Grammar and style** – Any errors may indicate use of a translator (note, however, that legitimate users make mistakes too). With the rise of ChatGPT and other AI-based tools, grammar and style can now be correct, so caution is needed even when the content appears polished.
- **Message intent** - Messages designed to prompt users to share their personal data under false pretenses – such as package delivery, e-mail account deactivation or prize notification are most likely phishing messages. The content, which lures the user to open the attached file by claiming that it is e.g. a paycheck, an invoice, an incriminating video, is also very likely to be an indicator of malicious intent behind the message.
- **Address and signature** - The form of address is also important, as attackers often use generic salutations (e.g. “Dear User”, “Dear Sir/Madam”) or simply the recipient's email address. At the bottom of the message they often include a generic signature like “IT Team” or “Admin” rather than the name of a specific person. While they generally avoid signing the message with a real name, this possibility cannot be entirely ruled out.

Message sender verification

Regarding the sender, it is essential to verify the sender's e-mail address by checking whether the domain corresponds to the organization the person claims to represent. Examples of suspicious e-mail addresses include: apache@beedesign-lamp02.dedie.ate.info, uzb51f3@bffcrcvbb.ewys.sbs, metamask@2fa.io etc.

It is advisable to verify whether the e-mail address is deliverable. This can be done using a variety of online tools (e.g. [Email Dossier](#), [Verifalia](#)), which attempt to connect to the corresponding e-mail server and act as if delivering a message. There are also tools assessing e-mail reputation ([Email Reputation Checker](#)), which should also be considered.

If the e-mail address is undeliverable, it is almost certainly associated with a malicious message. However, even if the message passes all validation checks, it still cannot be considered safe, and further analysis is required.

E-mail certificate verification

Some organizations require that special certificates be used when signing e-mail messages. If a reported e-mail is signed with a certificate, check its validity period and make sure the issuer is trusted and that the certificate links to a trusted root certification authority (CA). Some certificates that seem legitimate may have been revoked by the CA. View the Certificate Revocation List (CRL) to double-check on the certificate.

Analysis of links and attachments

If an e-mail contains any links to websites, these need to be thoroughly examined. Be careful not to click on the links directly. Instead, copy the URL to the clipboard. Check the text for names or e-mails of legitimate recipients before submitting the link to any analytical tools. Make sure to replace any such emails or names with more generic placeholders, such as “john.doe@examplecompany.com”. Now, you can analyze the link using your preferred URL analysis tool (for example, [VirusTotal](#), [URLScan](#), [PhishTank](#)).

The first step in the analysis of an attachment involves checking the file hash. We recommended using a virtual machine for this purpose. Download the attachment and generate the file hash using a command line tool (for Linux, use: `sha256sum fileName.extension`). The hash can be checked with various malware detection tools ([VirusTotal](#), [AlienVault](#), [HybridAnalysis](#)). Note that even if the tool does not recognize the hash, the attachment is not necessarily clean. If the header analysis performed in the next step remains inconclusive, the attachment can be further analyzed according to preferred malware analysis procedures.

Note: Do not upload the attachment to any online tools, unless you are certain it is not legitimate. Otherwise, you may expose confidential or sensitive information.

Some reported messages may not contain any malicious attachments or harmful URLs. They can, however, be classified as spam (unsolicited emails including chain messages and advertisements) and contain violent content, threats, fake news etc. Additionally, some attackers use innocent-looking e-mails to identify active accounts, with the actual attack occurring later in an ongoing conversation.

E-mail header analysis

Header analysis is usually performed when all previous steps remain inconclusive. It normally helps to determine whether the message is legitimate or it exhibits malicious characteristics. However, to conduct header analysis, the original message is required (the user should often be guided on how to send the original email properly, or if possible, it can be obtained directly on-site). A forward of the original message will be sufficient as it has completely different headers.

Before using any online tools, the original message should be opened in text format and anonymized by replacing any personal or company-specific information with generic placeholders. The header can also be viewed in notepad, but we strongly advise using parsing tools, such as MXToolbox, to view the date with improved clarity and readability.

E-mail headers contain various attributes depending on the e-mail client used and they appear without any specific order. The e-mail header section always ends after the last "X" attribute. In the example illustrated below (Figure 25), the header ends after the „X-Microsoft-Antispam-Message-Info" attribute.

```
X-Microsoft-Antispam: BCL:0;
X-Microsoft-Antispam-Message-Info:
=?us-ascii?Q?wfflpXL8sLmMiyzAs21LQdvBymKR511QeKEcTz4FYmeIjeFAHQ8MLiTaF2z?=?us-ascii?Q?f8qX1ts5I+mHEpXlksS4CdOEyQZjidHRLBKjdtWNoAnaWdNEHnwWm0RJXMfv?=?us-ascii?Q?godisWQ53uGKVlmpxTKSaN/nrwK2M+fMcZcETNatK71k8bmdmX0qOP51A1on?=?us-ascii?Q?byjejet5LuW6XaCtUQv1nXPsgFgybMB3yaK6oALnNi+0dmsvJrqWofzwTcHP?=?us-ascii?Q?egGhpo1eXa78s7hcmdoWwk5EcAFwf04bh2XGdQu3jY1JBpqWAL5JNorQqtP2?=?us-ascii?Q?qvZDXCwPaJ/8uWBH9DdTAYZqBDlms7POEXZj2LBzfpCWBr0v6HFw+24LH7sn?=?us-ascii?Q?ZzTXpeDAzYHqqk/4E/NnwLJL1nvffv4v8XUGpSruaum/e4Wic6HkL8TQp3k?=?us-ascii?Q?Vbk3mVZoTj04Hh0fdgHjrXbdsnyorY/h6tKGHfNrQAF799khekcV12zJ6Cib?=?us-ascii?Q?R1cRNMBj1tMnANzHU7zbd7oU001i2pQA+6/SKpGwNUUWImNOADpAa4//jw7f?=?us-ascii?Q?bTWmaCYmVp3YhaJ00oSu6ttmfCEOV32HaaA3rFuBo+UCoBOBAyuk5H11Wba?=?us-ascii?Q?FAus/HOZ1b1rUL/yRo87tiJVaphh6vRIgWvxfFeTCG7u0Y9oL1NQeG73HNP1?=?us-ascii?Q?aiUQc1Yk5eoyRdYEFev/8ZgmHrXlM0V224s5bK0wULGXTBfwkG5g7c2AKRL?=?us-ascii?Q?CgxDc8/Po/J8EDj8KSJS19zEuRs/tVAi1w3MdFVYmaEZH/9DTxjXgocpV50A?=?us-ascii?Q?sKUPI0ek1NrJg2HD4frn+niZA5j/mD//TMQ2qswV6bvEGjDphMtnFuddvLld?=?us-ascii?Q?4oTD0uYzja3Sw4uf7MGLdwcw+tUB+1UijsCcac3WnEqgSfJGYyQUgf/8jr8E?=?us-ascii?Q?0IN+ngP4+SojbWR4M9RgKxAdxYw7z0iuVjS0B2Zads27Vq71+cAZ+oSYSw?=?us-ascii?Q?=3D=3D=?
END
<html><head>
<meta name="GENERATOR" content="MSHTML 11.00.10570.1001">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
</head>
```

Figure 25: End of an e-mail header

The key attributes to observe are:

The SOCCER project, funded under Grant Agreement No. 101128073, is supported by the European Cybersecurity Competence Center (ECCC).

- **Receive-Path:** how the email was forwarded,
- **To:** message recipient (Figure 26),
- **From:** message sender (Figure 26),
- **Return-Path:** typically the same as the sender address (Figure 26),
- **Reply-To:** e-mail address to which a reply should be sent, normally the same as the sender's address. However, even legitimate senders may configure these fields differently.
- **Received fields:** The „received“ fields always appear at the top and capture the servers through which the e-mail has passed. It is important to inspect the entire chain of the received fields rather than rely on the first one alone, as attackers can manipulate headers or insert fake entries. **IP address** can be found in the last „**Received**“ field, in „**Authentication-Results**“ and „**X-Forefront-Antispam-Report**“ attributes, (see Figure 4). They can be further analyzed using online tools, such as [ip-api](#), [IPTrackerOnline](#), [IPInfo](#). Note the country and domain name. Spam level on [Talos Intelligence](#) is also relevant as well as reports of the IP by other users ([AbuseIPDB](#)).
- **X-Microsoft-Antispam:** The BLC value indicates the extent to which the message is reported as unsolicited. It ranges from 0 (no complaints about the sender) to 9 (the sender frequently sends bulk messages and receives numerous complaints)
- **Authentication-Results:** a form of message sender authentication which helps to determine the authenticity of email addresses, signatures and other elements. Not every email server supports these authentication methods and therefore they may not always be available. On the other hand, even successful authentication does not necessarily guarantee that the email is legitimate. Attackers can use the same platform (e.g., O365) as the target and configure SPF, DKIM, and DMARC correctly, making phishing e-mails appear legitimate to successfully bypass anti-spam filters. SPF, DKIM, DMARC are included in this attribute (description follows below, for details and issues click [here](#)).

To	meno.priezvisko@upjs.sk
Subject	! Review Error Report {3 Blocked Emails}
X-PHP-Script	mbegroupinc.com/mic.php for 62.217.191.202
X-PHP-Originating-Script	1090:mic.php
Date	Tue, 23 May 2023 10:59:57 +0000
From	"upjs.sk Server Administration" <support@mbegroupinc.com>
Message-ID	<04920d7a38b2a0f3ff6edfeaadc97310@mbegroupinc.com>
MIME-Version	1.0
Content-Type	multipart/alternative; boundary="b1_04920d7a38b2a0f3ff6edfeaadc97310"
Content-Transfer-Encoding	8bit
X-AntiAbuse	This header was added to track abuse, please include it with any abuse report
X-Get-Message-Sender-Via	host.rmacanada.com: authenticated_id: mbegroupinc/from_h
X-Authenticated-Sender	host.rmacanada.com: support@mbegroupinc.com
X-Source	
X-Source-Args	
X-Source-Dir	
Return-Path	support@mbegroupinc.com
X-EOPAttributedMessage	0

Figure 26: Example of header attributes (To, From, Subject, Return-Path)

Authentication-Results	spf=pass (sender IP is 96.125.173.221) smtp.mailfrom=mbegroupinc.com; dki m=none (message not signed) header.d=none; dmarc=permerror action=none header.from=mbegroupinc.com; compauth=pass reason=105
X-Forefront-Antispam-Report	CIP:96.125.173.221; CTRY:US; LANG:en; SCL:9; SRV:; IPV:NLI; SFV:SPM; H: host.rmacanada.com; PTR:host.rmacanada.com; CAT:HPHISH; SFS:(13230028)(84120400003)(7302899009)(4311899009)(451199021)(40265005)(26005)(6916009)(24736004)(33964004)(34756004)(108616005)(58800400005)(1096003)(9786002)(166002)(9746002)(5660300002)(8676002)(83380400001)(356005)(2616005)(36756003)(336012)(86362001)(22186003)(7596003)(7636003)(15650500001)(19588885015); DIR:INB;

Figure 27: Location of IP addresses

- **SPF:** a form of verification whether the sender really belongs to a given domain. SPF can take several values, as shown in Table 12.

Table 12 - Description of possible SPF values

Status	Description
pass	the client can send emails with a given domain
fail	verification failed
softfail	the sender is not allowed to send but is in transition

neutral	the tool neither permits nor refuses to send the message
none	SPF is not defined
temperror	temporary error, this status may change over time
permerror	SPF malfunction

- **DKIM:** digitally signs important elements of the message (e.g. the From attribute) and stores the signature in the header. The destination server verifies that the signed fields have not been altered. The verification results are described in Table 13.

Table 13 - DKIM verification result descriptions

Status	Description
pass	DKIM Verification is successful
fail	Verification failed. The reason for the failure is also given, e.g. the signature has not been not verified.
none	The message has not been signed

- **DMARC:** verifies whether SPF and DKIM (domains) are in agreement. The DMARC determines what should happen on the target server to a message with a failed DMARC status. Descriptions of all DMARC statuses are provided in Table 14.

Table 14 - Description of DMARC verification results

Status	Description
pass	Verification was successful
fail	Verification was unsuccessful
bestguesspass	No DMARC record. Otherwise, the verification would most likely be successful.
none	DMARC record does not exist

- **X-Forefront-Antispam-Report:** contains additional information in the *key:value* form. The relevant values are: "CAT", "CIP", "SCL" and "SFV". CAT is the category of the protection policy that has been applied to the message. CIP is the IP address of the sender. SCL indicates the likelihood of the message being spam, and SFV describes what happened to the message and why (message marked as spam, bulk email, etc.). A full description of these attribute values can be found [here](#).

Response and incident evidence

Once a reported incident has been investigated, a response must be sent to the reporting person. A template response has been appended to this chapter as Annexes.

If fraudulent messages are observed to originate from the domain of a relevant organization (such as a university or school), they should be informed of this fact (see Annex B for a sample notification message).

If fraudulent messages have been sent out by your organization, notify the receiving organization of this fact. Details of the administrators of the domain or the organization affected can be traced via [Whois Domaintools](#) or by checking contact information on their official websites. Additionally, it is recommended that you verify whether any user accounts within your organization have been compromised.

If the message is determined to be malicious, consider the following aspects to minimize further damage: block IoCs (indicators of compromise like e-mail address, IP address, hash, URL); request a takedown of malicious URLs; scan the recipient's device for potential malware; update e-mail filtering rules where possible.

The incident should be registered in your incident tracking system. In this section, provide brief instructions on how to log the incident, e.g. by specifying which template to use.

Once these steps have been completed and, where applicable, the sender's organization has confirmed resolution of the issue or no further fraudulent messages have been received, the incident can be closed as resolved.

Smishing

Summary of actions

- Analyze SMS content
- Verify links
- Verification the phone number
- Respond to the reporting party
- Document and close the incident.

SMS analysis

The analysis process is similar to handling phishing incidents. Start by checking the content of the message. Look for the same indicators as in phishing e-mails, except that text messages typically lack a greeting or a signature.

The next step is to **verify any links** found in the text message, following the same steps as in chapter 3.2. This will help you determine, whether the message is fraudulent. However, if it is not clear, the sender's phone number can be analyzed too.

Phone number analysis

- Check the area code and country of origin (e.g. on <https://areaphonecodes.com/>).
- Look up the phone number in your country's local database of phone numbers. You may use <https://whocallsme.com/> to check reputation of the number and view reports from other users.
- Use various OSINT tools, like <https://epieos.com/>, to obtain more information.

Incident closure

After completing all the required steps, respond to the reporting party and inform them about the outcome of the investigation – indicate whether the message is or is not fraudulent.

Record the incident in the ticketing system along with all the completed actions. In this section, provide brief instructions on how to log the incident, e.g. by specifying which template to use.

Now, the incident can be closed as resolved.

Vishing

One of the most interactive techniques, **vishing** is a form of social engineering conducted through a phone call. As with the previous techniques, the goal here is to persuade the victim to share sensitive information or download malicious files. A typical example is a fraudulent call from IT support, in which the victim is made to believe, there is a technical issue with their device, which can be fixed by following specific instructions recommended by the caller. The instructions normally enable the attacker to launch remote desktop protocol and obtain full access to the system. Another example is a false bank representative, requesting the victim's banking details to "confirm" their identity before explaining the reason for the call.

Vishing Incident Response

After the phone call has ended, the possibilities of investigation are limited, with the recipient of the suspicious call being the main source of information. Here are some steps to consider following a possible vishing incident:

- **Gather information** about the caller's claimed identity, the caller's language preferences and the nature of the request (what = information they sought, what actions they requested the victim to take).
- Ask the victim if any sensitive information was shared with the caller. If so, **reset their password** to your company's systems. Advise them to **stay alert** for any follow-up scam, closely watch their accounts, and report any unusual activity.
- Suggest blocking the phone number while explaining that this will not prevent similar future attacks undertaken from other numbers.
- Ask the victim if they performed any actions on their device. If so, find out what the actions were and respond accordingly. Afterward, watch the device for any unusual activity.
- Analyze the phone number, as described in Chapter 4.2.
- Check if the phone number is listed as an official contact of the claimed organization (if known).

These steps should help you determine whether the incident qualifies as vishing, and if so, what the extent of compromise is.

Incident closure

Communicate the findings to the reporting party/victim. Provide guidance on how to stay more alert in the future, indicate the areas that require improvement, and outline the actions to take if any sensitive information has been shared with the caller.

Consider alerting the employees to an ongoing vishing campaign and share tips on how to maximize protection.

Record the incident in the ticketing system along with all the findings and actions taken. In this section, you can provide brief instructions on how to log the incident indicating, for example, which template to use.

Annexes

Model reply to the reporting party

Note: The template contains several variants, which can be combined, modified, and supplemented at will to suit the specific situation.

Dear John Doe,

Thank you for reporting the e-mail. (This message has been reported by multiple employees.) The message in question has been classified as fraudulent and should not be responded to.

The attacker has made an impersonation attempt. However, we have established that the email address of the actual sender is different. / This is an email designed to identify active user accounts. If you do not know the user, we recommend that you do not respond to this email. / This is a fraudulent email with a malicious link to a form requesting your login details. We have already reported this link, and it is currently inactive. If you have completed the form, we recommend that you change your account password immediately.

If we can be of any further assistance, please do not hesitate to contact us.

Best regards,

Dear John Doe,

Thank you for reporting the e-mail. After performing a thorough analysis of all the links, attachments and details of the message, we consider it to be safe.

We always encourage caution when handling unexpected or unfamiliar messages, and we greatly appreciate the fact that you have taken action to verify its legitimacy.

Should you have any further concerns or encounter any suspicious activity, do not hesitate to contact us.

Best regards,

Informing the sender's organization

Subject: TLP:AMBER Phishing Campaign Alert

TLP:AMBER

Good morning/afternoon,

We would like to alert you to an ongoing phishing campaign from your organization.

On [date] we have noticed phishing emails being sent from your organization, specifically from the accounts [insert sender accounts here]. The emails contain a malicious link [insert e.g. URLscan of malicious link here].

We have enclosed an archive containing the phishing email. the attachment has been secured with the password: "infected".

We would also like to ask you to provide information on the actions you have taken to resolve this problem.

Thank you for your cooperation.

Sincerely

TLP:AMBER

2.5. Specific Procedure – Linux server compromise

Identifying the security incident phase

At this stage the "Incident Triage" and "Incident Recording and Registration" steps have already been executed. This means the incident has been classified as a security incident, it has been recorded and assigned to an investigator. The process now advances to the "Incident Response" phase.

The most common methods of detecting a server compromise involve notification emailed to the development team regarding unusual activity on another organization's device, or direct detection of such activity by the team itself. The person responsible for operating the device on which the server is running may also identify the issue. The development team will then report the incident to the security team for further investigation.

Incidents of this kind can fall under various categories, which include "Malicious Code", - further subcategorized into Virus, Worm, Trojan, Spyware, Dialler. Intrusion Attempts come with subcategories: Exploiting of known Vulnerabilities, Login attempts a New attack signature. Additional categories include Intrusions (Privileged Account Compromise, Unprivileged Account Compromise and Bot) Availability (DoS, DDoS, Sabotage and Other) and Information Content Security can be further divided into (Unauthorized access to information and Unauthorized modification of information). The methodology of categorizing incidents is described in detail in Chapter 1.3.

Summary of actions

- Isolate the system
- Check active processes
- Analyze running processes
- Analyze network connections
- Detect system changes
- Respond to the reporting party
- Notify the sending organization
- Document and close the incident

Addressing Linux server compromise

Several key steps should be taken when addressing a Linux server compromise. If the incident is ongoing, priority should be given to rapid response in order to halt malicious activities and minimize further damage. If the incident has already ended, the focus

should be on a thorough analysis of its causes and consequences. Incident handling should include system isolation, preservation of evidence, analysis of logs and data, identification and removal of malicious software, system restoration from backups, and implementation of additional security measures. Communication with all the involved parties is essential, as is preparation of a final report with recommendations for improving security measures.

System containment

Upon discovering a Linux server compromise, it is essential to immediately restrict the system in order to minimize damage and prevent the spread of the attack. The specific steps depend on the system type, the criticality of the services running on it and the particular circumstances of the compromise. Before taking action, it is important to consider the potential impact on service availability and to consult relevant teams. We advise minimizing interference with the compromised system to avoid destroying important evidence needed for forensic analysis. The incident should be communicated through separate and secure channels to prevent the attacker from gaining further insight into your response measures. Prepare the team and necessary resources for the analysis and eradication phase to address the incident efficiently and swiftly.

Here are key steps to consider:

- **Immediate Disconnection from the Network:** In the event of a compromise, immediately disconnect the server from the network. If possible, physically unplug the server to instantly halt any external communication or shut down the port on the affected switch. If physical disconnection is not feasible, block network connections using a firewall or VLAN isolation. Before disconnecting the server from the network we recommend recording information about open ports and connected sessions (see Chapter 3.2 for detailed instructions).

Command: `ifconfig interface down`

This command disconnects the network interface, interrupting all network communication.

- **Isolation of Compromised Accounts:** Temporarily disable all user accounts that may have been compromised in order to prevent misuse. Next, change the passwords to all critical accounts, especially administrative ones, to reduce the risk of further access by the attacker.
- **Monitoring Active Processes:** Review the list of running processes and terminate all those that appear suspicious to minimize damage caused by malicious software. Use tools such as `ps`, `top`, `htop`, `netstat`, `lsof` and `strace` to identify and monitor suspicious activities.

Command: `kill -9 [PID]`

Forcefully terminates the process with a specified PID. Use this command to stop malicious processes identified during the analysis.

Command: `chmod 000 /bin/bash`

This command changes the permissions of `/bin/bash`, preventing it from being executed. However, it only affects the bash shell and does not prevent the use of other shells that may be available on the system, such as `/bin/sh` or `/usr/bin/zsh`. To fully restrict shell access, you need to address all available shells or implement broader user restrictions.

- **Preservation of Evidence:** Retain all relevant logs for further analysis, such as `syslog`, `auth.log` and `dmesg`. Start by creating a RAM dump to capture system

memory contents, as this provides critical evidence that may not be available later. Create a complete image of the disk and memory as evidence for forensic analysis. Additionally, save information about open ports and connected sessions (using “netstat” or “ss”), and collect network flow data (using NetFlow or similar tools) to capture details of ongoing or recent connections that may be crucial for the investigation.

- **Basic System Integrity Check:** Examine system files and binaries for any modifications or suspicious files using tools like “rpm -Va for RPM-based systems” or “debsums for DEB-based systems”. Compare the hash sums of system files with their original values to identify any unauthorized modifications.

Command: `chattr +i /etc/passwd /etc/shadow /etc/group /etc/gshadow`

Sets the “immutable” attribute on critical system files, preventing their modifications.

- **Temporary Measures to Mitigate Damage:** Implement stricter firewall rules to block suspicious connections and reduce the risk of further attacks. Enable or adjust existing security mechanisms like SELinux or AppArmor to enhance system security.

Command: `iptables -P INPUT DROP a iptables -A INPUT -p tcp --dport 22 -s <trusted_ip_address> -j ACCEPT`

Sets the firewall policy to block all incoming traffic except SSH (port 22), restricting access to the server for remote management to only one specified trusted IP address.

Command: `mount -o remount,ro /`

remounts the root directory in read-only mode, preventing any further changes to the file system.

- **Termination of All Network Connections:** Identify and terminate all existing network connections to prevent further spread of the compromise within the network. Before disconnecting all the devices from the network, it is advised to record all the connections (see Chapter 3.2 for further instructions).

Command: `netstat -anop | grep ESTABLISHED | awk '{print $7}' | cut -d '/' -f1 | xargs kill -9`

This command identifies and forcefully terminates all processes associated with established network connections, effectively cutting off all active communications that may be part of the compromise.

Digital evidence gathering

In this section, we will focus on gathering diagnostic data from the compromised Linux server. The first step is to collect basic system information, which will provide an overview of its current state and configuration. The following commands will help in gathering the data:

- **Creating RAM Dump**

Command: `dd if=/dev/mem of=/final_destination/memdump`

The `dd` command creates a RAM dump by reading memory contents. This is essential for forensic analysis as it captures volatile data, including running processes, active

connections and other critical information that might not be available after a system reboot.

▪ **Creating a System Image for Forensic Analysis**

In certain cases, it may be necessary to preserve the system state for a more in-depth forensic analysis. One effective approach is to create an image of the system using the built-in `dd` command or the enhanced `dc3dd` tool, if available. For instance:

Command: `dd if=/dev/sda of=/path/to/output/image.dd bs=4M status=progress`

Command: `dc3dd if=/dev/sda of=/path/to/output/image.dd hash=sha256 log=/path/to/logfile.log`

▪ **Gathering Kernel Information:**

Command: `uname -a > /final_destination/uname`

The `uname -a` command displays detailed information about the system's kernel, including the hostname, kernel version, compilation date, architecture, and other details. This information helps identify the type and version of the operating system, which is important for understanding the environment in which the server is running.

▪ **Obtaining the hostname:**

Command: `hostname > /final_destination/hostname`

The `hostname` command displays the current hostname, which enables unique identification of a specific server, especially when managing multiple systems.

▪ **Obtaining IP configuration:**

Command: `ip a > /final_destination/ip`

The `ip a` command displays detailed information about the system's kernel, including the hostname, kernel version, compilation date, architecture, and other details. This information helps identify the type and version of the operating system, which is important for understanding the environment in which the server is running.

▪ **Obtaining the list of users:**

Command: `cat /etc/passwd > /final_destination/passwd`

The `cat /etc/passwd` command displays the contents of the `/etc/passwd` file, which contains a list of all system users. This file is crucial for understanding who has access to the system and whether any new, potentially unauthorized users, have been added.

▪ **Obtaining the list of groups:**

Command: `cat /etc/group > /final_destination/group`

The `cat /etc/group` command displays the contents of the `/etc/group` file, which contains a list of all groups and their members on the system. This file is useful for analyzing group permissions and access rights.

▪ **Obtaining system version information:**

Command: `cat /etc/system-release > /final_destination/system-release`

The `cat /etc/system-release` command displays the contents of the file that contains information about the operating system version. This is important for identifying the specific version of the OS running on the server.

Processes

When investigating a compromised Linux server, it is crucial to examine running processes to identify suspicious activity. The following commands will help gather the necessary information:

- **All Running Processes:**

Command: `ps -auxwf > /final_destination/ps_aux`

This command displays detailed information about all running processes on the system, including users who started each process, CPU and memory usage, PID (Process ID), and the process tree structure. The output is saved to the file `"/root/fa/ps_aux"`.

- **Search for processes related to a specific service:**

Command: `ps -auxwf | grep [service]`

This command searches for all processes that contain the string `[service]` in the description. It is useful for identifying processes associated with a particular application or service.

- **Iteration through PIDs from the processID file and information about network connections:**

Command: `for pid in $(cat processID); do echo "===" $pid; ps faux | netstat -lpeanut | grep $pid`

This command iterates through all PIDs stored in the `processID` file. For each PID, it displays information about the process and network connections using the `ps faux` and `netstat -lpeanut` commands respectively, filtering the results by PID.

- **Obtaining a list of open files for any PID:**

Command: `lsof -p [PID] > /final_destination/[PID]_lsof`

This command will save a list of open files for a process with a given PID to the file `/root/fa/[PID]_lsof`, where `[PID]` is replaced with a specific process PID.

- **Obtaining process information from the /proc directory:**

Command: `/proc/[PID]`

Navigate to the `/proc/[PID]` directory and replace `[PID]` with a specific process PID to obtain information about that process.

- **Calculation of SHA256 hash of process executable file:**

Command: `sha256sum exe`

Network connections

In this section, we will focus on analyzing current and detailed network connections, which is crucial for identification of potentially suspicious activity on the server.

- **Current network connections**

Command: `netstat -anop`

This command displays all active network connections along with information about open ports, protocols, connection states, and the PIDs of the processes that use them.

- **Detailed information about network connections**

Command: `netstat -lpeanut > /final_destination/netstat`

This command will display detailed information about all network connections, including open ports, protocols, PIDs, and the names of processes using them, and save the output to the “/root/fa/netstat” file.

Investigation of differences

When gathering digital evidence from a compromised server, it is important to identify and analyze differences between the current state of the system and known good states or backups. The following commands will assist in checking for such differences:

- **Creating a list of files and directories:**

Command: `ll /path/to/check/`

This command displays a detailed list of files and directories in a specified location, providing an overview of the current state of the file system.

- **Directory and file comparison:**

Command: `diff --recursive /path/to/check /path/to/known-good/`

This command will recursively compare all files and subdirectories between two directories. Any differences between the current state of the system and a known good state will be displayed, thus helping identification of unauthorized changes.

- **Specific directory inspection:**

Command: `diff --recursive /path/to/check/specific_directory /path/to/known-good/specific_directory`

This command compares the current state and a known good state.

- **Quick file and directory comparison:**

Command: `diff -qr /path/to/check /path/to/known-good`

This command performs a quick comparison of files and directories, identifying differences in file contents and any missing or added files.

Response and incident evidence

When addressing the compromise of a Linux server, it is crucial to effectively communicate with reporting individuals and properly document incidents. A sample response template to be used for reporting is provided as an appendix. After investigating the incident, relevant stakeholders - such as administrators - must receive relevant notification, especially in cases of severe compromises. You can contact those individuals using [Whois Domaintools](#) or by checking the official website and contact page of the organization.

The incident should be documented in the incident management tool by creating a new case, with relevant tags including IP addresses, identified vulnerabilities, and attack techniques used. This information can be obtained through a detailed system analysis.

When analysing an incident, it is important to use appropriate tools that enable detection of a compromise and its detailed analysis. Recommended tools include:

- `script`: logs the output of all commands entered in the terminal to a file, which is useful for documentation and further analysis.
- `uname -a` and `lsb_release -a`: provide detailed information about the operating system and its version.
- `w`: displays information about logged-in users
- `env`: shows the current system environment variables
- `ifconfig` and `arp -a`: display network configurations and ARP tables
- `netstat -ltn` and `ss -ltn`: allows the user to check current network connections.
- `ps -aux`: displays running processes.
- `lsmod` and `modinfo`: are used to view and examine loaded kernel modules.
- `crontab`: checks scheduled tasks.
- `cat /etc/group` and `cat /etc/passwd`: allow verification of system accounts and their group memberships.

Using these tools ensures a detailed analysis of the system's state and facilitates the identification of potential vulnerabilities and compromises. Based on this analysis, the most relevant data will be selected and included in tags, such as IP addresses, identified vulnerabilities, and attack techniques used. This process is crucial for a thorough resolution of the current incident and prevention of similar incidents in the future. Additionally, similar systems within the infrastructure will be checked for the same incident and analyzed. Any identified vulnerabilities will be addressed to ensure comprehensive security measures are in place. Once all necessary steps have been completed and the issue has been resolved, the incident can be formally closed.

References

- [1] Nikkel, B. (2021). Practical Linux Forensics: A Guide for Digital Investigators. No starch Press.
- [2] Casey, E. (2011). Digital evidence and computer crime: Forensic science, computers, and the internet. Academic press.
- [3] Ward, B. (2021). How Linux works: What every superuser should know. no starch press.
- [4] Linux Incident Response Guide. [online] Available at: <https://www.halkynconsulting.co.uk/a/2020/11/linux-incident-response-guide/> [Accessed on 22.7.2024]

3. Digital forensics

Digital forensics is a critical field within cybersecurity (incident response) and law enforcement, focusing on identifying, acquiring, preserving, and analyzing digital evidence. This chapter explores the fundamental principles of digital forensics, starting with the initial phases of forensic analysis, as defined by the ISO/IEC 27037:2012 security standard. These phases include the identification of digital evidence sources as well as collection, acquisition, and preservation of data integrity. Understanding these steps is essential for conducting a thorough and legally admissible digital forensic investigation.

The chapter then examines various sources of digital evidence, including computers, storage media, networks, and virtual machines, highlighting the importance of correctly identifying and securing forensic data. It then explores digital evidence acquisition, discussing the preparation and selection of reliable forensic tools and acquisition methods used to capture and verify digital artifacts. Special attention is given to forensic acquisition in virtualized environments, covering ESXi, Hyper-V, and Azure virtual machines.

Much of the chapter is dedicated to forensic tools, mainly the KAPE tool, which streamlines the collection and processing of forensic artifacts. The discussion includes target options, module configurations, and best practices for using KAPE in forensic investigations.

Additionally, the chapter covers collaboration with law enforcement agencies, emphasizing legal considerations, handling police requests, and ensuring compliance with forensic procedures. The final section focuses on drafting a forensic report, detailing its key components, structuring the analysis, documenting tools and items of evidence, and ensuring the report meets professional and legal standards.

3.1. Initial phases of digital forensics

The initial stages of digital forensics are governed by the security standard ISO/IEC 27037:2012. This standard defines the following four phases [1]:

- **Identification** - the process that involves searching, recognizing, and documenting potential digital evidence.
- **Collection** - the process of gathering devices that may contain potential digital evidence.
- **Acquisition** - creating a copy of data within a defined set.
- **Preservation** - the process of preserving and ensuring the integrity and original state of potential digital evidence.

Figure 28 shows that according to the ISO/IEC 27037:2012 security standard, collection and acquisition are two parallel processes as they focus on different aspects of working with digital evidence. Collection focuses on physical devices, while acquisition puts emphasis on the data itself. In the following subchapters, we will examine each phase of the forensic investigation in more detail

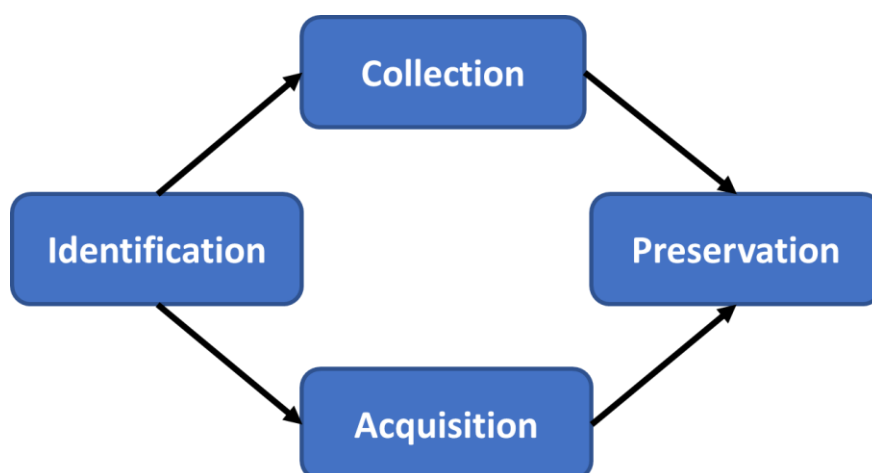


Figure 28: Initial phases of forensic investigation according to ISO/IEC 27037:2012 [1]

Identification

Security incidents or offences can be identified based on various indications, such as automated warnings or reports from individuals. Once a security incident has been identified, it is necessary to investigate, recognize, and determine the incident or offence to be investigated, which is the main task in this phase [2].

Identifying a security incident or offence normally also leads to the formation of multiple hypotheses about what may have happened. The forensic analyst may focus on identifying additional supporting information to determine whether a security incident has indeed occurred. Questions defined within the 5WH investigative model should always be asked during this phase [3]:¹⁹

- Who was involved?
- What happened?
- When did the cybercrime / cyber security incident occur?
- Where did the cybercrime / cyber security incident occur?
- Why did it happen?
- How did it occur?

Answers to these questions facilitate formulation of a hypothesis based on the information that triggered the forensic investigation. Additionally, this phase lays the foundation for the subsequent phases, as the obtained insights help identify which items are related to the case (cyber security incident), what digital evidence can be gathered, the extent of the security incident, etc.

Acquisition and collection

The goal of the acquisition phase is to collect data (digital evidence) from electronic devices in order to create a digital copy using proper forensic methods and techniques [4]. While most literature discussing the forensic process uses the term "collection", more technically oriented sources refer to it as "acquisition" or "extraction" and see them as two separate processes. The ISO/IEC 27037:2012 standard [1] provides clarity by distinguishing between the acquisition of digital evidence and the collection of devices that may contain potential digital evidence.

When acquiring data, it is essential to consider the order in which digital evidence is secured. This follows the principle according to which the most volatile data (those that change quickly) must be preserved first. Forensic analysts must account for this volatility

when beginning the process of securing digital evidence. Data should always be copied to a separate medium and forensic investigations, including analysis, should always be conducted on the copy to prevent accidental changes to the original source that might happen during the process. A device known as a write-blocker is typically used to ensure no data is written to the original storage medium. Finally, a digital fingerprint (hash) is created for the source medium and its copies.

Preservation

Once digital evidence is identified, protecting it from any alteration, including deletion, is essential. For digital evidence (e.g., logs), it may also be necessary to activate system elements that protect the records against modification or deletion.

For host systems, isolating the system from the rest of the computer network may be necessary using physical or logical controls or network access controls. Additionally, preventing users from accessing the suspect system is essential to ensure that users the digital evidence is not altered – whether intentionally or not.

Another key consideration in the context of evidence protection is the increasing reliance on virtual platforms. These systems can be preserved using snapshots and storage in non-volatile memory. The preservation phase should address the following issues:

- **Ensuring sufficient storage capacity to store the secured digital evidence** – The previous phase should determine the scope of the incident and identify the devices and system resources (memory, disk space). Based on this information, sufficient storage capacity must be allocated for storing the secured digital evidence (e.g., disk images, memory).
- **Ensuring data integrity** – Data integrity must be ensured for each secured piece of digital evidence by creating a digital fingerprint (hash).

3.2. Identification of digital evidence sources

This chapter focuses on the identification of digital evidence sources in the investigation of security incidents. It describes the importance of correctly determining locations and devices containing relevant digital evidence, including computers, storage media, computer networks, and virtual machines. It emphasizes the need to preserve data integrity, identify responsible individuals, and consider potential legislative constraints.

Upon arrival at the crime scene, it is necessary to identify locations/devices where digital evidence relevant to the case (incident) may be found. Correct identification of digital evidence requires knowledge of its location, type, and format [5]. This is an important and challenging task in forensic investigation.

In this phase, the forensic analyst should also consider collecting non-digital evidence. It is essential to create a list of individuals responsible for the identified devices, such as administrators, owners, or users. These individuals may provide information, such as system configuration or administrative access passwords, which may be useful when analyzing potential digital evidence.

This chapter is structured around four primary sources of digital evidence:

- computers and devices,
- data storage media,

- computer networks,
- virtual machines.

These four sources (computers and devices, storage media, computer networks, and virtual machines), representing the most common and critical areas where digital evidence is found in forensic investigations, have been identified based on practical experience, forensic methodologies, and standards such as ISO/IEC 27037:2012, which focus on the identification and preservation of digital evidence.

Computers / Devices

Another source of relevant data can be primary memory – RAM - which can provide information about running processes and the data they use, logged-in users etc. However, this data is only available in RAM while the system is powered, so in order to preserve it, the computer must not be shut down nor the memory modules be removed.

Data processing at the processor level, including registers and cache memory, is also of interest. However, as this data is even more volatile than that stored in RAM, any attempt at extracting it will result in partial or complete overwriting, due to the very nature of processor operation. Therefore, securing data from the processor is practically impossible.

Devices may be connected to various accessories and peripherals. It is essential to check for connections with external storage devices (e.g., NAS) or network interfaces that link the device to another network/device (e.g., using the Ethernet or Bluetooth protocol). Printers and scanners may contain forgotten documents and while these are not digital traces, they can provide valuable information (additional context). However, they may also come with inserted portable media for scanning or printing. Some printers and multipurpose devices have been equipped with hard drives that store copies of all scanned and printed documents. At the very least, metadata about the printed documents can be retrieved.

Inspecting peripherals connected to the devices, such as the monitor, keyboard, and mouse, is also highly recommended as many users stick important notes or access passwords on them. These are, again, not digital traces but they represent a valuable source of information. For example, identifying a simple password may help explain the attacker's initialization vector (i.e., how they compromised an email account).

Data storage media

Data storage media, mentioned in the previous section, come in various types and formats. The most widely known and used ones are the classic CDs/DVDs, USB drives (flash memory), as well as SD and MicroSD memory cards, but crucial evidence may also be stored on smart cards (usually certificates) or SIM cards. Various online storage solutions (cloud storage) are becoming increasingly widespread, so when securing evidence consider this possibility and request access to such storage systems if necessary.

In addition to standard storage media, we recommend paying special attention to physical objects as even a seemingly inconspicuous item – like a figurine or toy - may contain a hidden USB drive. These miniature portable media can be placed in multiple locations that are easy to overlook. This is especially relevant in cases where there is suspicion of data leaks involving an internal employee. Additionally, if a storage medium is connected to a device and the device is powered on, it should not be removed as it could be a hardware key or an encrypted USB drive, which, once disconnected, will remain encrypted without any possibility of decryption.

Computer network

Most universities or academic institutions have multiple branches located in a number of geographical areas. The computer networks of these institutions usually require several active network components and additional security features, such as firewalls, intrusion detection/prevention systems (IDS/IPS), security information and event management systems (SIEM), and VPN concentrators through which remote branches as well as employees and students, can connect. These devices may contain relevant digital evidence related to network communication, the assignment of a specific IP address, identities associated with the assigned IP address, network communication, and other relevant information.

It is also important not to overlook the network services provided within the computer network. For example, services such as DHCP, DNS + and email servers can yield very interesting and valuable insights. When addressing a security incident within such a network, malicious activity targeting one part of the network may adversely affect other segments within the organization. For instance, a ransomware attack targeting one faculty can have a negative impact on other parts of the institution.

Identifying sources of digital evidence requires careful consideration of possible complications caused by the unavailability of certain network segments during a cybersecurity incident. Other problems may arise from insufficient cooperation with the network administrator, which may be due to geographical distance. A further challenge could be cross-border data transfer (e.g., personal data or telecommunications traffic), being subject to various legislative regulations. Legislative requirements will significantly constrain the subsequent phases of digital forensic analysis.

Virtual machines

Forensic acquisition of evidence from virtual machines and hypervisors is a critical process in digital investigations involving virtualized environments. These hypervisors host multiple virtual machines, making it essential to capture data in a manner that preserves its integrity while ensuring comprehensive analysis. We focus on virtual machines in more detail in the guide that provides a brief overview of the key steps and best practices for acquiring forensic evidence from Hyper-V and ESXi hypervisors.

References

- [1] ISO/IEC 27037:2012, Information technology - Security techniques - Guidelines for identification, collection, acquisition, and preservation of digital evidence, 2012.
- [2] Reith, Mark, Clint Carr, and Gregg Gunsch. An examination of digital forensic models. *International Journal of digital evidence* 1.3, 2002: 1-12.
- [3] Palmer, Gary. A road map for digital forensic research. *First digital forensic research workshop, Utica, New York*, 2001.

3.3. Digital evidence acquisition

Introduction

This material presents available methods of producing forensic copies of evidence media. It also offers practical advice on how to approach the collection process itself, which is crucial in digital forensics to ensure the integrity and reliability of digital evidence.

Preparation

In order to acquire data from a computer's SSD/HDD drive it is necessary to prepare:

- **hardware write-blockers.** To prevent any modification of the source media a special device can be used to block write commands from being sent to the drive controller. A typical write blocker is a protocol bridge containing modified firmware or special electronics to intercept certain protocol commands. This type of equipment enables seamless creation of forensic copy of SATA, ATA, SCSI, SAS and USB media;
- **forensic software.** Forensic copies are most often created using dedicated programs. Both commercial and free versions of these tools are available for users. They generate forensic copies also in the form of multi-part archives - in formats that allow further forensic analysis of the data contained therein (RAW, E01, AFF4), calculation and verification of checksums as well as possible compression. Most importantly, this type of software produces a bit-by-bit copy of the evidence drive without altering the time stamps and other metadata of the files contained there;
- **an external storage device.** The process requires an external SSD or HDD to store the data collected from the compromised system. The storage device should be securely wiped with dedicated software, performing at least three passes to eliminate any risk of contamination. Additionally, consider the appropriate file system for formatting. At the moment, exFAT seems to be the most suitable for devices running Windows and MacOS, as they natively support writing and reading. Some Linux distributions also support exFAT without the need for additional software packages.
- **chain of custody form.** This is a document intended to prove the chain of custody, which should contain detailed information on how to handle digital evidence at each stage of the procedure. A detailed description of the form can be found in Chapter 1.6.

Reliable tools

Below we present examples of tools that can be used for collecting digital evidence:

- **Windows** - two software solutions running under the control of the Microsoft Windows are available for acquiring digital data: FTK Imager Lite by Exterro and Fex Explorer by GetData. Commercial solutions include EnCase Forensic from OpenText and X-Ways Forensics from X-Ways Software Technology AG;
- **MacOS** - two commercial solutions are recognized as reliable tools for creating forensic copies of disks installed on devices running MacOS: Digital Collector by Cellebrite and Recon ITR by Sumuri;
- **Linux** - an image of the evidence system can be created using the built-in `dd` command or "dc3dd" tool (if installed). For creating a physical copy (see acquisition types) of the Linux system media, you can use a laboratory computer equipped with the software referred to above.

Location and environment

The acquisition process should be conducted in a safe and controlled environment with stable electrical power (if the computer in question is a portable device, it should be supplied with adequate power supply). Access to the work area should be limited exclusively to essential personnel.

Prior to commencing the collection process, it is advisable to create photographic documentation of the environment where the evidence computer is located, as well as the device itself together with any connected equipment and, if the computer is running, the content displayed on the monitor. These photographs should be attached to the documentation prepared during the process, as will be discussed later in this document.

Types of acquisition

Physical acquisition

Physical acquisition involves a bitstream duplicate of data stored on a device including slack space that may be expected to contain deleted data.

Whenever possible, hardware write-blockers should be used to prevent changes to the original evidence

Forensic images should be acquired using software that is capable of capturing a complete bitstream image of the original media.

Logical

Logical acquisition is the process of obtaining structured data, such as folders and files, typically by utilizing the native file system in which they reside.

Forensic images should be acquired using software that is capable of capturing a logical image of the original content.

Remote acquisition

Remote acquisition is the process of acquiring structured data, such as folders and files, using network connections. This type of acquisition is an obvious choice when dealing with geographical challenges, or when on-site access is not immediately possible.

Forensic images should be acquired using software that is capable of capturing a logical image of the original media over the network, if possible.

Acquisition process

The guiding principle for computer forensic acquisitions is to minimize, to the fullest extent possible, changes to the source data. This is typically accomplished by using hardware devices, software configurations, or applications designed to enable reading data from a storage device without permitting modifications (writes) to the source media (i.e., write-blocking).

The first step is to identify the specific storage device that needs to be copied. Therefore, it will be necessary to contact the IT department responsible for the hardware within the organizational unit to determine the type of storage media installed in the evidence device.

This will enable the security team to prepare a sufficiently large disk to preserve the evidence data and take any additional steps, if necessary. However, if such verification is not possible, portable computers should be checked for the manufacturer and details of the model, that are shown on the factory sticker typically located on the bottom of the device. Alternatively, this information can be retrieved online. If the storage device is permanently mounted on the motherboard, a logical copy must be made.

Physical acquisition

Make sure the evidence computer is powered off and open its case using appropriate tools. Next, find the installed storage device and document its location with photographs. If possible, remove the storage device and attach it to a write blocker.

Select the correct device type (some solutions available on the market support only one interface type, e.g. SATA or NVMe) - connecting to the wrong interface may result in data loss or, in extreme cases, in damage to the evidence device!

Next, connect the write blocker to a lab workstation or laptop. Run the forensic information program installed on it (see Chapter 3: Reliable tools). Select the evidence disk as the source of the forensic copy process. In the next step, select the lab disk (external medium) where the copy will be made. Indicate the format of the copy (for example E01) and the checksum type (for example MD5). With the checksum it will be possible to confirm the integrity of the entire process. Once the process has been fully completed, place the medium back into the evidence device.

Logical acquisition

If the medium cannot be removed from the evidence device, creating a forensic copy will only be possible through a logical acquisition. Connect a previously prepared external laboratory medium, onto which a forensic copy will be made, to the running evidence computer. This drive should contain a portable version of one of the above-mentioned forensic computer tools (e.g. FTK Imager).

Next, run the program with administrator privileges and use it to identify all logical drives present in the evidence system. Indicate the selected logical drives (these will normally be the ones that may contain system logs and user files) and select the external medium as the destination for the forensic copy. Specify format of the copy and type of checksum. Once the copying process is complete, close the forensic software and disconnect the external drive.

Remote acquisition

Remote acquisition is a special form of logical acquisition, which requires installation of an agent program on the target computer to establish the connection. Installation of the agent must be done in cooperation of the IT team at the location of the evidence device as it requires administrative privileges and opening appropriate ports through which communication will be conducted. Once the agent is installed, the forensic software that connects to it (e.g. Magnet Axiom) will require constant and uninterrupted access to the network.

After establishing the connection with the agent, all the logical drives in the remote evidence system must be identified. Select the drives that are critical for the forensic process and choose the destination for the forensic copy. Specify the format of the copy and the type of checksum. Once the copying process is complete, uninstall the agent from the evidence system and close the forensic software on the laboratory side.

Verification and preview

Verification involves validating the integrity of the acquired data by comparing its hash to the hash of the acquisition source data. This process might not verify all data read from the subject media (for example, damaged sectors).

Investigators should review the acquired data to make sure that all intended items have been captured. They should also review output and error logs for indications of failures in the collection process, documenting - and, if possible, resolving - those failures as appropriate. Also, hash values should be computed over the acquired data using MD5 or SHA1 hash algorithm to facilitate subsequent data integrity validation.

After making a copy of the evidence media, it is essential to immediately verify the result using available tools (such as FTK Imager Lite) by mounting the image in read-only mode and attempting to access the contents of the files contained in it. If the contents of a file, such as a text document, are visible, the copy has been created correctly.

Please note, that any further actions related to the evidence material should be performed on the copy using software that prevents any modification or damage to its contents.

Documentation

Investigators should document every step of the digital evidence acquisition process. The documentation should include a description that will be sufficiently detailed to allow the definitive identification of the items to the exclusion of all others.

This information may include:

- Place where the action was performed (address, administrative unit, building, floor)
- Source of digital evidence (description of its location at the time of discovery)
- Unique task identifiers (case number, name of the investigator)
- Identifiers of source machine (vendor, model, serial number)
- Identifiers of source data carrier (vendor, model, capacity, serial number)
- Acquisition details (type of acquisition, imaging tool and version number)
- Hash value(s) of the acquired data
- Identifiers of destination data carrier (vendor, model, capacity, serial number)
- Photographs of the evidence that were taken during the entire acquisition process;
- Acquisition start/end date and time (including time zone)
- Errors encountered during acquisition
- Every interaction with evidence must be documented, creating a transparent and traceable history. This includes details of the person who collected the evidence, time and location of the occurrence as well as any subsequent changes to its location or condition.
- Investigators should document chain of custody. When digital evidence is transferred from one person to another, the chain of custody should include, at a minimum, the following details:
 - Unique identification of the item (vendor, model, capacity, serial number);
 - Details of transferring individual (name, position);
 - Details of receiving individual (name, position);
 - Date and time of receipt and transfer (including time zone);
 - Purpose of transfer (handover).

The document should be prepared in both paper and electronic formats. The paper version should remain with the unit that conducted the acquisition to ensure it is

available upon request from law enforcement agencies. The electronic version should be stored with the unit where the data was collected.

3.4. Digital evidence acquisition - virtual machines

Forensic acquisition of evidence from Hyper-V and ESXi hypervisors is a critical process in digital investigations involving virtualized environments. These hypervisors host multiple virtual machines, making it essential to capture data in a manner that preserves its integrity while ensuring comprehensive analysis. This guide provides a brief overview of the key steps and best practices for acquiring forensic evidence from Hyper-V and ESXi hypervisors.

ESXi

Suspending virtual machine

We recommend suspending a virtual machine before acquiring its disk in order to ensure data consistency by freezing the system state. This action will ensure that the active processes will not alter or corrupt evidence during the forensic capture.

To suspend a virtual machine, click on “Virtual Machines” in the left panel, select the desired virtual machine and click on “Suspend”.

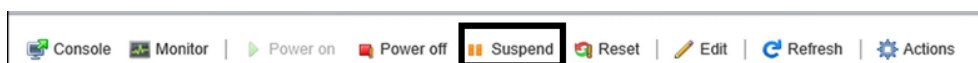


Figure 29: Suspend a virtual machine

Downloading images

To download the images of the virtual machine, click on “Storage” in the left panel. Then click on “Datastore browser”. Navigate to the location of the desired virtual machine and download its files one-by-one to a USB drive.



Figure 30: Download the images of the virtual machine

Hyper-V

Tested on Hyper-V Manager (version 10.0.22621.1). Applicable to all recent version of Windows 11 and Server 2022 operating systems.

Pausing virtual machine

Before extracting a virtual machine from Hyper-V, we recommend pausing it. To pause a virtual machine, right-click on the specific virtual machine and click on “Pause”.

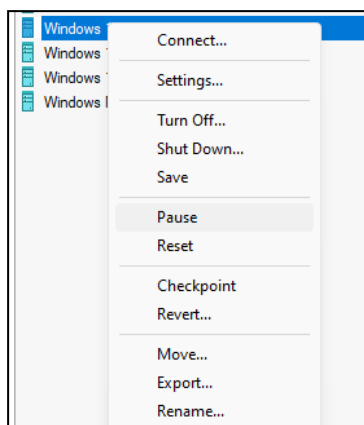


Figure 31: Pausing virtual machine

Extracting virtual machine

To extract an entire virtual machine, use the export button in Hyper-V by right-clicking on the virtual machine.

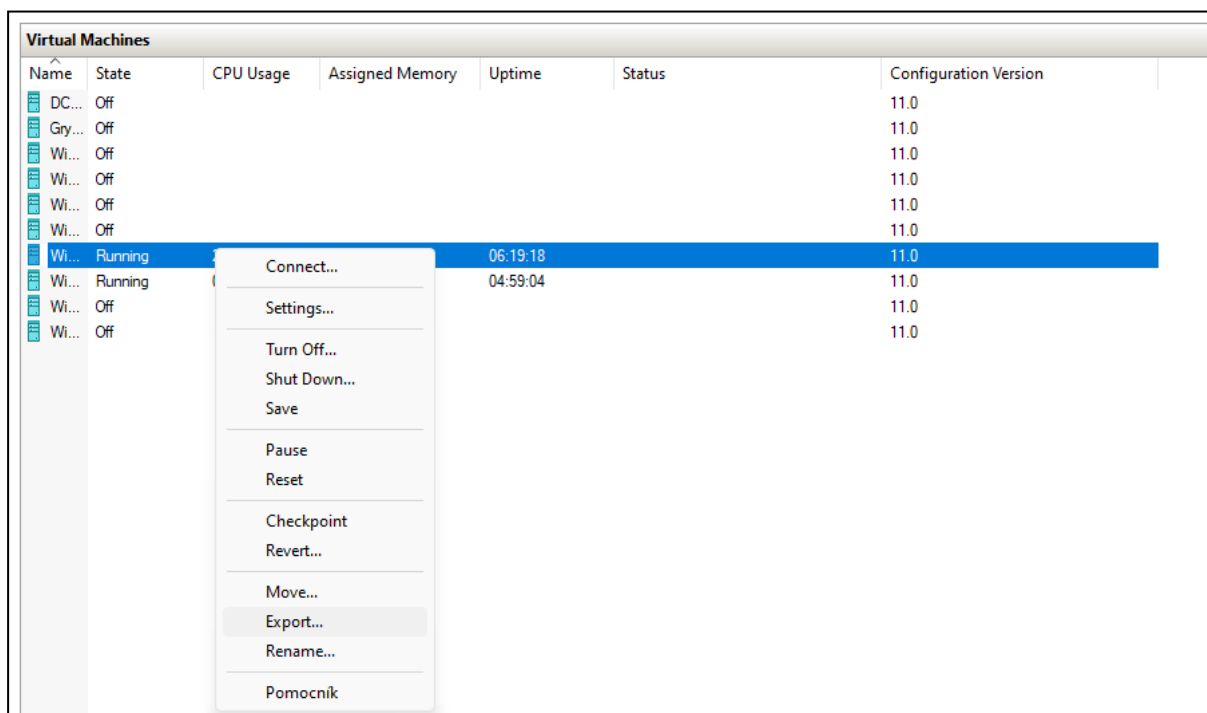


Figure 32: Extracting virtual machine

When checkpoints are enabled or used within a virtual machine, virtual hard drives are stored as a differencing virtual hard disk type. Differencing virtual hard disk type uses the base image as a read-only medium, while storing subsequent modifications of checkpoint images. This can cause potential problems when opening and/or forensically analyzing disk evidence.

When working with a checkpoint-enabled virtual machine, we recommend merging the checkpoints, or exporting only the current checkpoint of the virtual machine to perform the merge operation automatically. To export a single checkpoint, right-click on the checkpoint (not on the virtual machine) and select “Export”.

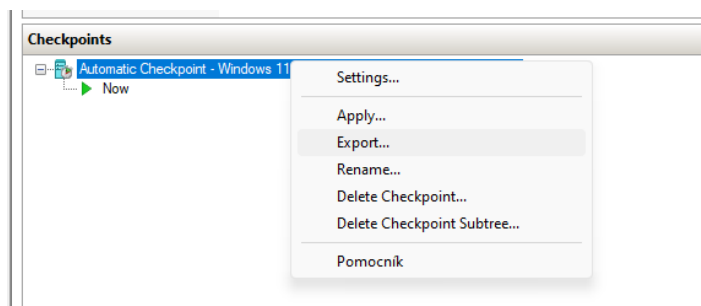


Figure 33: Exporting virtual machine

Azure virtual machine

To conduct a forensic investigation of a virtual machine in Azure, a VHD image must be extracted for further analysis. Extract a virtual machine requires a snapshot of the virtual machine's managed disk to be created in order to capture its current state without shutting it down. As a next step, use the snapshot to create a new disk, which serves as a forensic copy.

For detailed instructions see the following guide developed by Microsoft: <https://msrc.microsoft.com/blog/2019/09/acquiring-a-vhd-to-investigate/>

3.5. Digital evidence acquisition – KAPE tool

This chapter provides a comprehensive guide to using KAPE (Kroll Artifact Parser and Extractor), with a focus on its graphical user interface (gkape.exe). It explains how to configure targets and modules, execute data acquisition and parsing, and customize workflows for forensic investigations. The chapter offers practical insights for efficient evidence collection and analysis.

KAPE offers significant advantages in DFIR, including speed, flexibility, remote acquisition capabilities as well as detailed logging, making it ideal for rapid triage and targeted artifact collection. However, it is limited to targeted collection rather than full disk imaging and lacks built-in analysis capabilities.

Introduction

KAPE (Kroll Artifact Parser and Extractor) is a highly efficient and configurable triage program that will target essentially any device or storage location, find forensically valuable artifacts, and parse them within a matter of minutes.²⁰ **The key feature of the tool is triage** (selection of essential data), performed on a device – whether it is powered on or off. This capability proves invaluable in time-sensitive situations or when dealing with large volumes of data. KAPE also supports remote data acquisition, provided the incident responder has access to the remote device.

KAPE can be used to capture memory, extract most valuable forensic artifacts, and apply parsers. Even when full disk imaging is required, KAPE can be a valuable tool as it can offer a quick insight into the security incident.

KAPE generates a log file, which includes all executed commands, the time of execution, and possible errors. This log can be used to describe the process of evidence collection,

ensuring compliance with best forensic practices without the need to manually document every step and its timestamp.

Using KAPE

KAPE can be run from command line (“kape.exe”) on Windows OS with administrator privileges or through a GUI (“gkape.exe”). This guide focuses on the GUI version, which, essentially provides a command based on the entire application. The command can be executed manually at any time, if needed.

Figure 34 shows the gkape.exe window with its two main sections — one for configuring targets (left) and one for modules (right). There is one additional section at the bottom of the interface for displaying the current command.

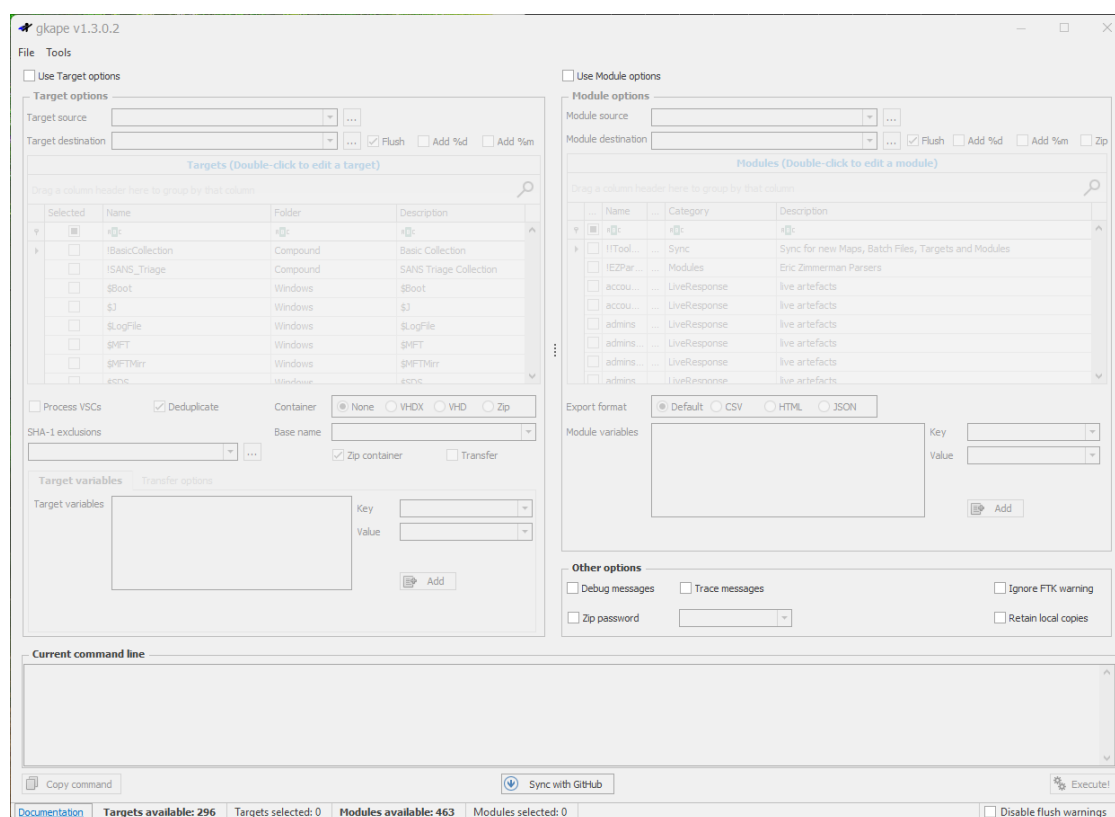


Figure 34: Graphical interface of KAPE

The “Target” section is used to select specific artifacts to be extracted by KAPE, while the “Modules” section contains various parsers and tools that enhance selected targets through additional analysis. For example, if the user extracts registry files, a module called “RegRipper” can be executed on the extracted data.

Target options

After selecting “Use Target options” both source and destination folders must be defined (Figure 35, point 1.). The **Target source** refers to the disk partition of a mounted image or a live system partition. Additionally, a specific folder can be selected. The **Target destination** is usually an empty disk designated for data acquisition or a folder in incident responder’s laptop.

Figure 35, point 2. shows three options that can be used under specific conditions: the “**Flush**” checkbox clears the Target destination prior to all other actions; the “**Add %d**”

option adds a timestamp to the destination folder to ensure uniqueness; finally, “Add %m” adds a machine name to the destination folder, when more devices are examined.

☒ Use Target options

1.

Target options

Target source: C:\

Target destination: D:\output

2.

☒ Flush ☐ Add %d ☐ Add %m

Targets (Double-click to edit a target)

3.

Drag a column header here to group by that column

Selected	Name	Folder	Description
☒	c:\	c:\	c:\
☐	!BasicCollection	Compound	Basic Collection
☐	!SANS_Triage	Compound	SANS Triage Collection
☐	\$Boot	Windows	\$Boot
☐	\$J	Windows	\$J
☐	\$LogFile	Windows	\$LogFile
☐	\$MFT	Windows	\$MFT
☐	\$MFTMirr	Windows	\$MFTMirr
☐	\$SDS	Windows	\$SDS

4.

☐ Process VSCs ☒ Deduplicate

SHA-1 exclusions

Container: ☒ None ☐ VHDX ☐ VHD ☐ Zip

Base name:

Target variables

Transfer options

Target variables

Key: Value:

Add

Figure 35: Main Target options

Point 3 marks a list of all available targets. You can search through the targets by typing into an “ABC” marked line. Double-click on any of the targets to open its editor. Various output formats are available, as shown in Figure 35, no. 4.

You can select “None” to copy folders and files as they appear in the source file system. Other options require the “**Base name**” field to be filled in -we recommend entering the name of the source machine. When creating custom targets, you can mark a variable. The variable is surrounded by %, e.g., C:\Users%\user%\ . These variables can be replaced by values in the “**Target variables**” section, where key (in this case - user) and value (e.g., Default) are added. This allows target customization for each case.

The tool also supports direct transfer of captured data using SFTP, S3, AWS S3 or Azure storage (Figure 36). To start the transfer, select a container, together with base name provided. Next, you can select “Transfer” and fill in transfer options.

KAPE allows you to prevent unique files from being copied by specifying their SHA-1 hash. Additionally, to ensure only unique files are copied, you can enable the “Deduplicate” option. When this checkbox is selected, only files with unique SHA-1 hashes will be copied.

Figure 36: Possible transfer options

Module options

Modules can be used in two distinct ways. You can set both “Targets” and “Modules” in one session and run them together, allowing all the files captured by “Targets” to serve as input for “Modules”. As a result, the “Module” source is not a required field, as shown in Figure 37.

Figure 37: Using targets and options at once.

Sel...	Name	Folder	Category	Description
☐	DensityScout	Apps	FileMetadata	DensityScout
☐	DHParser	GitHub	Antivirus	Retrieve Defender DetectionHistory threat data into...
☐	dns	Windows	LiveResponse	live artefacts
☐	DumpIt_Memory	Apps	Memory	DumpIt Memory Acquisition
☐	DumpIt_Memory...	Apps	Memory	DumpIt Memory Acquisition for Windows ARM
☐	Events-Ripper	Compound	EventLogs	Events-Ripper: parse all supported events
☐	Events-Ripper_All	Events-Ripp...	EventLogs	Events-Ripper: All
☐	Events-Ripper_e	Events-Ripp...	EventLogs	Events-Ripper: Events File

Figure 38: Main options for Modules

Alternatively, you can run modules only where **Module source** and **Module destination** are required. There is one additional checkbox “Zip”, which, if checked, returns all the outputs in one compressed file.

Point 2 displays a list of all the modules and with Point 3 you can select preferred output formats. Most modules have their default formats but these, of course, can be changed.

The Module variables section and the Targets function in the same way. Use “Other options” in Point 4 to display more detailed information in the console log (Debug messages, Trace messages), protect zip files by provided password (Zip password checkbox), ignore warnings if FTK process is detected during execution (Ignore FTK warnings), and retain local copies of data created during the execution process (Retain local copies).

KAPE targets and modules

KAPE targets

Table 15 - Description of KAPE targets

Target Name	Description
!Sans_Triage.mkape	Compound module. The SANS Triage Collection target collects antivirus information, cloud storage metadata, combined system logs, group policy data, evidence of execution details, file system data, LNK files and Jump Lists, messaging client details, recycle bin information files, registry hives, remote access logs, scheduled task details, SRUM and SUM data, Windows Error Reporting (WER) information, ThumbCache data, WBEM information, BITS details, web browser artifacts, Windows Index Search data, and Windows Timeline evidence.
KapeTriage.mkape	Compound module. KapeTriage collects evidence from File System files (\$MFT, \$LogFile, \$Boot, \$J, \$SDS), Registry Hives, Event Logs, Scheduled Tasks, Evidence of Execution, SRUM data, SUM data, Cloud metadata, WER, WBEM, Web Browser data (IE/Edge, Chrome, Mozilla history), LNK Files, JumpLists, 3rd party remote access software logs, 3rd party antivirus software logs, Windows 10/11 Timeline database, and \$I Recycle Bin files.
!Basic_collection.mkape	Compound module. In addition to SANS, it collects USB logs, powershell console logs.
FileSystem.mkape	Collects Master File Table (\$MFT), Log File (\$LogFile), Journals (\$J), Security Descriptors (\$SDS), Boot sector (\$Boot), and Transactional NTFS (\$T)
BrowserCache.mkape	Collects cache folders from Chrome, Chromium Edge, Firefox, Internet Explorer (versions 9/10 and 11), and Brave. It also includes the collection of specific cache files such as Index.dat for IE 9/10 and the Edge WebcacheV01.dat file.

ChromeExtensions.mkape	Collects files related to Chrome extensions from the user's local application data folder for both modern (C:\Users%user%\AppData) and XP (C:\Documents and Settings%user%\Local Settings\Application Data) profiles, facilitating analysis of Chrome extensions.
WebBrowsers.mkape	Collects data related to web browser history, bookmarks, and other relevant information from Chrome, Edge, Edge Chromium, Firefox, Internet Explorer, Opera, Puffin Secure Browser, and Brave Browser
CombinedLogs.mkape	Collects event logs, trace logs, powershell, firewall, and USB logs
RDPLogs.mkape	Collects Remote Desktop Protocol (RDP) related event logs from RemoteConnectionManager, LocalSessionManager, RDPClient and RDPCoreTS
Applications	
AnyDesk.mkape	Collects artifacts related to the AnyDesk remote access tool, including trace logs (both .trace and .conf) from user profiles and ProgramData, videos (.anydesk) recorded during sessions, connection trace logs (connection_trace.txt), logs associated with the System user account, and chat logs (.txt) from the user profile
ClipboardMaster.mkape	Collects artifacts related to ClipboardMaster, a universal multi-clipboard tool for Windows XP+
Discord.mkape	Collects artifacts related to the Discord application

KAPE modules

Table 16 - Description of KAPE modules

Name	Category	Description	Output
!EZParser.mkape	Compound	Compound module using various Eric Zimmerman Parsers that cover a range of artifacts such as Amcache, AppCompatCache, Event Logs, Jump Lists, Prefetch, Recent File Cache, and more	csv, html, json
AmcacheParser.mkape	EZTools - Program Execution	AmcacheParser: extracts program execution information	csv

AppCompatCacheParser.mkape	EZTools - ProgramExecution	extracts AppCompatCache (shimcache) information	csv
iisGeoLocate.mkape	EZTools - IISLogs	geolocates IP addresses found in IIS logs, extracts unique IPs, records bad data from logs	csv
JLECmd.mkape	EZTools - FileFolderAccess	processes jumplist files	csv, html, json
LECmd.mkape	EZTools - FileFolderAccess	processes .lnk files	csv, html, json
PECmd.mkape	EZTools - ProgramExecution	processes prefetch files	csv, html, json
RBCmd.mkape	EZTools - FileDeletion	processes recycle bin artifacts	csv
RecentFileCacheParser.mkape	EZTools - ProgramExecution	extracts file names from RecentFilecache.bcf	csv, json
SBECmd.mkape	EZTools - FileFolderAccess	processes shellbags	csv
SrumECmd.mkape	EZTools - SRUMDatabase	SRUM Parser	csv, txt
SumECmd.mkape	EZTools - SUMDatabase	processes Microsoft User Access Logs	csv, txt
WxTCmd.mkape	EZTools - FileFolderAccess	processes Windows Timeline/Activities Cache files	csv
EvtxECmd.mkape	EZTools - EventLogs	uses bstrings to GREP for US Zip Codes	csv, html, json
RECmd_Kroll.mkape	EZTools - Registry	analyzes and exports information from the Windows Registry	csv

SQLCmd.mkape	EZTools - SQLDatabases	processes SQLite databases	csv, json
MFTECmd.mkape	Compound - EZTools - FileSystem	MFTECmd: processes all MFT files	csv, json
MFTECmd_\$Boot.mkape	EZTools - FileSystem	processes \$Boot files	csv, json
MFTECmd_\$J.mkape	EZTools - FileSystem	processes \$J / \$UsnJrnl\$J files	csv, json
MFTECmd_\$MFT.mkape	EZTools - FileSystem	processes \$MFT files	csv, json
MFTECmd_\$SDS.mkape	EZTools - FileSystem	MFTECmd: processes \$SDS files	csv, json
bstrings.mkape	Compound	Compound module running bstrings utilities (bstrings_Bitlocker.mkape, bstrings_CreditCards.mkape, bstrings_Email.mkape, bstrings_IPv4.mkape, bstrings_MACAddresses.mkape, bstrings_SSN.mkape, bstrings_UNC.mkape, bstrings_URLs.mkape, bstrings_USPhone.mkape, bstrings_WinPath.mkape, bstrings_ZipCodes.mkape)	txt
bstrings_Bitlocker.mkape	EZTools - KeywordSearches	uses bstrings to GREP for Bitlocker recovery keys	txt
bstrings_CreditCards.mkape	EZTools - KeywordSearches	uses bstrings to GREP for Credit Card numbers	txt
bstrings_Email.mkape	EZTools - KeywordSearches	uses bstrings to GREP for email addresses	txt
bstrings_IPv4.mkape	EZTools - KeywordSearches	uses bstrings to GREP for IPv4 addresses	txt

bstrings_MACAddresses.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for MAC Addresses	txt
bstrings_SSN.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for US Social Security numbers	txt
bstrings_UNC.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for UNC Paths	txt
bstrings_URLs.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for URLs	txt
bstrings_USPhone.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for US Phone Numbers	txt
bstrings_WinPath.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for Windows style paths	txt
bstrings_ZipCodes.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for US Zip Codes	txt
bstrings_AeonWallet.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for Aeon Wallets	txt
bstrings_BitCoinWallet.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for BitCoin Wallets	txt
bstrings_ByteCoinWallet.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for ByteCoin Wallets	txt
bstrings_DashCoinWallet.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for DashCoin Wallets	txt
bstrings_DashCoinWallet2.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for DashCoin Wallets	txt
bstrings_FantomCoinWallet.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for FantomCoin Wallets	txt
bstrings_MoneroWallet.mkape	EZTools - KeywordSe arches	uses bstrings to GREP for Monero Wallets	txt

bstrings_SumoKoinWallet.mkape	EZTools - KeywordSearches	uses bstrings to GREP for SumoKoin Wallets	txt
MFTECmd_\$I30.mkape	EZTools - FileSystem	processes \$INDX/\$I30 files	csv, json
MFTECmd_\$MFT_DumpResidentFiles.mkape	EZTools - FileSystem	processes \$MFT files, then dump resident files within the \$MFT	csv, json
MFTECmd_\$MFT_FileListing.mkape	EZTools - FileSystem	processes \$MFT files, then output file listing	csv, json
MFTECmd_\$MFT_ProcessMFTSlack.mkape	EZTools - FileSystem	MFTECmd: processes \$MFT files with FILE slack recovery	csv, json
MFTECmd_BulkExtractorCarvedMFTRecords.mkape	EZTools - FileSystem	MFTECmd: processes bulk extractor carved MFT files	csv, json
RECmd_AllRegExecutablesFoundOrRun.mkape	EZTools	RECmd: AllRegExecutablesFoundOrRun	csv
SQLECmd_Hunt.mkape	EZTools - SQLDatabases	SQLECmd: processes SQLite databases with Hunt mode	csv, json, txt
RegRipper.mkape	Compound - Registry	Compound module consisting of multiple other modules for tool RegRipper (RegRipper_SAM.mkape, RegRipper_SECURITY.mkape, RegRipper_SOFTWARE.mkape, RegRipper_SYSTEM.mkape, RegRipper_NTUser.mkape, RegRipper_UsrClass.mkape). It parses all supported hives	txt
RegRipper_NTUser.mkape	Registry	analyses the NTUser hive	txt

RegRipper_SAM.mkape	Registry	parses the SAM hive. The SAM hive contains Security Account Manager information, including user account details and password hashes.	txt
RegRipper_SECURITY.mkape	Registry	analyses the Security hive.	txt
RegRipper_SOFTWARE.mkape	Registry	analyses the Software hive	txt
RegRipper_SYSTEM.mkape	Registry	analyses the System hive	txt
RegRipper_UsrClass.mkape	Registry	analyses the UsrClass hive	txt
LogParser.mkape	Compound	Compound module that uses LogParser for parsing and analyzing various types of log files	csv
LogParser_ApacheAccessLogs.mkape	Webservers	parses NCSA log format	csv
LogParser_DetailedNetworkShareAccess.mkape	RemoteAccess	parses Windows Event Log entries (Event ID 5145) related to network share access.	csv
LogParser_LogonLogoffEvents.mkape	AccountUsage	processes Windows Event Logs, extracts relevant fields related to logon and logoff events	csv
LogParser_RDPUsageEvents.mkape	RemoteAccess	processes Windows Event Logs, extracts relevant fields related to Remote Desktop Protocol (RDP) usage events	csv
LogParser_SMBServerAnonymousLogons.mkape	RemoteAccess	processes Windows Event Logs, extracts relevant fields related to SMB server events with an anonymous logon attempt	csv

Hayabusa.mkape	Compound	generates timelines from Windows event logs using the Hayabusa tool, which includes processors for event statistics, live response, and logon summary	csv
hayabusa_EventStatistics.mkape	EventLogs	performs analysis on ETW logs using hayabusa.exe	csv
hayabusa_LiveResponse.mkape	EventLogs	conducts a live analysis of system events	csv
hayabusa_LogonSummary.mkape	EventLogs	conducts a live analysis of system events	csv

3.6. Collaboration with Law Enforcement Agencies

Police requests

Czech Police occasionally contact the Masaryk University (MU) Seecurity Team regarding incidents for which they require additional data, such as user identity or communication logs. Each request must be officially submitted through the Masaryk University Data Box or as an official physical letter addressed to the University Rector's Office. If the request is received through any other means (email, phone, in-person submission, etc.), the request must be declined and the requestor should be redirected to the official Data Box. However, to speed up the process, the Securit Team can start gathering preliminary data before the request is received through the official channels.

Request handling process

- The request is submitted through the Data Box (official Czech eGovernment tool) and received by the Rector's office, specifically by the Security Director of MU (CSO).
- The CSO forwards the request to the Head of the CSIRT-MU Team through an encrypted channel.
- CSIRT-MU Team processes the request and gathers the requested data. During this stage, it is essential to assess whether the data can be legally provided. In most cases, consultation with the legal department is required.
- Results are then transferred to the MU letterhead (a template is available online).
- The Head of the CSIRT-MU Team signs the letter, and sends it to the CSO through an encrypted channel.
- CSO reviews the letter, and submits it to the Police through the Data Box.
- If the volume of data is too large to be transmitted via Data Box, a personal visit from a Police officer is required so that the data can be transferred using physical media.

A specific example of a police request refers to CCTV camera footage. However, this is an area of physical security where cybersecurity team has no mandate or access. If a CCTV request is received, the CSIRT-MU Team redirects it to the Rector's office through official channels. Finally the CSO forwards the request to the appropriate department.

Legal considerations

Czech law contains several provisions, which allow law enforcement agencies to request data. The provisions specify what data that can be requested and under what conditions. In many cases, the request is not aligned with the data and the applicable legal provisions. Hence, it is vital to check the legality of the request and deny it, if there are any doubts. Extra caution should be taken if you are asked to provide details of your identity or personal information.

Another critical consideration is the data retention period. There are multiple laws defining the minimum duration of data storage and mandating permanent deletion of data after a specified period of time. Handing over data that should have been deleted may constitute an acknowledgment of legal non-compliance. Similarly, not retaining the required data we may indicate a violation of the law.

3.7. Digital forensics report

In the realm of digital forensics, the forensic report is a critical component of the investigative process. It serves not only as a summary of findings but also as a detailed account of the methodologies employed and the evidence analyzed. A well-crafted forensic report can mean the difference between a successful resolution of an incident and a legal challenge that undermines the entire investigation. This chapter explores the structure, sources, and best practices for creating an effective forensic report.

Foundation of a forensic report

The forensic report emerges from a comprehensive examination of digital evidence, and its complexity reflects the intricacies of the particular case. At its core, the report must document observations, conclusions, and the supporting data that validate those conclusions. Forensic analysts must ensure that their reports are precise, technically accurate, and free of oversimplifications, especially since these documents may be subject to legal scrutiny.

Key components of a forensic report

A forensic report can be drawn up using a security incident report template, including key sections such as executive summary, timeline, methodology, limitations and conclusion. As this kind of report is more technical than the incident report, it requires some additional sections.

Examiner bio/background

It is essential to introduce the forensic analyst by outlining their qualifications, including education, training, and experience. A well-documented background fosters trust and credibility, particularly in legal scenarios where the analyst may be summoned to testify.

Tools utilized

A detailed inventory of hardware and software tools used in the investigation is crucial. This section should include the name of the manufacturer, the model and serial numbers of physical tools, as well as names and versions of the software, to ensure transparency with regard to the methodologies used.

Evidence

The report must provide a comprehensive list of all items of the evidence obtained, detailing the date, time, location, and the analyst responsible for acquiring each piece of evidence. This meticulous record-keeping is essential for maintaining a chain of custody and supporting the integrity of the evidence.

Forensic analysis

This section should clearly outline the specific actions taken during the investigation. Dates, times, and detailed descriptions of procedures are vital, as they provide context for the findings and support the overall narrative of the report.

Tool output

Outputs from forensic tools—often complex and extensive—should be included in the report. Analysts should focus on capturing relevant data, incorporating large outputs as appendices while ensuring that critical findings are highlighted in the main body of the report.

Report on language and structure

The language used in a forensic report is as important as its content. Analysts must be adept at categorizing their statements into three distinct types:

- **Technical Statements:** These provide objective descriptions of findings related to specific data points, such as noting the presence of a program execution entry in a security log.
- **Investigative Statements:** These offer interpretations of technical findings, such as linking the presence of an executable to its execution history stored in prefetch files.
- **Evaluative Statements:** Here, analysts express their opinions on the strength of the evidence and its implications, such as assessing the likelihood of malicious activity based on the findings.

Analysts should avoid conjecture and ensure that all conclusions are supported by data. A well-written report must allow for repeatability, allowing another analyst to replicate the findings based solely on the documented evidence.

The Importance of peer review

Before finalizing a forensic report, implementing a peer-review process is essential. This step ensures that all actions taken, analyses performed, and conclusions drawn are validated against evidence. A review helps identify potential oversights and enhances report credibility, preparing it for potential legal review.

Conclusion

The creation of a forensic report is a meticulous process that requires attention to detail, a clear structure, and a strong emphasis on technical accuracy. By carefully documenting every step of the investigation, analysts not only contribute to the immediate resolution of incidents but also safeguard the integrity of the entire forensic process. As digital threats continue to evolve, the ability to produce a thorough and credible forensic report remains an indispensable skill for all forensic analysts.

4. Live forensics

Live forensics is a crucial component of digital forensic investigations, enabling the collection of volatile data from a running system before it is powered off. It allows investigators to retrieve critical information, such as active processes, network connections, and open files, which are essential for forensic analysis. This chapter explores live forensic techniques across different operating systems. It begins with the Windows operating system, where the collection of live artifacts such as memory contents, active sessions, and system logs is critical for forensic investigations. The chapter then examines Linux live forensics, detailing the tools and methodologies required to extract and analyze system state, running processes, and network activity. Finally, it covers macOS live forensics, focusing on the unique challenges of acquiring forensic data from macOS systems, including file system structures, encryption mechanisms, and alternative forensic acquisition methods.

4.1. Live forensics – Windows operating system

Introduction

This chapter presents guidance on how to approach the process of live acquisition of Windows operating systems. The purpose is to collect as much volatile data as possible before the system is powered down or altered in any way, ensuring the integrity and availability of critical information that could otherwise be lost. The focus is on network configuration and connections, running processes and services, and system users.

Preparation

Prepare an external medium for data acquisition, formatted to NTFS or exFAT. Ensure the device is clean and has sufficient space available. If possible, estimate the required space in advance (e.g. by consulting an administrator about the disk capacity of the device).

Begin by **documenting the time of arrival** and everyone present at the site to establish the chain of custody. Before making any changes, take photographs of the scene, including electronic devices, monitors, and wiring.

Obtain administrator access to the system before starting forensic activities. The best approach is often to ask the system administrator for the password.

Consider capturing the memory before proceeding with individual artifact acquisition.

Run the following commands in **Powershell with administrator privileges**, redirecting their output to the prepared external medium. Use the following format:

```
command >> path_to_the_file/file_name.txt
```

Keep a record of all executed commands and update the chain of custody accordingly.

After running all the commands below, consider isolating the device from the network (if not already done), to prevent further damage.

Artifacts

Note: All of the following commands were tested on Windows 11, version 22H2, and PowerShell PSVersion 5.1.22621.4111.

- **OBTAINING NETWORK CONFIGURATION**

```
ipconfig /all
```

This command lists all network devices and their associated configurations.

- **OBTAINING ACTIVE NETWORK CONNECTIONS**

```
netstat -ano
```

If using an admin account: `netstat -anob`

This command lists active network connections. The `-b` parameter displays the executable responsible for creating each connection.

- **OBTAINING LOCAL DNS CACHE**

```
ipconfig /displaydns
```

This command lists the local DNS cache.

- **OBTAINING ROUTING TABLE**

```
netstat -rn
```

This command displays the routing table, including interfaces.

- **OBTAINING ARP CACHE**

```
arp -a
```

This command lists the ARP cache.

- **OBTAINING OPEN SESSIONS**

```
net use
```

```
net session
```

```
nbtstat -s
```

These commands show open sessions, network drives, printers, etc.

- **OBTAINING FIREWALL RULES**

```
netsh advfirewall show allprofiles
```

```
Get-NetFirewallRule | where {($_.enabled -eq $True)}
```

These commands show the FW configuration for all profiles and all enabled rules.

- **OBTAINING LIST OF LOGGED USERS**

```
query user
```

```
query session
```

These commands show logged-on users and information about remote sessions.

- **OBTAINING RUNNING PROCESSES**

```
tasklist
```

```
tasklist /m
```

These commands show running processes, their loaded modules, PID, session name, number of sessions, and memory usage.

```
wmic process list brief
```

```
wmic process get commandline,name,parentprocessid,processid
```

These commands also show running processes, their priority, PID, number of threads, and working set size. The last command specifies selected properties to display.

- **OBTAINING SERVICES**

```
tasklist /svc
```

```
sc.exe query state= all
```

```
Get-Service | select-object name, serviceType, displayName, Status
```

```
net start
```

These commands list all running and stopped services. The second command using `sc.exe` is a legacy command for “Service Control”, so it may stop working overtime.

- **OBTAINING LIST OF USERS**

```
net user
```

```
wmic useraccount list
```

These commands show all user profiles on the system. The latter provides more details, such as SID, status, and more.

- **OBTAINING LIST OF ADMINISTRATORS**

```
net localgroup "Administrators"
```

This command shows all users with administrator privileges.

- **OBTAINING LIST OF SCHEDULED TASKS**

```
schtasks /query /v /fo LIST
```

This command shows scheduled tasks and their details, such as timestamps, status, tasks to run, and associated users.

Additional information

All the commands above are based on tools available within the system. If it is suspected that the system files may be compromised, it is recommended to use a clean copy of the executables and run them from external media rather than from the compromised device.

External tools, such as Sysinternal tools²¹, Kroll Artifact Parser and Extractor (KAPE)²², or Eric Zimmerman tools²³, can also be used for live analysis. These tools can facilitate the acquisition process. However, it is important to use their command-line versions whenever possible to avoid changes to the artifacts. It is also necessary to ensure that the details of the tools used, including their versions, are documented for reference.

4.2. Live forensics – Linux operating system

Introduction

This material provides guidance on how to approach the process of live acquisition on Linux computers. The purpose of this activity is to gather as much volatile data as possible – information that is only available while the system is running and will be lost upon shutdown. This includes memory contents, active processes, current system information, and unsaved data.

DISCLAIMER

These commands have been tested on Ubuntu 24.04. However, it is important to note that some command outputs or options may vary slightly depending on specific system configurations or updates in this release.

Preparation

Before beginning the live acquisition on an Ubuntu 24.04 Noble Numbat system, it is essential to prepare an external storage medium where all collected data will be saved. The external storage device should be formatted with a filesystem compatible with both Linux and other operating systems, such as exFAT, ensuring seamless data accessibility across different platforms. While Linux supports ext4, exFAT is recommended when transferring data between Linux and non-Linux systems.

In order to run the required commands, root or sudo privileges are necessary. Ensure that you have access to the administrator password or credentials to execute commands that require elevated privileges. Without this access, the data acquisition process may be incomplete or even impossible.

The environment where the live analysis is conducted must be stable and controlled. Begin by documenting the time of arrival and noting anyone present at the site. This marks the start of the chain of custody. Before interacting with anything, take photographs of the scene, including electronic devices, monitors, and wiring. If possible, ensure the machine has a stable power supply.

Before starting any forensic activities, ensure you have obtained administrator access to the system. The most effective approach is typically to request the password directly from the system administrator.

When beginning the analysis, use the script command to record the entire terminal session. This will log all executed commands and their outputs, providing a complete and verifiable record of the live acquisition process.

As you execute the commands, redirect their output to files on the external medium using the following format:

```
command >> /path_to_the_file/file_name.txt
```

This will append the output of each command to a file, ensuring that the data is properly stored for future analysis.

Artifacts

- **OBTAINING DATE**

`date` Retrieves the current date and time from the system's internal clock and saves it to the specified file.

The time zone used is determined by `/etc/localtime` or the `TZ` environment variable.

- **OBTAINING SYSTEM INFORMATION**

`uname -a` Retrieves a detailed description of the system's kernel, including the kernel name, version, and machine hardware.

`hostname` Displays the system's hostname, used to identify the machine on a network.

`cat /proc/version` Shows the Linux kernel version and build information from the `/proc` filesystem.

`lsmod` Lists all currently loaded kernel modules.

`service --status-all` Provides a complete list of services running on the system, indicating which ones are active or inactive.

- **OBTAINING DRIVE INFORMATION**

`fdisk -l` Lists all disk partitions, including their sizes, sector details, and I/O configurations, which helps map out the system's storage devices.

`lsof -i` Lists all active network connections and open files associated with internet protocols, providing insight into network activity and potential remote connections.

- **OBTAINING NETWORK CONFIGURATIONS**

`ifconfig -a` Displays the configuration of all network interfaces, including both active and inactive ones.

`netstat -apetul`, `netstat -plan`, `netstat -plant` Show detailed statistics about active network connections, routing tables, interface statistics, and masquerade connections.

`ss -l`, `ss -ta`, `ss -tp` Provide socket statistics, showing the state of TCP connections and other network protocols.

- **OBTAINING CURRENT USER AND SESSION INFORMATION**

`whoami` Displays the username of the current user.

`who` Lists all users currently logged into the system.

- **OBTAINING LOGIN HISTORY**

`last` Shows the login history of users, including login and logout times.

`lastb` Displays a list of failed login attempts.

- **OBTAINING USER AND AUTHENTICATION LOGS**

`cat /var/log/auth.log` Displays authentication logs, including login records and failed attempts.

`cat /etc/passwd` Lists all user accounts on the system.

`cat /etc/shadow` Displays encrypted passwords and account expiration details.

`cat /etc/sudoers, cat /etc/sudoers.d/*` Display users with sudo privileges.

- **OBTAINING USER LISTINGS**

`cut -d: -f1 /etc/passwd` Retrieves usernames from the `/etc/passwd` file.

`getent passwd | cut -d: -f1` Provides a similar listing but includes network authentication services.

- **OBTAINING PROCESS INFORMATION**

Various `ps` commands (`ps -s, ps -l, ps -o, ps -t, ps -m, ps -a, ps -aef, ps -auxwf`) provide details on active processes.

`top` Offers a real-time snapshot of system resource usage.

- **OBTAINING SYSTEM AND USER STARTUP CONFIGURATIONS**

`cat /etc/profile, ls /etc/profile.d/, cat /etc/profile.d/*` Display system-wide environment variables and startup scripts.

`cat ~/.bash_profile`, `cat ~/.bashrc` Show user-specific shell initialization scripts.

- **OBTAINING SCHEDULED TASKS (CRON JOBS)**

`ls /etc/cron.*, cat /etc/crontab` Show scheduled tasks executed at specific times.

- **OBTAINING ADDITIONAL CONFIGURATION FILES**

`ls /etc/*.d/*` Lists configuration directories storing system settings.

- **OBTAINING KERNEL MODULES**

`ls -la /lib/modules/*/kernel/*` Lists all loaded kernel modules with details.

- **OBTAINING BROWSER PLUGINS AND EXTENSIONS**

Commands like `ls -la ~/.mozilla/plugins` and `ls -la ~/.config/google-chrome/Default/Extensions/` list browser plugins/extensions.

- **OBTAINING PLUGGABLE AUTHENTICATION MODULES (PAM)**

`cat /etc/pam.d/sudo, cat /etc/pam.conf, ls /etc/pam.d/` Display authentication module configurations.

- **OBTAINING HIDDEN DIRECTORIES**

`find / -type d -name ".*"` Lists all hidden directories in the system.

- **OBTAINING EXECUTABLES**

`find / -type f -exec file -p '{}' \; | grep ELF` Identifies executable files in the system.

- **OBTAINING HIDDEN EXECUTABLES**

`find / -name ".*" -exec file -p '{}' \; | grep ELF` Searches for hidden ELF executables.

Additional information

The above steps outline commands that use built-in Linux tools for live forensic acquisition. The process can be customized or automated through scripting based on the specific requirements of the investigation. Additionally, alternative tools such as Volatility, LiME, and other third-party forensic suites are available to streamline the collection process in live environments.

4.3. Live forensics – macOS operating system

Introduction

This material provides guidance on how to approach the process of live acquisition of Apple computers. The goal is to collect as much volatile data as possible. This data refers to information available only while the system is running and will be lost once the system is powered off. It includes memory contents, active processes, current system information, and unsaved data, such as open documents. While RAM cannot be directly imaged, there are alternative methods available to gather volatile data.

DISCLAIMER

The commands presented below have been tested in the macOS 14.5 (Sonoma) environment. In earlier versions of the system, there may be discrepancies due to changes in the system architecture. In such cases, it is recommended to refer to the system documentation.

Preparation

In order to acquire data from a computer running macOS, it is necessary to prepare an external medium to which the data collected from the compromised system will be copied (assuming that the device in question has been preventively isolated from the network environment). Since the native macOS operating system does support saving data to the NTFS file system, it is recommended to prepare a device formatted with the exFAT file system. This will allow the device to be freely read on both Microsoft and Apple operating systems.

Before starting the acquisition, it is necessary to obtain the password that grants administrator rights to the system being tested. Without this, the acquisition process may be incomplete or, in the worst case, impossible. Only with access to the console with administrator rights will the correct execution of commands be possible, allowing for the collection of volatile data still present in the compromised environment.

Nothing prevents the subsequent actions outlined here from being automated by combining them into a single script executed with administrator privileges on the compromised machine. However, presenting them step by step below provides a detailed understanding of how each command works.

Location and environment

The triage process should be conducted in a safe and controlled environment with stable electrical power. If the computer in question is a portable device, ensure it has an adequate power supply. Take photographic documentation of the location where the evidence device is found and identify the user.

Information specific to live forensic acquisition of macOS

Launch a terminal and verify administrator privileges. Ensure that the terminal application has full access to the disk (under "Privacy and Protection" in "System Settings").

Attach a collection drive: Connect a drive to the Mac for data collection. Verify where the operating system mounts the media.

Use the terminal for documentation: Open the terminal and begin documenting actions with the script command: **script** /Volumes/your_drive/Evidence.txt. This will allow the recreation of the data collection process if necessary.

Execute data collection commands: The following commands can be executed in any order. Replace /your_drive/ with the identifier of the portable media where the data will be stored.

- **OBTAINING CURRENT DATE AND TIME**

```
date >> /Volumes/your_drive/Collection/Log.txt
```

When invoked without arguments, the date utility displays the current date and time as read from the kernel clock. By default, it shows the time in the time zone defined by `/etc/localtime` or the TZ environment variable.

- **OBTAINING COMMAND LINE HISTORY**

```
history >> /Volumes/your_drive/Collection/Command_History.txt
```

By default, the history command displays the last 15 executed commands along with their command numbers. The command above lists the entire command history.

- **OBTAINING SYSTEM INFORMATION**

```
system_profiler SPSoftwareDataType SPHardwareDataType >>
/Volumes/your_drive/Collection/Log.txt
```

```
fdsetup status >> /Volumes/your_drive/Collection/Log.txt
```

These two commands generate a brief description of the system's hardware and software configuration. The resulting text file includes information such as the device model, serial number, chipset type, and RAM capacity. It also contains details about the installed operating system version, FileVault status, and the time elapsed since the last boot.

- **LISTING SUSPENDED AND ACTIVE JOBS**

```
jobs >> /Volumes/your_drive/Collection/Jobs.txt
```

When invoked without any arguments, this command displays only active jobs in the operating system.

- **ENUMERATING RUNNING PROCESSES**

```
ps -afx >> /Volumes/your_drive /Collection/Running_Processes.txt
```

The “ps” utility displays a header line followed by lines containing information about all processes with controlling terminals. It includes details about both the user's processes and those belonging to other users, while omitting processes that do not have a controlling terminal. The “f” parameter provides additional details, such as UID, PID, parent PID, recent CPU usage, process start time, controlling TTY, elapsed CPU usage, and the associated command.

- **LISTING NETWORK DEVICES**

```
networksetup -listallhardwareports >> /Volumes/your_drive  
/Collection/Network_Dev.txt
```

This command is normally used to configure network settings in System Preferences. When used with the above switch, it displays a list of hardware ports along with their corresponding device names and Ethernet addresses.

- **OBTAINING NETWORK CONFIGURATIONS**

```
ifconfig >> /Volumes/your_drive /Collection/Network_Configuration.txt
```

The `ifconfig` utility is typically used to assign an address to a network interface and/or configure network interface parameters. It displays the current configuration for all network interfaces when no optional parameters are supplied.

- **LISTING ACTIVE NETWORK CONNECTIONS**

```
netstat -av >> /Volumes/your_drive /Collection/Netstat.txt
```

The `netstat` command symbolically displays the contents of various network-related data structures. In this case, it shows the state of all sockets with an increased level of detail.

- **GENERATING ROUTING TABLE**

```
arp -a >> /Volumes/your_drive /Collection/Routing_Table.txt
```

This command displays the Internet-to-Ethernet address translation tables used by the Address Resolution Protocol (listing all ARP entries).

- **GENERATING A LIST OF USERS AND CURRENT USER**

```
id >> /Volumes/your_drive /Collection/User_List.txt
```

```
who >> $1/Collection/User_List.txt
```

The `id` utility displays the user and group names, along with their numeric IDs, of the calling process, to the standard output. If the real and effective IDs are different, both are displayed; otherwise, only the real ID is displayed. The second command, `who`, provides information about currently logged-in users. By default, this includes login name, TTY name, date and time of login, and remote hostname, if applicable.

- **LISTING MOUNTED VOLUMES**

```
diskutil list >> /Volumes/your_drive /Collection/Mounted_Volumes.txt
```

```
df -h >> /Volumes/your_drive /Collection/Mounted_Volumes.txt
```

```
mount >> /Volumes/your_drive /Collection/Mounted_Volumes.txt
```

The `diskutil` command provides information about and allows administration of partitioning schemes, layouts, and formats of disks. This includes hard disks, solid-

state disks, optical discs, disk images, APFS volumes, CoreStorage volumes, and Apple RAID sets. The second command, `df -h`, displays statistics about the amount of free disk space on the specified mounted file system, or on the file system of which a file is a part. The third command, `mount`, prints a list of currently mounted file systems.

- **LISTING OPEN FILES**

```
ls -l >> /Volumes/your_drive /Collection/Open_Files.txt
```

This command shows a list of open files. An open file can be a regular file, directory, block special file, character special file, executing text reference, library, stream, or a network file. Rather than a formatted display, `ls -l` produces output that can be parsed by other programs.

- **GENERATING A LIST OF FILES FROM USER DIRECTORY**

```
ls -lRain ~/ >> /Volumes/your_drive /Collection/User_File_List.txt
```

This command generates a recursive list of directories and files from the user directory in long format. Each line includes detailed information, such as file mode, number of links, owner name, group name, file size, last modification date (abbreviated month, day, hour, and minute), and pathname. If a file or directory has extended attributes, the permissions field obtained by the `-l` option is followed by a '@' symbol. If the file or directory has extended security information (such as an access control list), the permissions field obtained by the `-l` option is followed by a '+' symbol. The output also includes directory entries starting with a dot ('.') and the file serial number (inode number) for each entry.

- **EXPORTING SYSTEM LOG FILES**

```
log collect --output /Volumes/your_drive/Collection/ --start YYYY-MM-DD
```

This command collects system logs into single `.logarchive` file, which can later be viewed with appropriate tools. If an output path is not specified, a file named `system_logs.logarchive` will be created in the current directory. If the specified path is a directory, a file named `system_logs.logarchive` will be saved in that directory. If the path includes the `.logarchive` extension, the file will be saved with that name at the specified location. `--start` limits the log capture to entries starting from the specified date and time up to the present. The accepted date/time formats are: "YYYY-MM-DD", "YYYY-MM-DD HH:MM:SS", and "YYYY-MM-DD HH:MM:SSZZZZ". Please note that the process of copying system logs can be time-consuming due to their large size, so it is advisable to define a time frame when possible.

- **OBTAINING TIME WHEN THE PROCESS WAS FINISHED**

```
date >> /Volumes/your_drive/Collection/Log.txt
```

Using this command saves information about the end time of the data triage process to the file, which may be useful for the preparation of further documentation and examination.

End the session:

- Type `exit` to end the script recording.
- Unmount the flash drive.
- Review the collected data: Access and review the output in the "Evidence.txt" file on the attached drive. Check the content of the files located in the "Collection" directory.

Alternatives

Please note that the algorithm presented above is based on the tools available in the system and can be freely modified by the user. There are also commercial tools available on the market that have graphical user interfaces, which can speed up and streamline the process (e.g. Cellebrite Digital Collector and Sumuri Recon ITR).

For free tools currently under development, UAC (Unix-like Artifacts Collector) can be mentioned. It is an incident response script that facilitates the collection of volatile data, not just from macOS systems. Another tool, Fuji (Forensic Unattended Juicy Imaging), is designed for performing forensic acquisition of Mac computers (it requires Python to be installed on the evidence system).

5. Digital artifacts analysis

Digital artifact analysis is a key step in digital forensic analysis as it enables the reconstruction of events, verification of forensic hypotheses, and retrieval of digital evidence related to security incidents. This chapter focuses on various categories of digital artifacts that may be subject to forensic examination. It first provides a detailed analysis of Windows operating system artifacts, including Master File Table (MFT), registry, event logs, and records of process execution and file operations. The chapter covers the analysis of artifacts across Linux and macOS operating systems, addressing critical system logs, configurations, and metadata. Another significant area is mobile device forensics (Android, iOS), exploring data extraction methods and available forensic tools. The chapter also delves into memory (RAM) analysis revealing active processes, network connections, or files stored in volatile memory, focusing on Windows. Network forensic analysis provides tools for examining network traffic, identifying malicious communications, and correlating data with other security incidents. Finally, the chapter explores Microsoft Entra ID (Azure AD) forensic analysis, focusing on security monitoring and examination of compromised accounts.

5.1. Analysis of Windows system artifacts – system artifacts

In this chapter, key artifacts integral to Windows forensic analysis are explored. These artifacts are essential for understanding and uncovering various activities and events across the Windows environment. Our focus is primarily on systems using the NTFS (New Technology File System), the most widely utilized file system in the Windows Operating System.

Master File Table (MFT)

Description

The Master File Table is a forensic artifact on NTFS volumes that maintains a record of every file on the volume (including itself), providing detailed timestamps, size information, alternate data streams, access control lists, and many more. In certain cases (when the file is small enough), the MFT may even include "resident" files that can be extracted directly from the MFT without having the rest of the volume.

The user can't directly access MFT even with administrative privileges – it needs to be extracted using forensic tools like FTK Imager or KAPE. Also, the format of the extracted MFT is usually non-human readable and needs to be parsed.

When files are deleted from an NTFS file system volume, their MFT entries are marked as free and may be reused.

Location

The \$MFT file is located under the root of each volume.

Interpretation

- Metadata of each file on the partition.
- Creation of thorough incident timeline.
- Proof of existence of some deleted files (as long as the data has not been overridden).

- Anomaly detection in access time patterns.

MFTECmd Tool

Note: tested on Windows 10 using version v1.0.0.0

MFTECmd²⁴ is a tool for analyzing \$MFT, \$Boot, \$J, and \$SDS files, it can be run using the KAPE tool. The MFTEExplorer²⁵ is recommended for viewing the outputs.

Sample usage:

```
MFTECmd.exe -f C:\$MFT --csv D:\Evidence
```

where the *f* parameter specifies the location of the extracted MFT file and the CSV parameter specifies the location of the output in the CSV format.

Sources

- <https://learn.microsoft.com/en-us/windows/win32/fileio/master-file-table>
- <https://digitalinvestigator.blogspot.com/2022/03/the-ntfs-master-file-table-mft.html>

EZViewer v1.0.0.0 - 20220209114243_MFTECmd_\$MFT_Output.csv

File Tools Help

	A	B	C	D	E	F	G	H	I
157	159	1	TRUE	66	1	.\Program	Windows NT		0
158	160	1	TRUE	159	1	.\Program	Accessories		0
159	161	1	TRUE	160	1	.\Program	en-US		0
160	162	1	TRUE	159	1	.\Program	TableTextService		0
161	163	1	TRUE	162	1	.\Program	en-US		0
162	164	1	TRUE	66	1	.\Program	Windows Photo Viewer		0
163	165	1	TRUE	164	1	.\Program	en-US		0
164	166	1	TRUE	66	1	.\Program	Windows Portable Devices		0
165	167	1	TRUE	66	1	.\Program	Windows Security		0
166	168	1	TRUE	167	1	.\Program	BrowserCore		0
167	169	1	TRUE	168	1	.\Program	en-US		0
168	170	1	TRUE	66	1	.\Program	Windows Sidebar		0
169	171	1	TRUE	170	1	.\Program	Gadgets		0
170	172	1	TRUE	170	1	.\Program	Shared Gadgets		0
171	173	1	TRUE	66	1	.\Program	WindowsApps		0
172	174	1	TRUE	173	1	.\Program	Deleted		0
173	175	4	TRUE	1E+05	2	.\Program	dmrc.idx	.idx	715504

Figure 39: Sample output created from \$MFT using the MFTECmd tool.

USN Journal

Description

This artifact functions as the filesystem's journal. The \$UsnJrnl file contains two alternate data streams - the \$Max and the \$J. The \$Max data streams contain information about the journal's settings such as the maximum size whilst the \$J data stream contains information about the changes.

Location

It is part of the MFT table, stored in a variable `$Extend\UsnJrnl`.

Interpretation

- File modification (or other action) time
- Type of action performed on the file (delete, rename)
- MFT entry identification (for correlation with the files in MFT).
- Timeline enhancement.

Tools

- MFTECmd (<https://github.com/EricZimmerman/MFTECmd>)
- Windows Journal Parser (https://tzworks.com/prototype_page.php?proto_id=5)

Sources

- <https://github.com/Psmths/windows-forensic-artifacts/blob/main/file-activity/usn-journal.md> (common events and examples)
- <https://countuponsecurity.com/2017/05/25/digital-forensics-ntfs-change-journal/>
- <https://forensafe.com/blogs/usnjournal.html>

Registry

Description

The registry acts as a hierarchical database that stores low-level settings for the operating system and applications. The data is considered volatile as it can change every second. The registry organizes data in a tree-like structure, each node being referred to as a key. Keys can contain subkeys and values which hold the actual data.

Keys: Serve as containers holding either values or subkeys. They are identified by names that do not include backslashes and are case-insensitive.

Values: Represent data items, similar to files, stored within keys. They can hold various forms of data, including text, numbers, and binary data.

The main registry files: System, Software, Security, and SAM registry.

All the changes are stored in transaction files and only after some time they are applied to the actual registry file. The transaction files have the same name as registry files, but their extension is .LOG. Make sure to collect those as well. Registry files without the transaction logs are often referred to as *dirty*.

Folder/predefined key	Description
HKEY_CURRENT_USER (HKCU)	Contains the root of the configuration information for the user who is currently logged on. The user's folders, screen colors, and Control Panel settings are stored here.
HKEY_USERS (HKU)	Contains all the actively loaded user profiles on the computer. HKEY_CURRENT_USER is a subkey of HKEY_USERS.

HKEY_LOCAL_MACHINE (HKLM)	Contains configuration information particular to the computer (for any user).
HKEY_CLASSES_ROOT (HKCR)	It is a subkey of HKEY_LOCAL_MACHINE\Software. The information that is stored here ensures that the correct program opens when you open a file using Windows Explorer.
HKEY_CURRENT_CONFIG (HKCC)	Contains information about the hardware profile that is used by the local computer at system startup.

Location

- C:\Windows\System32\config
- C:\Users\<Username>\NTUSER.DAT
- C:\Users\<Username>\AppData\Local\Microsoft\Windows\UsrClass.dat

Some useful locations in the registry:

- HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Mountpoints2 – tracks removable devices
- HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery – tracks Explorer search history
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run – start a program on user logon
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce – start program on user logon
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList – SSID, network type, profile GUID

Interpretation

- System configuration
- Installed programs
- User settings
- Group policies, access control settings, password hashes
- Previously connected external devices

Tools

- Registry Editor (built-in) – view and edit the registry files
- Access FTK Imager – registry acquisition
- Registry Explorer (<https://download.mikestammer.com/net6/RegistryExplorer.zip>) – view the content
- Registry Changes View (https://www.nirsoft.net/utils/registry_changes_view.html) – compare changes between snapshots

Registry Explorer

Note: tested on Windows 11 using version v2.0.0.0

To view live system hives, it is necessary to run the tool as administrator. All files of interest have to be loaded and data from default locations will not load automatically. While loading files, the tool detects dirty hives and gives you an option to update them. Then the selected hives can be searched manually.

In the tab named “Available bookmarks”, the most common and most important registry keys can be found, including Users, CurrentVersion, or MountedDevices (Figure 41), allowing for filtering the values or searching strings.

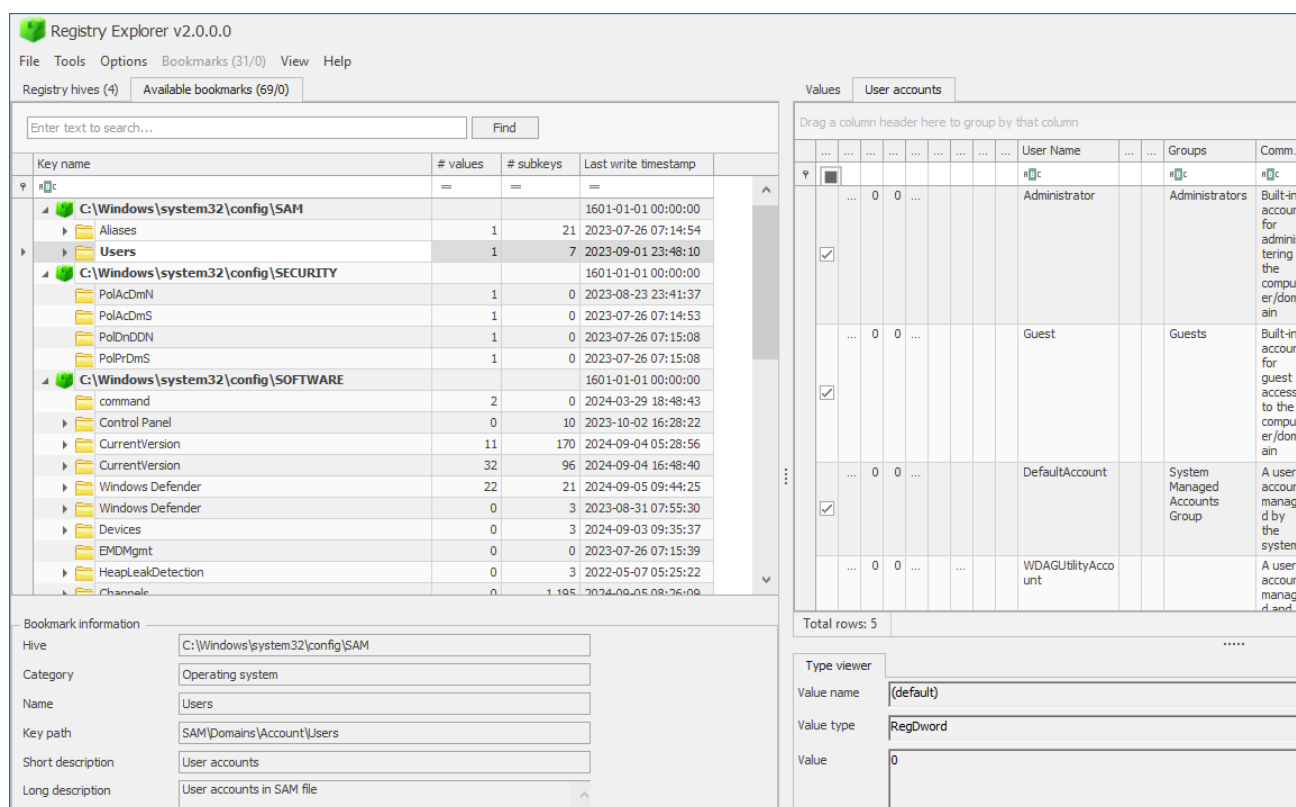


Figure 40: Example of Available bookmarks

Sources

- <https://learn.microsoft.com/en-us/troubleshoot/windows-server/performance/windows-registry-advanced-users>
- <https://www.forensicfocus.com/articles/windows-registry-analysis-101/>
- <https://medium.com/@mohit.r.phy/exploring-the-hive-deep-inside-the-windows-registry-pt-2-a46d7a3a7f7>
- https://13cubed.s3.amazonaws.com/downloads/windows_registry_cheat_sheet.pdf

Event Logs

Description

Windows event logs are a valuable source of evidence in digital forensics examinations. They provide a detailed record of system events, security-related activities, and application behavior. The data in logs is structured, but each type of log has a different pattern. Each event contains an ID, which specifies the type of action logged. Some useful IDs are presented below, in the Interpretation section.

The widely used logs include:

- Security log – contains information related to security events such as logon attempts, account management, and privilege use.

- System log – contains information related to system events such as driver installation, service startup, and shutdown.
- Application log – contains information related to application events such as software installations and application errors.

Location

Note: this is the default location which can be changed.

Windows XP: C:\Windows\System32\config*.evt

Windows 7/8/10/11: C:\Windows\System32\winevt\Logs

Interpretation

- Identify the initial point of compromise.
- Determine the extent of the attack and the damage caused.
- Identify any data exfiltration attempts.
- Reconstruct a timeline of events.

Table 17 - Security Event IDs of Interest

Event ID	Description
4624	An account was successfully logged on. (See Logon Type Codes)
4625	An account failed to log on.
4634	An account was logged off.
4648	A logon was attempted using explicit credentials. (RunAs)
4672	Special privileges assigned to new logon. (Admin login)
4720	A user account was created.
4688	A new process has been created. (If audited; some Windows processes logged by default)
4698	A scheduled task was created. (If audited)
1102	The audit log was cleared. (Security)

Table 18 - Logon Type Codes

Type	Description
2	Console
3	Network
4	Batch (Scheduled Tasks)
5	Windows Services

7	Screen Lock/Unlock
8	Network (Cleartext Logon)
9	Alternate Credentials Specified (RunAs)
10	Remote Interactive (RDP)
11	Cached Credentials (e.g., Offline DC)
12	Cached Remote Interactive (RDP, similar to Type 10)
13	Cached Unlock (Similar to Type 7)

Table 19 - System Event IDs of Interest

Event ID	Description
7045	A new service was installed in the system. (4697 in Security)
7034	The x service terminated unexpectedly. It has happened y time(s).
104	The x log file was cleared. (Will show System, Application, and other logs cleared)

Table 20 - Microsoft-Windows-TaskScheduler/Operational Event IDs of Interest

Event ID	Description
106	The user x registered the Task Scheduler task y. (New Scheduled Task)
100	Task Scheduler started the x-th instance of the y task for user z.
102	Task Scheduler successfully finished the x-th instance of the y task for user.

Table 21 - Microsoft-Windows-Windows Defender/Operational Event IDs of Interest

Event ID	Description
1116	The antimalware platform detected malware or other potentially unwanted software.
1117	The antimalware platform performed an action to protect your system from malware or other potentially unwanted software.

Tools

- Windows Event Viewer (built-in)

- EvtxECmd (<https://download.mikestammer.com/EvtxECmd.zip>) – parses to CSV, XML, JSON; is a part of KAPE.
- Chainsaw (<https://github.com/WithSecureLabs/chainsaw>) – uses sigma rules to hunt for relevant logs; a part of the KAPE tool.

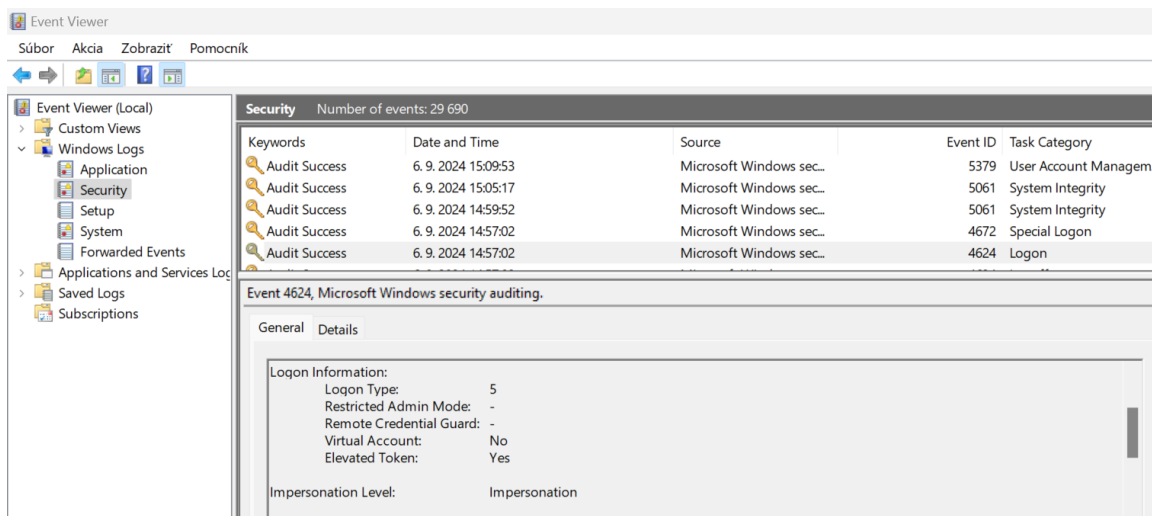


Figure 41: Example of a security event log in Event Viewer

Sources

- <https://redbeardsec.com/windows-event-logs-detection-and-forensics-power/>
- https://13cubed.s3.amazonaws.com/downloads/windows_event_log_cheat_sheet.pdf

5.2. Analysis of Windows system artifacts – process execution

The following artifacts are mainly used to prove a process has been executed. They are crucial in helping investigators understand the sequence of events on a system, identify malicious activities and reconstruct timelines.

Prefetch

Description

- Increases system performance by preloading code pages of commonly used applications.
- The Cache Manager monitors all files and directories referenced by each application or process and maps them into a .pf file.
- It is used to determine that an application has been launched in the system.
- Limited to 128 files in Windows XP and Windows 7 systems.
- Limited to 1024 files in Windows 8, 10 and 11 system.

Location

Windows XP/7/8/10/11: C:\Windows\Prefetch

Interpretation

- Each .pf file contains the last execution time, the number of executions, device drivers and files used by the program.
- At first run, the file named as the executed file and the path to the file is created.
- Creation date of the .pf file (-10 seconds).
- At the last run, a file named as the executed file and the path to the file is created.
- The time of the last execution of the .pf file.
- Date of the last modification of the .pf file (-10 seconds).
- Windows 8-10 contains the last 8 executions.
- Can examine each .pf file and search for recently used file handles or device drivers.

PECmd Tool

Note: tested on Windows 10 using version 1.5.0.0 of the tool.

The PECmd tool enables the analysis and generation of outputs from prefetch files. The tool supports both loading individual files and recursively scanning and loading entire folders recursively, allowing shadow copies of loaded files to be processed as well. With PECmd, outputs can be created in CSV, JSON, and HTML formats.

Example usage:

```
PECmd -f C:\Windows\Prefetch\MFTECMD.EXE-145DB73A.pf --json D:\Evidence
```

where the `-f` parameter specifies the prefetch file for analysis and the `--json` parameter specifies the location of the JSON output.

```
PECmd -d C:\Windows\Prefetch\ --vss --csv D:\Evidence
```

where the `-d` parameter specifies the folder for recursive browsing, the `--vss` switch allows for analysis of shadow copies of files, and the `--csv` parameter specifies the location of the CSV output.

```

C:\Users\Admin\Downloads\PECmd>PECmd -f C:\Windows\Prefetch\MFTECMD.EXE-145DB73A.pf
PECmd version 1.5.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/PECmd

Command line: -f C:\Windows\Prefetch\MFTECMD.EXE-145DB73A.pf

Keywords: temp, tmp

Processing C:\Windows\Prefetch\MFTECMD.EXE-145DB73A.pf

Created on: 2022-02-09 10:25:54
Modified on: 2022-02-09 11:42:40
Last accessed on: 2022-02-09 14:55:44

Executable name: MFTECMD.EXE
Hash: 145DB73A
File size (bytes): 294,466
Version: Windows 10 or Windows 11

Run count: 4
Last run: 2022-02-09 11:42:38
Other run times: 2022-02-09 11:42:28, 2022-02-09 10:26:04, 2022-02-09 10:25:53

Volume information:

#0: Name: \VOLUME{01d81c7a29bc9ec6-b229d252} Serial: B229D252 Created: 2022-02-07 23:26:41 Directories: 52 File references: 150

```

Figure 42: Sample of a loaded prefetch file in PECmd

Sources

- Windows Forensic Artifacts Guide - <https://github.com/Psmths/windows-forensic-artifacts>
- <https://4n6shetty.in/How-Windows-Artifact-Prefetch-Can-Help-in-Digital-Forensics-Investigations-in-Windows-11-Machine>
- SANS DFIR Windows Forensic Analysis poster - <https://www.sans.org/posters/windows-forensic-analysis/>
- Windows Prefetch File (PF) format - [https://github.com/libyal/libscca/blob/main/documentation/Windows%20Prefetch%20File%20\(PF\)%20format.asciidoc](https://github.com/libyal/libscca/blob/main/documentation/Windows%20Prefetch%20File%20(PF)%20format.asciidoc)
- Investigating Prefetch - <https://forensafe.com/blogs/prefetch.html>
- Shashidhar, N., & Novak, D. (2015). Digital forensic analysis on prefetch files. International Journal of Information Security Science, 4(2), 39-49. - <https://dergipark.org.tr/en/download/article-file/147969>
- Budhrani, A., Singh, U., & Singh, B. (2022, December). Forensic Analysis of Windows 11 Prefetch Artifact. In 2022 IEEE Bombay Section Signature Conference (IBSSC) (pp. 1-6). IEEE. - <https://ieeexplore.ieee.org/abstract/document/10037260>

Windows Activity Timeline (Windows 10 Timeline)

Description

Windows 10 records current running and recently used apps and files in a "timeline" accessible via the WIN + TAB key combination. The data is recorded in the SQLite database **ActivitiesCache.db** (with 7 tables). The data is stored for 3–4 days, and when the user is logged on with a Microsoft account the data is stored for up to 30 days.

Location

Windows 10: C:\Users\<profile>\AppData\Local\ConnectedDevicesPlatform\L.<profile>\ActivitiesCache.db

Interpretation

- Information about the executed applications, including the application name, time when the application was launched, and usage time.
- Timestamp of activity creation.
- Timestamp of activity last modification.

WxTCmd Tool

Note: tested on Windows 11 using tool version 0.6.0.0

The WxTCmd²⁶ tool allows you to analyze the database used for the Windows Timeline.

The tool supports processing only one file at a time, while the output is supported in CSV format. The output generates up to 3 CSV files.

Sample usage:

```
WxTCmd -f
"C:\Users\Admin\AppData\Local\ConnectedDevicesPlatform\L.Admin\ActivitiesCache.db" --csv "D:\Evidence"
```

where the `-f` parameter specifies the `ActivitiesCache.db` file to parse and the mandatory `--csv` parameter specifies the location of the output in CSV format.

Sources

- <https://forensafe.com/blogs/windowstimeline.html>
- <https://medium.com/@boutnaru/the-windows-forensic-journey-windows-timeline-9aa0fc5ec044>
- <https://medium.com/@joeforensics/digital-forensics-windows-10-timeline-activitescache-db-82b4bc9f715a>

UserAssist

Description

Every GUI-based program launched from the desktop can be tracked. UserAssist data will include information on whether an application was run from a shortcut (LNK file) or directly from the executable. RunPath entry is encrypted using the **ROT-13**²⁷ algorithm (substitution cipher).

Location (in registry)

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist

Interpretation

- Frequency of program execution -- per user.

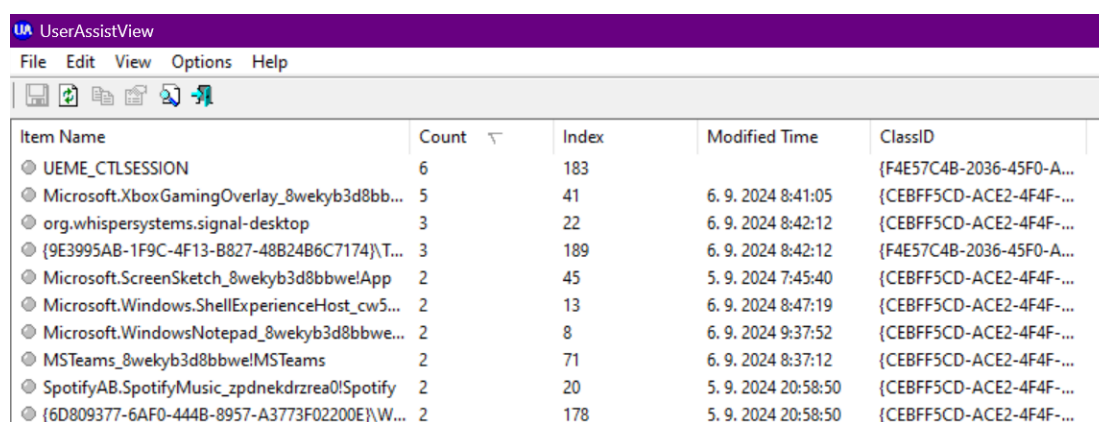
- Last time the program was launched.
- where from the items were being launched most often.
- System date/time changes.
- Evidence of programs after deletion/uninstall.
- How long a user has interacted with a given program (Win7).
- Evidence of absence. (i.e. "items were in a specific location at one time.")

UserAssistView Tool

Note: tested on Windows 11 using version v1.02 of the tool.

This utility²⁸ decrypts and displays the list of all UserAssist entries stored under HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist key in the Registry. You can save the list of UserAssist entries into text/html/xml/csv file.

To display the list of all UserAssist items stored in your Registry, run the UserAssistView.exe by selecting it and double-clicking.



Item Name	Count	Index	Modified Time	ClassID
UEME_CTLSESSION	6	183		{F4E57C4B-2036-45F0-A...
Microsoft.XboxGamingOverlay_8wekyb3d8bb...	5	41	6. 9. 2024 8:41:05	{CEBFF5CD-ACE2-4F4F-...
org.whispersystems.signal-desktop	3	22	6. 9. 2024 8:42:12	{CEBFF5CD-ACE2-4F4F-...
{9E3995AB-1F9C-4F13-B827-48B24B6C7174}\T...	3	189	6. 9. 2024 8:42:12	{F4E57C4B-2036-45F0-A...
Microsoft.ScreenSketch_8wekyb3d8bbwe!App	2	45	5. 9. 2024 7:45:40	{CEBFF5CD-ACE2-4F4F-...
Microsoft.Windows.ShellExperienceHost_cw5...	2	13	6. 9. 2024 8:47:19	{CEBFF5CD-ACE2-4F4F-...
Microsoft.WindowsNotepad_8wekyb3d8bbwe...	2	8	6. 9. 2024 9:37:52	{CEBFF5CD-ACE2-4F4F-...
MSTeams_8wekyb3d8bbwe!MSTeams	2	71	6. 9. 2024 8:37:12	{CEBFF5CD-ACE2-4F4F-...
SpotifyAB.SpotifyMusic_zpdnekdrzrea0!Spotify	2	20	5. 9. 2024 20:58:50	{CEBFF5CD-ACE2-4F4F-...
{6D809377-6AF0-444B-8957-A3773F02200E}\W...	2	178	5. 9. 2024 20:58:50	{CEBFF5CD-ACE2-4F4F-...

Figure 43: Graphical Interface of the UserAssistView tool

Sources

- <https://www.4n6k.com/2013/05/userassist-forensics-timelines.html>
- <https://www.magnetforensics.com/blog/artifact-profile-userassist/>

RecentApps

Description

Program execution launched only on a **Win10** system is tracked in the RecentApps registry key. When a new entry is added to RecentItems, the keys are rearranged alphabetically, removing the last entry from the list.

Location

Windows 10: NTUSER\Software\Microsoft\Windows\Current Version\Search\RecentApps

Interpretation

- LastWriteTime of RecentItems subkeys are the execution times of the file.
- LastAccessedTime in RecentItems is not updated on subsequent accesses to the file.
- Each **GUID** key points to a recent application.

- Path of the item

Tools

- RegRipper (<https://github.com/keydet89/RegRipper3.0>)
- RECmd (<https://download.mikestammer.com/net6/RECmd.zip>)

RegRipper

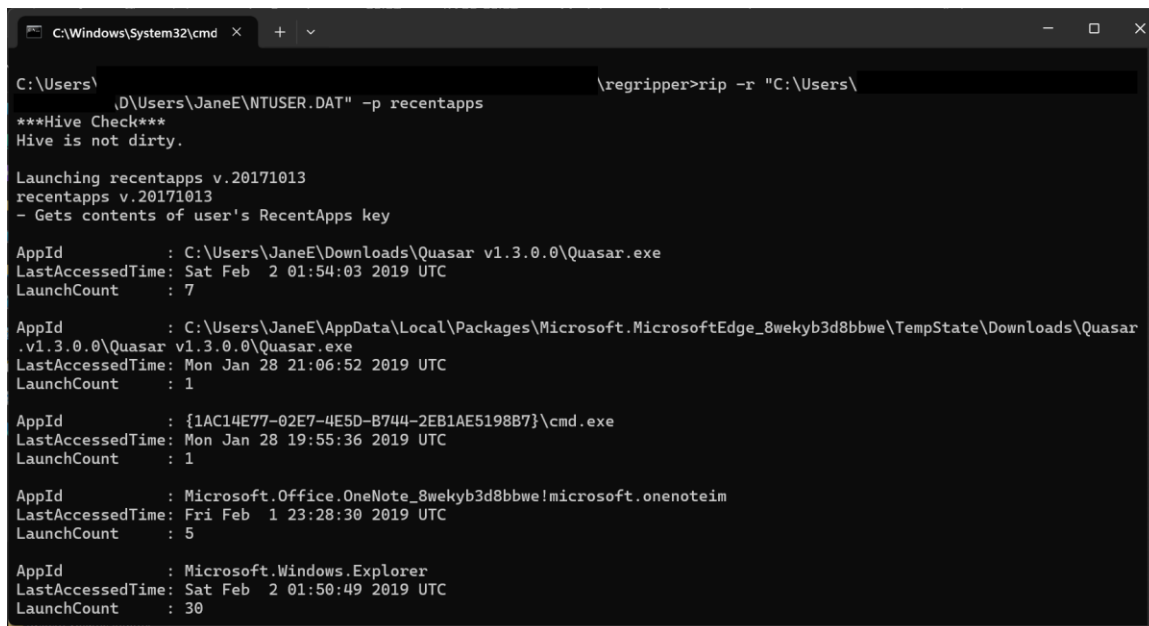
Note: tested on Windows 11 using tool version v.2.8_20200220

Example usage:

```
rip -r "C:\Users\\NTUSER.DAT" -p recentapps
```

where `-r` is the location of the registry and `-p` is a plugin profile to apply.

It is more convenient to run RegRipper as one of the modules in KAPE and have it analyze the registry as a whole.



```
C:\Users\JaneE>rip -r "C:\Users\JaneE\NTUSER.DAT" -p recentapps
***Hive Check***
Hive is not dirty.

Launching recentapps v.20171013
recentapps v.20171013
- Gets contents of user's RecentApps key

AppId      : C:\Users\JaneE\Downloads\Quasar v1.3.0.0\Quasar.exe
LastAccessedTime: Sat Feb  2 01:54:03 2019 UTC
LaunchCount  : 7

AppId      : C:\Users\JaneE\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\TempState\Downloads\Quasar
.v1.3.0.0\Quasar v1.3.0.0\Quasar.exe
LastAccessedTime: Mon Jan 28 21:06:52 2019 UTC
LaunchCount  : 1

AppId      : {1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\cmd.exe
LastAccessedTime: Mon Jan 28 19:55:36 2019 UTC
LaunchCount  : 1

AppId      : Microsoft.Office.OneNote_8wekyb3d8bbwe!microsoft.onenoteim
LastAccessedTime: Fri Feb  1 23:28:30 2019 UTC
LaunchCount  : 5

AppId      : Microsoft.Windows.Explorer
LastAccessedTime: Sat Feb  2 01:50:49 2019 UTC
LaunchCount  : 30
```

Figure 44: Example output from RegRipper using recentapps plugin.

Sources

- <https://nk0.gitbook.io/dfir/windows/forensics/evidence-of-execution/recentapps>

ShimCache (Application Compatibility Cache)

Description

- Designed to provide backward compatibility for programs meant to run on older versions of Windows.
- Tracks executable file name, file size, last modified time, and in Windows XP, last updated time
- The ShimCache registry data is written on system shutdown.
- Any executable file on Windows can be found in this key.

Location (in the Registry)

XP: SYSTEM\CurrentControlSet\Control\SessionManager\AppCompatibility

Windows 7/8/10/11: SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache

Interpretation

- Identify the systems on which specific malware has been run.
- Determine the last time of execution or activity on the system.
- Determine which applications were executed and when.
- Holds information regarding: File Full Path, File Size, \$Standard_Information (SI) Last Modified time.
- Modifications made to the system as a result of application execution (helps with understanding malware actions, which files were created or modified, and which registry keys were accessed).
- Windows XP contains a maximum of 96 entries
- LastUpdateTime is updated when files are executed
- Windows 7 contains a maximum of 1 024 entries
- LastUpdateTime does not exist in Windows 7

AppCompatCacheParser Tool

Note: tested on Windows 10 using tool version 1.5.0.0

The AppCompatCacheParser²⁹ performs a trace extraction from the shimcache to CSV format, with the output containing the path to the file, information about whether the file was run, and the last modification of the record.

Sample uses:

```
AppCompatCacheParser.exe --csv D:\evidence -t
```

where the `-t` switch sorts the records by time in the ascending order. The mandatory parameter `--csv` specifies the location of the output file. The input is not defined, so the registry of the operating system on which the tool was run is used instead.

```
AppCompatCacheParser.exe --csv D:\evidence -f D:\ExtractedData\SYSTEM
```

where parameter `-f` specifies the location of the SYSTEM subregistry.

```
ControlSet      : 1
CacheEntryPosition : 497
Path            : C:\Windows\system32\svchost.exe
LastModifiedTimeUTC : 2021-06-05 12:05:23
Executed        : NA
Duplicate       : False
SourceFile      : Live Registry
```

Figure 45: Example of a shimcache record extracted by AppCompatCacheParser

Sources

- <https://medium.com/@mehrnoosh/shimcache-amcache-forensic-analysis-99a8a9733772>
- <https://github.com/Psmths/windows-forensic-artifacts/blob/main/execution/shimcache.md>

- <https://www.inceptionsecurity.com/post/shimcache-a-crucial-tool-for-digital-forensics-and-incident-response>

Amcache.hve

Description

Amcache is a special registry file which is not usually shown along ProgramDataUpdater (a task associated with the Application Experience Service) , it uses the Amcache.hve registry files to store data during process creation. This registry stores the first execution of a program on the system, including portable programs executed from external storage.

Location

Windows 7/8: C:\Windows\AppCompat\Programs\Amcache.hve

Windows 10/11: C:\Windows\appcompat\Programs\Amcache.hve

Interpretation

- execution paths (including disk volume)
- installed applications and drive
- installation, deletion times
- not a 100% proof of execution - the program may be present just because it was automatically scanned
- SHA1 hashes of the programs
- Time of first execution = time of last key modification

Tools

- AmcacheParser (<https://ericzimmerman.github.io>)
- Registry Explorer (<https://ericzimmerman.github.io>)
- RegRipper (<https://github.com/keydet89/RegRipper3.0>)

AmcacheParser Tool

Note: tested on Windows 10 using version 1.5.1.0 of the tool.

AmcacheParser³⁰ performs the extraction of selected, relevant traces from the amcache.hve file, presenting the output as several CSV files.

Sample usage:

```
AmcacheParser.exe -f C:\Windows\appcompat\Programs\Amcache.hve -i --csv
D:\Evidence
```

where the mandatory parameter `-f` specifies the location of the `Amcache.hve` file. The `-i` switch extracts additional files with additional traces, such as records of programs installed on the operating system and records of files associated with other programs. The mandatory parameter `--csv` specifies the location of the output files.

```

ApplicationName      : Unassociated
ProgramId            : 00067a84bfb14a1e3daef32a8d34ede1f88c00000904
FileKeyLastWriteTimestamp : 2021-12-30 17:04:05
SHA1                 : f0db5615ff6659ce7bd7891e5345217e0e0bba46
IsOsComponent        : False
FullPath             : c:\program files (x86)\microsoft\edgeupdate\microsoftedgeupdate.exe
Name                 : MicrosoftEdgeUpdate.exe
FileExtension        : .exe
LinkDate             : 2021-04-24 21:10:26
ProductName          : microsoft edge update
Size                 : 214936
Version              : 1.3.143.57
ProductVersion       : 1.3.143.57
LongPathHash         :
BinaryType           : pe32_i386
IsPeFile             : False
BinFileVersion       : 1.3.143.57
BinProductVersion    : 1.3.143.57
Usn                  : 4488872
Language             : 1033
Description          :

```

Figure 46: Sample output from AmcacheParser, specifically a record of program execution.

Sources

- <https://medium.com/@mehrnoosh/shimcache-amcache-forensic-analysis-99a8a9733772>
- <https://www.thedfirspot.com/post/evidence-of-program-existence-amcache>

System Resource Usage Monitor (SRUM)

Description

SRUM is intended to track the application usage, network utilization, and system energy state. It records 30 to 60 days of system performance. While SRUM continuously collects data on a running system, the database is updated at regular intervals. By default, the database is updated every hour. It is set to write information 30 minutes after each hour (for example 12:30, 13:30, 14:30 ...) until a system shutdown occurs triggering an immediate database update. On a live system the file SRUDB.dat will be locked.

SRUM can be divided into several different sub-artifacts, which have their attributes and interpretations. For their deeper understanding refer to the sources at the end of this section. These sub-artifacts are Application Resource Usage Artifact, Energy Usage Artifact, Energy Usage (Long Term) Artifact, Network Connections Artifact, Network Usage Artifact and Push Notification Data Artifact.

Location (on Win8+)

- C:\Windows\System32\sru\SRUDB.dat
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SRUM (primary registry key)
- SOFTWARE\Microsoft\WindowsNT\CurrentVersion\SRUM\Extensions (Extension DLLs)
- SOFTWARE\Microsoft\WlanSvc\Interfaces\

Interpretation

- program executions, power consumption, network activities, and much more information that can be retrieved even if the source has been deleted
- Applications started; the user account responsible for it

Scheduled Tasks

Description

The Windows operating system allows the creation of Scheduled Tasks – processes configured to run at predefined times. Scheduled Tasks are one of the most widely used methods to ensure the persistence of a malicious code within the operating system. For example, a scheduled task will test if the malicious code is running on the operating system after a certain point in time, and if not, it will introduce the malicious code into the operating memory.

Scheduled tasks in Windows are stored and managed through both the file system and the Windows Registry.

Location

Windows XP: C:\windows\tasks

Windows 7, Vista: C:\windows\system32\tasks

Windows 8/10/11:

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks

HKLM\Software\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree

Interpretation

- the task itself may lead to a malicious file being executed or an IP address being contacted
- the user who created the task
- the time or trigger for the task to run
- the path to the command or program to be executed

Tools

- Task Scheduler (built-in)
- Autoruns (<https://learn.microsoft.com/en-us/sysinternals/downloads/autoruns>)

Autoun Entry	Description	Publisher	Image Path	Timestamp	VirusTotal
Task Scheduler					
Adobe Acro...	Adobe Reader and Acro...	(Verified) Adobe Inc.	c:\program files (x86)\common file...	31. 7. 2024 19:10	
hewlett-Pac...	BingPopUp	(Verified) HP Inc.	c:\program files (x86)\hp\hp suppo...	25. 6. 1956 10:09	
Microsoft\Of...	Microsoft Office Click-to...	(Verified) Microsoft Corp...	c:\program files\common files\micr...	25. 8. 2024 22:24	
Microsoft\Of...	Microsoft Office Click-to...	(Verified) Microsoft Corp...	c:\program files\common files\micr...	25. 8. 2024 22:24	
Microsoft\Of...	Microsoft Office SDX He...	(Verified) Microsoft Corp...	c:\program files\microsoft office\ro...	25. 8. 2024 23:12	
Microsoft\Of...	Microsoft Office SDX He...	(Verified) Microsoft Corp...	c:\program files\microsoft office\ro...	25. 8. 2024 23:12	
Microsoft\Of...	Office Performance Mon...	(Verified) Microsoft Corp...	c:\program files\microsoft office\ro...	1. 8. 2024 9:33	
Microsoft\Of...	Manages coordination o...	(Verified) Microsoft Corp...	c:\program files\common files\micr...	13. 8. 2024 5:45	
Microsoft\W...	Microsoft Compatibility T...	(Verified) Microsoft Corp...	c:\windows\system32\compattelru...	22. 8. 2011 6:15	
Microsoft\W...	Microsoft Compatibility T...	(Verified) Microsoft Corp...	c:\windows\system32\compattelru...	22. 8. 2011 6:15	
Microsoft\W...	Microsoft Compatibility T...	(Verified) Microsoft Corp...	c:\windows\system32\compattelru...	22. 8. 2011 6:15	
Mozilla\Firef...		(Verified) Mozilla Corpor...	c:\program files\mozilla firefoxi...	29. 8. 2024 11:04	
OneDrive P...	Standalone Updater	(Verified) Microsoft Corp...	c:\program files\microsoft onedriv...	28. 12. 2020 23:21	
OneDrive R...	Standalone Updater	(Verified) Microsoft Corp...	c:\program files\microsoft onedriv...	28. 12. 2020 23:21	

Figure 48: Example of scheduled tasks in Autoruns

Sources

- <https://medium.com/@thismanera/persistence-forensics-windows-scheduled-tasks-t1053-c074aa9c19c6>
- <https://www.cybertriage.com/blog/windows-scheduled-tasks-for-dfir-investigations/>

5.3. Analysis of Windows system artifacts – file-related artifacts

The main forensic value of the following artifacts is information about files – their presence, modifications, location, deletion etc.

Jump Lists

Description

- The Taskbar is designed to allow users to quickly and easily "jump" or access items.
- The Taskbar stores its data in so-called Jump Lists.
- They contain files that are used frequently.
- They contain recently used files.
- The feature can be turned off by regular users in settings.
- In raw format, the timestamps of Jump List files are based on UTC.
- Jump Lists are associated with software applications through Application IDs (App IDs).

Location

Windows 7/8/10: C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations

Windows 11: C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent Items

Interpretation

- The first time the application is run
- Recently accessed files and tasks, including their full paths and associated metadata.
- Creation Time = The first time an entry is added to the AppID file
- Last time the app was started with the file open
- Modification Time = Last time the item was added to the AppID file
- Which programs were actively used or manipulated
- Patterns of access frequency (may reveal user habits, preferences, or potentially malicious behavior)
- Jump List identifiers -> http://www.forensicswiki.org/wiki/List_of_Jump_List_IDs
- Using the Structured Storage Viewer tool, one of the AutomaticDestination jumplist files can be opened.

JLECmd Tool

Note: tested on Windows 10 using version 1.5.0.0 of the tool.

JLECmd³² can analyze individual files as well as entire folders recursively. Extracting .lnk files from popup lists is also a supported functionality.

Output records can be generated in CSV, JSON, and HTML formats.

```
Entry #: 9
MRU: 3
Path: C:\Users\Admin\Downloads\hasher
Pinned: False
Created on: Tue 2022 02 08 16:43:42.500
Last modified: Tue 2022 02 08 18:19:56.274
Hostname: desktop-j65g1i3
Mac Address: 00:15:5d:32:9a:10
Interaction count: 1

--- Lnk information ---
Absolute path: My Computer\C:\Users\Admin\Downloads\hasher
```

Figure 49: A sample record of a folder recently accessed using the JLECmd tool

Sample application:

```
JLECmd.exe -d C:\Users\Admin\AppData\Roaming\Microsoft\Windows\Recent -html
D:\Evidence
```

Where the `-d` parameter recursively scans and parses the specified folder, and the `-html` parameter saves the output as html in the specified folder.

```
JLECmd.exe -d C:\Users\Admin\AppData\Roaming\Microsoft\Windows\Recent -
dumpTo D:\Evidence
```

where the `-dumpTo` parameter saves the exported lnk files to the specified folder.

Sources

- <https://www.magnetforensics.com/blog/exploring-the-significance-of-jump-lists-in-digital-forensic-examinations/>
- <https://dfir.pubpub.org/pub/wfuxlu9v/release/1>

LNK files

Description

An LNK file is a Windows shortcut file that points to an application or an executable file and has the extension .lnk. Each link file has its own MAC times, and each link file contains MAC times of the target file. If the target file exists on the media, then the file will have its own MAC times. These files are created when a user accesses any file. LNK files can be associated with the original files that no longer exist on the target machine. LNK files are user profile specific in that LNK files are recorded per user on the system.

Location

Windows XP: C:\%USERPROFILE%\Recent

Windows 7/8/10/11:

C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\

C:\%USERPROFILE%\AppData\Roaming\Microsoft\Office\Recent\

Note: These are the primary LNK file locations. They can also be found in other locations.

Interpretation

- The creation time of a LNK is the first time the file/directory is opened.
- The original path of the file.
- Timestamps for both the target file and the LNK file itself.
- Volume name, serial number, NetBIOS name, and MAC address of the host where the linked file is stored.
- Network details whether the file was stored on a network share or remote computer.
- File size of the original file.
- Identify files opened from a specific removable USB device but never saved locally to the system.

LECmd Tool

Note: tested on Windows 10 using tool version 1.5.0.0.

The tool³³ allows both single file analysis and analysis of multiple files in a folder. The output can be saved in CSV, XML, HTML, and JSON formats.

Options in the LECmd tool allow us, for example, to process only lnk files referencing locations on removable disks, or to change the date and time format in the output.

Sample usage:

```
LECmd.exe -f C:\Users\Admin\Desktop\ransomware.lnk --json D:\Evidence
```

where the `-f` parameter specifies the file to parse and the `--json` parameter specifies the location of the output in the json format.

```
LECmd.exe -d C:\Users\Admin\Desktop\ -r
```

where the `-d` parameter specifies the folder to be recursively scanned, and the `-r` switch selects the analysis mode for only those lnk files that reference removable disks.

Sources

- <https://www.magnetforensics.com/blog/forensic-analysis-of-lnk-files/>
- <https://dfir.pubpub.org/pub/wfuxlu9v/release/1>

Thumbnails

Description

Most operating systems create thumbnail files to provide a preview of image files. Thumbnails are typically stored in JPEG, PNG, or BMP format. Additionally, previews of documents (.docx, .pptx, .pdf) and video files can be also created. They are stored in a thumbnail cache file called **thumbs.db**.

If the user has deleted the images but has not deleted the thumbs.db file, it will be possible to restore the thumbnail version of the images deleted from this directory. In

addition to the thumbnail version of the image, the thumbs.db file also contains the file name and the date and time of the last modification.

Location

Windows XP/Windows 8

- Automatically created anywhere with a home group enabled

Windows 7/8/10/11

- %LocalAppData%\Microsoft\Windows\Explorer
- Other automatic locations are also possible

Note: usually stored in the same directory as the folder whose thumbnails are cached, though it is not a must.

Interpretation

- Thumbnail image of the original image
- Document thumbnail
- Proof that a file of interest was stored on a Windows system hard drive even if deleted.
- Last modified time (XP only)
- Original file name (XP only)
- The thumbcache file does not store the original path of the files.

Tools

- Thumbcache Viewer (<https://thumbcacheviewer.github.io/>) - extracts thumbnail images from the thumbcache_*.db and iconcache_*.db
- Thumbs Viewer (<https://thumbsviewer.github.io/>) - extracts thumbnail images from the Thumbs.db, ehthumbs.db, ehthumbs_vista.db, Image.db, Video.db, TVThumb.db, and musicThumbs.db database files found on various Windows operating systems.

Sources

- <https://medium.com/@mehrnoosh/thumbnail-forensic-analysis-b094a8a5a68b>

OpenSave MRU

Description

- Tracks files that have been opened or saved within a Windows shell dialog box.
- This key is also responsible for tracking auto-complete terms for that dialog box.
- This data exists so that Windows can give a list of previously used file names.

Location (in HKCU registry)

Windows XP:

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSave MRU

Windows 7/8/9/10/11:

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSave PIDIMRU

Interpretation

- Items that do not have file extensions associated with them
- Auto-complete information for the dialog box in Explorer. Consider an OpenSave dialog box that allows you to choose your file type from a list (e.g. .jpg, .png, .bmp). User input into this dialog will typically be the name of the file without the extension. Therefore, it is stored in the OpenSaveMRU.
- Most recent files of any extension input in an OpenSave dialog.
- File info from the OpenSave dialog by specific extension.
- List file paths and times for files that have since been deleted or were on a removable drive.

Tools

- Registry Editor (built-in)
- Registry Explorer (<https://ericzimmerman.github.io>)
- RegRipper (<https://github.com/keydet89/RegRipper3.0>)

Sources

- <https://www.sans.org/blog/opensavemru-and-lastvisitedmru/>
- <https://www.cybertriage.com/artifact/windows-opensave-mru-artifact/>
- <https://forensafe.com/blogs/opensavemru.html>

Last-visited MRU

Description

Tracks the specific executable used by the application to open files documented in the OpenSaveMRU key. In addition, each value also tracks the directory location for the last file accessed by the application. Data is stored in the binary format.

Example: notepad.exe was last run using the directory C:\%USERPROFILE%\Desktop

Location

Windows XP:

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU

Windows 7/8/10/11:

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidIMRU

Interpretation

- Tracks the application used to open/save files stored in OpenSave MRU
- Tracks the folder location of the last file that was accessed (opened or saved) by the application.

Tools

- Registry Editor (built-in)
- Registry Explorer (<https://ericzimmerman.github.io>)

- RegRipper (<https://github.com/keydet89/RegRipper3.0>)

Sources

- <https://forensafe.com/blogs/lastvisitedmru.html>
- <https://www.sans.org/blog/opensavemru-and-lastvisitedmru/>

Recycle bin

Description

The Recycle Bin is a very important place in the Windows file system. Any file that is removed from Windows is generally placed in the Recycle Bin first (except when anti-forensic techniques are used to permanently remove the file). The Recycle Bin is divided into sub-directories, each of them related to a specific user's SID.

The removed data and metadata are in separate files. The first file begins with the value "\$R" followed by a random string and contains the actual contents of the recycled file. The second file starts with "\$I", ends in the same string as the "\$R" file, and contains the metadata for that specific file. The "\$I" file contains the original filename, path, file size, and when the file was deleted.

Location

Windows Vista+: C:\\$Recycle.bin\%SID%\

Interpretation

- The presence of this artifact indicates that a file has been deleted.
- SID can be mapped to a user using register analysis.
- The \$I file contains the size of the original file, full path, and deletion time and date.
- The \$R file contains the full contents of the original deleted file.
- Maps the file name to the actual name and path from which it was removed

RBCmd Tool

Note: tested on Windows 10 using RBCmd version 1.5.0.0.

RBCmd³⁴ allows an easy analysis of the recycle bin in Windows OS. The tool supports both single file analysis and recursive folder scanning. The output is available in the CSV format.

Sample usage:

```
RBCmd -d C:\$Recycle.Bin --csv D:\Evidence
```

where the CSV parameter `-d` specifies the folder to be recursively browsed and the `--csv` parameter specifies the location of the output in CSV format.

```

C:\Users\Admin\Downloads\RBCmd>RBCmd -d C:\$Recycle.Bin
RBCmd version 1.5.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/RBCmd

Command line: -d C:\$Recycle.Bin

Looking for files in C:\$Recycle.Bin

Found 1 files. Processing...

Source file: C:\$Recycle.Bin\S-1-5-21-1539281265-4126948046-2234257553-1001\I6Y28X5.xlsx

Version: 2 (Windows 10)
File size: 8,694 (8.5KB)
File name: C:\Users\Admin\Desktop\Zosit1.xlsx
Deleted on: 2022-02-09 10:27:56

Processed 1 out of 1 files in 0.0499 seconds

```

Figure 50: Example of the RBCmd tool.

Sources

- <https://medium.com/@mehrnoush/windows-recyclebin-forensics-a0855b957a31>
- <https://andreafortuna.org/2019/09/26/windows-forensics-analysis-of-recycle-bin-artifacts/>

Thumbcache

Description

Thumbnails of images, office documents, and folders exist in a database called a thumbcache. Each user has their database based on the size of the thumbnails the user is viewing (there are up to 15 different size formats for thumbcache databases). Windows 8 and Windows 10 introduced a new database known as **iconcache**, which serves a similar purpose as thumbcache except for icons.

The file names aren't in a human-readable format. To remedy that, it is possible to cross-reference the Windows Search database to map the file paths of images within the Thumbcache database. The Windows Search database is located at C:\ProgramData\Microsoft\Search\Data\Applications\Windows\Windows.edb. Windows 11 stores the same information in SQLite database.

Location

Windows Vista+: C:\%USERPROFILE%\AppData\Local\Microsoft\Windows\Explorer

Interpretation

- Deleted file preview
- Proof of the file's existence
- Full path of not deleted file

Tools

- Thumbcache Viewer – also available in KAPE tool (<https://thumbcacheviewer.github.io/>)

Sources

- <https://windowsforensics.net/database/file-knowledge/thumbcache.html>
- https://forensics.wiki/windows_thumbcache/

5.4. Analysis of Windows system artifacts – web browser artifact analysis

Browser forensics plays a vital role in digital forensics and incident response by offering valuable insights into a user's online activities. This includes information such as websites visited, login credentials, download history, and installed extensions or plugins.

History

Description

Browsing history contains data about the user's browsing activity. It can be used to track whether the user has visited malicious or illegal websites or examine the activities in the case of an insider threat.

Location

- Google Chrome
C:\Users\%USERPROFILE%\AppData\Local\Google\Chrome\User Data\<Profile>\History
- Mozilla Firefox
C:\Users\%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<random text>.default\places.sqlite
- Edge
C:\Users\%USERPROFILE%\AppData\Local\Microsoft\Edge\User Data\Default\History

Interpretation

- ULRs of visited websites and time of the visit
- Search history

BrowsingHistoryView Tool

BrowsingHistoryView³⁵ is a tool that displays the browsing history data of different web browsers (Mozilla Firefox, Google Chrome, Internet Explorer, Microsoft Edge, Opera). The browsing history table contains the following information: URL visited, name, time of visit, number of visits, web browser, and user profile. The browsing history can be exported to a tab-delimited, CSV, HTML, or XML file from the user interface or using the command line.

Sources

- <https://medium.com/@laurent.mandine/browser-forensics-89429fe0749f>

Downloads

Description

Downloaded files are among the important forensic artifacts within web browsers, providing a wealth of information about files the user has downloaded.

Location

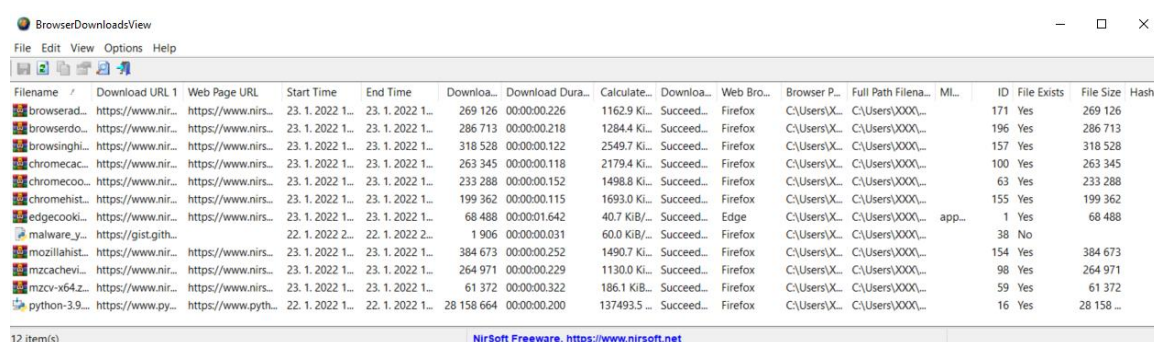
- Mozilla Firefox
C:\Users%\USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\[profileID].default\downloads.sqlite
- Google Chrome
C:\Users%\USERPROFILE%\AppData\Local\Google\Chrome\User Data\[profile]\History
- Microsoft Edge
C:\Users%\USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat

Interpretation

- Downloaded files
- Time of the download
- URL of the download
- File metadata – original location in the file system, filename, file size...

BrowserDownloadsView Tool

BrowserDownloadsView³⁶ is a Windows tool that displays details of downloaded files for Chrome and Firefox web browsers. For each downloaded file, the following information is displayed: file name, download URL, web page URL, start time, end time, download duration, downloaded file size, web browser, and more. You can select one or more items and export them to a comma-separated, tab-delimited, HTML5, XML, or JSON file or calculate the hash value (MD5, SHA1, SHA256, SHA512) of the downloaded files.



Filename	Download URL	Web Page URL	Start Time	End Time	Download...	Download Dura...	Calculate...	Download...	Web Bro...	Browser P...	Full Path Filena...	ML...	ID	File Exists	File Size	Hash
browserad...	https://www.nir...	https://www.nirs...	23.1.2022 1...	23.1.2022 1...	269 126	00:00:00.226	1162.9 Ki...	Succeed...	Firefox	C:\Users\X...	C:\Users\XXX...		171	Yes	269 126	
browserdo...	https://www.nir...	https://www.nirs...	23.1.2022 1...	23.1.2022 1...	286 713	00:00:00.218	1284.4 Ki...	Succeed...	Firefox	C:\Users\X...	C:\Users\XXX...		196	Yes	286 713	
browsinghi...	https://www.nir...	https://www.nirs...	23.1.2022 1...	23.1.2022 1...	318 528	00:00:00.122	2549.7 Ki...	Succeed...	Firefox	C:\Users\X...	C:\Users\XXX...		157	Yes	318 528	
chromecca...	https://www.nir...	https://www.nirs...	23.1.2022 1...	23.1.2022 1...	263 345	00:00:00.118	2179.4 Ki...	Succeed...	Firefox	C:\Users\X...	C:\Users\XXX...		100	Yes	263 345	
chromeccc...	https://www.nir...	https://www.nirs...	23.1.2022 1...	23.1.2022 1...	233 288	00:00:00.152	1498.8 Ki...	Succeed...	Firefox	C:\Users\X...	C:\Users\XXX...		63	Yes	233 288	
chromehist...	https://www.nir...	https://www.nirs...	23.1.2022 1...	23.1.2022 1...	199 362	00:00:00.115	1693.0 Ki...	Succeed...	Firefox	C:\Users\X...	C:\Users\XXX...		155	Yes	199 362	
edgecook...	https://www.nir...	https://www.nirs...	23.1.2022 1...	23.1.2022 1...	68 488	00:00:01.642	40.7 KiB/...	Succeed...	Edge	C:\Users\X...	C:\Users\XXX...	app...	1	Yes	68 488	
malware_y...	https://gist.gith...		22.1.2022 2...	22.1.2022 2...	1 906	00:00:00.031	60.0 KiB/...	Succeed...	Firefox	C:\Users\X...	C:\Users\XXX...		38	No		
mozillahist...	https://www.nir...	https://www.nirs...	23.1.2022 1...	23.1.2022 1...	384 673	00:00:00.252	1490.7 Ki...	Succeed...	Firefox	C:\Users\X...	C:\Users\XXX...		154	Yes	384 673	
mzcachevi...	https://www.nir...	https://www.nirs...	23.1.2022 1...	23.1.2022 1...	264 971	00:00:00.229	1130.0 Ki...	Succeed...	Firefox	C:\Users\X...	C:\Users\XXX...		98	Yes	264 971	
mzcv-x64_z...	https://www.nir...	https://www.nirs...	23.1.2022 1...	23.1.2022 1...	61 372	00:00:00.322	186.1 KiB...	Succeed...	Firefox	C:\Users\X...	C:\Users\XXX...		59	Yes	61 372	
python-3.9...	https://www.pyth...	https://www.pyth...	22.1.2022 1...	22.1.2022 1...	28 158 664	00:00:00.200	137493.5 ...	Succeed...	Firefox	C:\Users\X...	C:\Users\XXX...		16	Yes	28 158 ...	

Figure 51: Example of the BrowserDownloadView tool.

Sources

- <https://www.digitalforensics.com/blog/articles/an-overview-of-web-browser-forensics/>

- <https://www.foxtonforensics.com/web-browser-forensics#downloads>

Cookies

Description

Cookies are small text files that various websites may place on a user's computer or smart device. There are 3 main reasons why cookies are used: tracking, session management, and personalization.

Location

- Mozilla Firefox
- C:\Users\%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\cookies.sqlite
- Google Chrome
- C:\Users\%USERPROFILE%\AppData\Local\Google\Chrome\User Data\<Profile>\Network\Cookies
- Edge
- C:\Users\%USERPROFILE%\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies

Interpretation

- Visited websites
- Time of last cookie use
- The type of cookie may indicate the user logged in to the website

Tools

Depending on the browser, you can use IECookiesView (<https://www.nirsoft.net/utils/iecookies.html>) for Internet Explorer, MZCookiesView (<https://www.nirsoft.net/utils/mzcv.html>) for Mozilla Firefox, or ChromeCookiesView (https://www.nirsoft.net/utils/chrome_cookies_view.html) for Chrome browser. All of these are tools with graphical interface, allowing you to neatly view the cookie attributes, or export the results to CSV, HTML XML, and txt file.

Sources

- <https://www.salvationdata.com/knowledge/cookie-file-forensics-types-and-directives-explained/>

Extensions (add-ons)

Description

Browser extensions may become a source of malicious activity. The extensions need different permissions for their functionality – access to device storage, access to settings, viewing, and even editing the content of web pages, tracking their content, and so on. Therefore, it can be a powerful tool for an attacker. It can be an extension created with a malicious purpose or a vulnerable extension, exploited by an attacker. The extensions should be verified and checked for known vulnerabilities.

Location

- Mozilla Firefox
- C:\Users\%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\extensions.json
- Google Chrome
- C:\Users\%USERPROFILE%\AppData\Local\Google\Chrome\User Data\<Profile>\Extensions\<Extension ID>
- Edge
- C:\Users\%USERPROFILE%\AppData\Local\Microsoft\Edge\User Data\Default\Extensions\<Extension ID>

Interpretation

- All current extensions of a browser.
- Their source files, metadata and settings.

Hindsight Tool

Note: tested on Windows 11 using Hindsight version 1.0.0. To open SQLite database, DB Browser for SQLite Version 3.12.2 was used.

This tool³⁷ allows viewing and parsing different artifacts of chromium-based browsers. It has a simple web interface. The results can be saved in JSON, SQLite, or XLSX format. A summary of identified artifacts is displayed on the website as well as in the command line (Figure 52). We found the SQLite format to be the best for manual viewing. It contains 3 tables – installed_extensions, storage, and timeline. In our testing, the extensions were doubled: 4 were identified, but 8 were shown in the SQLite database, as seen in Figure 53.

```

Profile: C:\Users\HP\AppData\Local\Google\Chrome\User Data\Default
Detected Chrome version: [ 124 ]
URL records: [ 57 ]
Download records: [ 5 ]
Cache records: [ 0 ]
GPU Cache records: [ 0 ]
Autofill records: [ 6 ]
Bookmark records: [ 1 ]
Local Storage records: [ 48 ]
Session Storage records: [ 233 ]
Extensions: [ 4 ]
Login Data records: [ 4 ]
Preference Items: [ 75 ]
Site Characteristics records: [ 45 ]
DIPS Popup Items: [ 0 ]
DIPS Items: [ 18 ]
Cookie records: [ 150 ]
HSTS records: [ 46 ]
Chrome Extension Names (v20240428): - 0 extension URLs parsed -
Generic Timestamp Decoder (v20240428): - 0 timestamps parsed -
Google Analytics Cookie Parser (v20170130): - 0 cookies parsed -
Google Searches (v20160912): - 19 searches parsed -
Load Balancer Cookie Decoder (v20200213): - 0 cookies parsed -
Quantcast Cookie Parser (v20160907): - 0 cookies parsed -
Query String Parser (v20170225): - 12 query strings parsed -
Time Discrepancy Finder (v20170129): - 0 differences parsed -

```

Figure 52: Example of identified browser artifacts.

Database Structure Browse Data Edit Pragmas Execute SQL					
Table: installed_extensions Filter in any column					
name	description	version	app_id	profile	
Filter	Filter	Filter	Filter	Filter	
1 Google Translate	View translations easily as you...	2.0.15	aapbdbdomjkkjkaonfhkkikfgjllc...	Default	
2 Google Docs Offline	Edit, create, and view your ...	1.80.1	ghbmnnjooekpmoecnninlnbdl...	Default	
3 QR Code Generator	QR Code Extension: A faster, ...	1.0.4	lkdokbndiffkmddlfpbjokmfkaf...	Default	
4 Chrome Web Store Payments	Chrome Web Store Payments	1.0.0.6	nmmhkkegccagdldgiimedpiccm...	Default	
5 Google Translate	View translations easily as you...	2.0.15	aapbdbdomjkkjkaonfhkkikfgjllc...	Default	
6 Google Docs Offline	Edit, create, and view your ...	1.80.1	ghbmnnjooekpmoecnninlnbdl...	Default	
7 QR Code Generator	QR Code Extension: A faster, ...	1.0.4	lkdokbndiffkmddlfpbjokmfkaf...	Default	
8 Chrome Web Store Payments	Chrome Web Store Payments	1.0.0.6	nmmhkkegccagdldgiimedpiccm...	Default	

Figure 53: Example of Chrome extensions in SQLite database (extracted by Hindsight).

Sources

- <https://cybernews.com/security/the-cybersecurity-threat-of-browser-extensions/>

5.5. Analysis of Linux system artifacts

Introduction

This document provides a comprehensive list of key Linux artifacts relevant to system administration and forensic analysis. Each artifact is briefly described, highlighting its purpose, location, and significance in system configuration, security, and forensic investigations. The artifacts covered include essential components like partition tables, superblocks, login policies, and various system logs.

Basic tools overview

Before discussing specific artifacts, it's essential to understand the role of basic tools commonly used for viewing and analysing Linux files and logs. These tools include:

- `cat`: Displays the content of files.
- `less`: Allows navigation through large text files.
- `grep`: Searches for specific text patterns within files.
- `nano`: A simple text editor for modifying files.
- `vi`: A text editor used for viewing and editing files.

These tools are foundational for working with Linux artifacts and will be referenced collectively as "basic tools" throughout this document to avoid repetition.

DISCLAIMER

All commands and tools referenced in this document have been tested on Ubuntu Linux. Depending on your system setup, you may need to use elevated privileges (`sudo` or `root` access) to execute certain commands. Keep in mind that while the descriptions and

usage examples are generally applicable, the actual behavior might vary slightly across different Linux distributions.

AnacronFiles

Description

Anacron is a tool for executing commands on a periodic basis, with the interval specified in days. Unlike cron, Anacron does not assume that the machine is always running. Therefore, it is suitable for use on machines that do not operate continuously, allowing for the scheduling of regular tasks such as daily, weekly, and monthly jobs.

Location

- '/etc/anacrontab'
- '/etc/cron.daily/*'
- '/etc/cron.hourly/*'
- '/etc/cron.monthly/*'
- '/etc/cron.weekly/*'
- '/var/spool/anacron/cron.daily'
- '/var/spool/anacron/cron.hourly'
- '/var/spool/anacron/cron.monthly'
- '/var/spool/anacron/cron.weekly'

Analysis Value

Anacron configuration files and logs provide insight into scheduled tasks on systems that do not run continuously. Analysing these files can reveal the timing, frequency, and success of various maintenance scripts, which is critical for understanding system behavior and task execution.

Interpretation notes

Anacron is especially relevant for systems that may be intermittently powered on. The presence and configuration of Anacron tasks can indicate the importance of certain maintenance activities and their execution history on the system.

Tools

Basic tools

- cat, grep - Useful for viewing and analysing artifacts

*anacron*³⁸

- **Purpose:** Anacron is the primary tool responsible for managing and running scheduled tasks defined in the Anacron configuration files. It ensures that tasks which were missed due to system downtime are executed as soon as possible when the system is powered on again.
- **Usage:**
anacron -n
- **Features:**

- -n Run all scheduled jobs immediately that have not been executed yet, but only if they are due.
- -s Serialize execution of jobs (execute them sequentially, one after another).
- -d Run Anacron in debug mode, useful for troubleshooting task execution.

Sources

- <https://www.man7.org/linux/man-pages/man8/anacron.8.html>
- <https://www.digitalocean.com/community/tutorials/how-to-schedule-tasks-with-anacron-on-a-linux-vps>
- <https://askubuntu.com/questions/848610/confused-about-relationship-between-cron-and-anacron>
- <https://www.tecmint.com/cron-vs-anacron-schedule-jobs-using-anacron-on-linux/>
- <https://www.quora.com/What-is-the-similarity-and-difference-between-cron-and-anacron-Which-one-would-you-prefer-to-use>

Authentication logs

Description

The authentication logs are the essential system logs in Linux operating systems, recording all events related to authentication, authorization, and system security. These logs help administrators track logins, failed login attempts, and actions performed with administrative privileges (e.g. `sudo` commands).

Location

- `/var/log/auth.log` (Debian/Ubuntu systems): This file logs all authentication-related events on the system, including `sudo` commands, SSH logins, and other PAM (Pluggable Authentication Module) activities.
- `/var/log/secure` (RedHat/CentOS systems): Similar to `auth.log`, this file logs authentication events, focusing on login attempts, `sudo` usage, and other security-related activities.

Analysis Value

The analysis of authentication logs is crucial for identifying potential security incidents, such as unauthorized access attempts, brute-force attacks, and unusual `sudo` usage. These logs help in creating a timeline of access events, which is vital in forensic investigations. They also assist in verifying compliance with security policies by documenting who accessed the system and what actions were performed with elevated privileges.

Interpretation notes

Regularly reviewing these logs can help identify patterns of failed logins, which may indicate brute-force attacks. The presence of successful logins from unusual IP addresses or at odd hours can be a red flag for unauthorized access. Monitoring `sudo` logs helps in tracking privilege escalation activities, which are crucial for detecting potential misuse of administrative rights. Correlating authentication logs with other system logs, such as `syslog` or kernel logs, can provide a comprehensive view of system activity during an incident.

Tools

Basic tools

- `cat`, `less`, `grep` - Useful for viewing and analysing artifacts

*logwatch*³⁹

- **Purpose:** `logwatch` is a log analysis tool that generates detailed reports based on log files. It helps administrators review system logs efficiently by summarizing events and highlighting potential issues.
- **Usage:**
`logwatch --detail High --service sshd --range today`
- **Features:**
 - `--detail` Sets the level of detail in the report.
 - `--service` Specifies the service to focus on (e.g., `sshd`).
 - `--range` Defines the time range (e.g., `today`, `yesterday`).

Sources

- Nikkel, B. (2021). Practical Linux Forensics: A Guide for Digital Investigators. No starch Press.
- <https://help.ubuntu.com/community/LinuxLogFiles>
- <https://www.digitalocean.com/community/tutorials/how-to-view-and-configure-linux-logs-on-ubuntu-debian-and-centos>
- <https://www.thegeekstuff.com/2011/08/linux-var-log-files/>

Authorized SSH keys

Description

This file contains a list of public SSH keys authorized for passwordless login to the system. Each key corresponds to a remote user or system permitted to access the account.

Location

- `/.ssh/authorized_keys`

Analysis Value

Forensic analysis of this file can reveal who has access to the system and track potential unauthorized access through SSH.

Tools

Basic tools

- `cat` - Useful for viewing and analysing artifacts

Sources

- <https://www.digitalocean.com/community/tutorials/how-to-configure-ssh-key-based-authentication-on-a-linux-server>

Daemon log files

Description

The `/var/log/daemon*` files store logs generated by various system daemons which are background services running on the Linux system. These logs capture the activity, status, and any errors related to these daemons, making them essential for monitoring, troubleshooting, and ensuring the proper functioning of critical services.

Location

- `/var/log/daemon*`

Analysis Value

Analysis of daemon log files can reveal detailed information about the operation and performance of system services, helping to diagnose issues, detect abnormal behavior, and understand the overall health of the system.

Interpretation notes

Care should be taken when interpreting daemon logs as they can contain a high volume of entries. Understanding the context and correlating logs from different sources may be necessary to draw accurate conclusions.

Tools

Basic tools

- `cat, grep` - Useful for viewing and analysing artifacts

Sources

- <https://www.man7.org/linux/man-pages/man8/anacron.8.html>
- <https://help.ubuntu.com/community/LinuxLogFiles>

Email and communication artifacts

Description

- `/var/mail/`: This directory contains the user mailbox files, where each file corresponds to an individual user's stored emails. Depending on the mail server configuration, this file may contain the user's incoming messages in plain text.
- `/var/spool/mail/`: Similar to `/var/mail/`, this directory may hold user emails. In many systems, email is temporarily stored here before being delivered to the user's mailbox or processed by mail clients.

Location

- `'/var/mail/'`
- `'/var/spool/mail/'`

Analysis Value

- `'/var/mail/'`: Provides direct access to user emails, most useful for examining communications in forensic investigations. Emails stored in this directory offer insight into user correspondence, account details, and message content.
- `'/var/spool/mail/'`: This directory contains emails in transit or messages that have not yet been delivered. Analysing them can give insight into pending communications or server issues.

Interpretation Notes

- `'/var/mail/'`: Investigators should be mindful of the structure of user mail files, as emails are often stored in a format such as mbox. The content of emails can be searched for keywords or analyzed to trace communication patterns and interactions.
- `'/var/spool/mail/'`: Emails stored here are usually temporary. However, they can reveal failed delivery attempts or queued messages, providing clues about email server issues or disruptions in user communications.

Tools

Basic tools

- `cat`, `less`, `grep` - Useful for viewing and analysing artifacts

*mboxgrep*⁴⁰:

- **Purpose:** Search and extract specific information from mbox-format email files stored in `/var/mail/` or Thunderbird directories.
- **Usage:**

```
mboxgrep -f sender@example.com /var/mail/username
```

Sources

- <https://superuser.com/questions/306163/what-is-the-you-have-new-mail-message-in-linux-unix>
- https://www.usenix.org/legacy/publications/library/proceedings/lisa93/full_papers/zadok.pdf
- <https://serverfault.com/questions/946912/the-difference-between-mail-in-var-mail-vs-home-user-mail>

Grub bootloader

Description

This is the primary configuration file for the **GRUB** bootloader. It contains boot options, kernel parameters, and other settings that define how the system boots. This file can

provide evidence of modifications to boot parameters, such as changes in kernel versions, boot modes, or additional arguments passed to the kernel.

'/var/log/boot.log': This log file captures messages generated during the system's boot process. It logs the sequence of services and modules initialized during startup, along with any errors or warnings. It is useful for diagnosing boot-related issues, system startup failures, and detecting unauthorized changes to the boot process.

Location

- **'/boot/grub/grub.cfg':** Located in the `/boot/grub/` directory, this file stores the configuration for the GRUB bootloader, responsible for loading the operating system.
- **'/var/log/boot.log':** Found in the `/var/log/` directory, this log file records all boot-time messages for review after the system has started.

Analysis Value

- This file can reveal changes to system boot parameters, such as modifications in the boot order, kernel options, or security-related settings (e.g., disabling SELinux or enabling single-user mode). It is important in forensic investigations where boot tampering or unauthorized kernel changes are suspected.
- **'/var/log/boot.log':** The boot log provides a sequential record of system initialization. By reviewing this log, investigators can identify system startup issues, failed service startups, or unauthorized modifications to the boot process. This file can also help trace the root cause of failed boots or prolonged boot times.

Interpretation Notes

- Any modifications to this file can indicate potential bootloader tampering, which may be used to manipulate system startup, enable insecure boot modes, or bypass certain security features.
- **'/var/log/boot.log':** Repeated errors or warnings in this log may indicate hardware issues, misconfigured services, or tampering with critical system files. Investigators should correlate this file with other logs, such as system logs or kernel logs, to get a complete picture of the system's boot sequence.

Tools

Basic tools

- `cat`, `nano`, `vi` - Useful for viewing, modifying and analysing artifacts

*grub-mkconfig*⁴¹

- **Purpose:** To regenerate the GRUB configuration after making changes to GRUB settings.
- **Usage:**

```
grub-mkconfig -o /boot/grub/grub.cfg
```

*grub2-editenv*⁴²

- **Purpose:** To modify the GRUB environment variables, which impact boot options.
- **Usage:**

```
grub2-editenv list
```

```
grub2-editenv set saved_entry=entry_name
```

Sources

- <https://opensource.com/article/17/3/introduction-grub2-configuration-linux>
- <https://runcloud.io/blog/what-are-linux-logs>

Hostname

Description

Stores the system's hostname, a unique identifier that the system uses to communicate within the network. The hostname is crucial for identifying the machine in various networked environments, whether for system administration or user communication via services like SSH.

Location

- `/etc/hostname`: This file is located directly within the `/etc/` directory. It holds the system's hostname as a simple text entry.

Analysis Value

The system hostname is critical for network administration and system identification. In environments with multiple servers or virtual machines, the hostname helps track and manage different systems.

Interpretation Notes

- Changing the hostname impacts system services that rely on network communication, like DNS or mail servers. It's essential to ensure hostname changes propagate correctly across all network-related services.

Tools

Basic tools

- `cat` - Useful for viewing and analysing artifacts

*hostnamectl*⁴³

- **Usage:**
 - To set or change the hostname.

```
hostnamectl set-hostname new-hostname
```

*hostname*⁴⁴

- **Usage:**
 - To display the current system hostname.

```
hostname
```

Sources

- <https://www.tecmint.com/set-hostname-permanently-in-linux/>
- <https://runcloud.io/blog/change-hostname-in-linux>

Independent server application logs

Description

Many server applications manage their own log files, independent of system logging services like syslog or the systemd journal. These logs typically reside in specific directories, often under `/var/log/`, and can include multiple log files for different subsystems or components, such as technical errors, user authentication, transactions, security alerts, and archived logs.

Location

Commonly found in `/var/log/` or within the application's directory structure. Examples include:

- `/var/log/apache2/` or `/var/log/httpd/` for Apache logs.
- Application-specific directories, such as `nextcloud/data/nextcloud.log` for Nextcloud.

Analysis Value

Independent server logs are crucial for understanding application-specific events, troubleshooting errors, monitoring user activity, and investigating security incidents. These logs provide a detailed view of application behavior, including errors, user interactions, and potential security violations.

Interpretation Notes

- Logs may be spread across different files and locations, requiring thorough examination of each relevant log. Additionally, some applications may store logs in databases rather than text files, necessitating database querying to retrieve log data.

Tools

Basic tools

- `cat`, `less`, `grep`, `vi`, `nano` - Useful for viewing, modifying and analysing artifacts

Sources

- Nikkel, B. (2021). Practical Linux Forensics: A Guide for Digital Investigators. No starch Press.
- <https://httpd.apache.org/docs/current/logs.html>
- <https://coralogix.com/guides/application-performance-monitoring/application-logging-best-practices/>

Independent user application logs

Description

When a user logs into a Linux system, standard system logs are generated, and additional logs specific to the user's activities may also be created. These logs are typically saved in locations specific to the user, including within their home directory and under systemd journal paths. They capture details such as user session activities, application logs, and virtual machine management logs.

Location

- `/var/log/journal/<MACHINEID>/<userID>.journal`
- User's home directory (e.g., `~/.config/`, `~/log/`, and other application-specific directories).

Analysis Value

These logs provide insights into a user's activities on the system, including application usage, virtual machine management, and other session-specific actions. They are crucial for understanding user behavior, tracing specific actions, and investigating potential security incidents.

Interpretation Notes

Logs specific to user activities can be spread across various locations within the user's home directory. It's important to conduct thorough searches for log files and directories to capture the full scope of user-generated data.

Tools

Basic tools

- `cat`, `less`, `grep` - Useful for viewing and analysing artifacts

*journalctl*⁴⁵

- **Usage:**

```
journalctl _UID=<userID>
```

Sources

- Nikkel, B. (2021). *Practical Linux Forensics: A Guide for Digital Investigators*. No starch Press.

Journal config file

Description

The Linux Systemd Journal Configuration determines how the systemd journal service manages log data. It allows administrators to configure various options, including log retention policies, disk space usage, logging levels, and whether logs are stored persistently or temporarily. Proper configuration ensures balanced resource usage while maintaining efficient log management and system monitoring, which is essential for maintaining system integrity and performance.

Location

- `/etc/systemd/journald.conf`

Analysis Value

Understanding the systemd journal configuration is vital for ensuring the proper retention, availability, and integrity of log data, which is crucial during forensic investigations. Properly configured logs can help track system events, user activities, and potential security incidents. The configuration impacts how long logs are stored, how much space they consume, and whether critical data is preserved across reboots, all of which are essential for reconstructing events and analysing system behavior over time. Misconfigurations could lead to data loss or missed evidence in investigations.

Interpretation notes

Misconfiguration may lead to either insufficient logging or excessive disk usage, potentially missing critical events or affecting system performance.

Tools

Basic tools

- nano, vi - Useful for viewing, modifying and analysing artifacts

Sources

- <https://wiki.archlinux.org/title/Systemd/Journal>
- <https://linuxconfig.org/introduction-to-the-systemd-journal>
- <https://www.golinuxcloud.com/systemd-journald-how-logging-works-rhel-7/>

Journal file

Description

Journal files in Linux are used by the systemd journal to record logs in a binary format. These files store logs with additional metadata, making them a crucial resource for forensic analysis. The `journalctl` command is used to view, search, and export these logs, providing insights into system events, program outputs, and potential tampering.

Location

Journal files are typically located in `/var/log/journal/` or another systemd-configured directory.

If the directory `/var/log/journal/` does not exist, system logs are kept in a temporary location: `/run/log/journal/`. In this case, logs are not persistent across reboots unless the `/var/log/journal/` directory is created.

Analysis Value

Journal files offer a detailed and structured log of system activities, including timestamps, program outputs, and system metadata. They are vital for understanding system behavior, verifying log integrity, and identifying suspicious activities.

Interpretation Notes

The binary format and structured metadata within journal files allow for comprehensive analysis, but they require specific tools, such as `journalctl`. Verifying the integrity of these files is essential, as they can be tampered with. Regular extraction and analysis of logs can reveal crucial information for forensic investigations.

Tools

journalctl

- **Purpose:** View and search logs stored by systemd in journal files.
- **Usage:**
 - **View All Logs:**

```
journalctl
```

- **View Logs for a Specific Service:**

```
journalctl -u ssh
```

- **View Logs Between Specific Dates:**

```
journalctl --since "2023-09-01" --until "2023-09-02"
```

Kernel logs

Description

Kernel logs in Linux are crucial for monitoring the core activities of the operating system, including hardware interactions, kernel module management, and system errors related to the kernel's functionality. These logs provide deep insights into system stability, hardware performance, and potential issues at the kernel level, such as kernel panics and driver failures.

Location

- `/var/log/kern.log`.

Analysis Value

Kernel logs are essential for diagnosing hardware-related issues, identifying kernel panics, and troubleshooting driver problems. They provide critical data for understanding the root causes of system crashes and hardware failures, making them indispensable for maintaining system stability.

Interpretation notes

Frequent errors in `kern.log` may point to hardware malfunctions or driver conflicts. Kernel panics and module load/unload messages can help trace the cause of system instability. Regular monitoring can prevent minor issues from escalating into critical failures.

Tools

Basic tools

- `less`, `grep` - Useful for viewing and analysing artifacts

Sources

- Nikkel, B. (2021). *Practical Linux Forensics: A Guide for Digital Investigators*. No starch Press.
- <https://linuxjourney.com/lesson/kernel-logging>

Kernel ring buffer

Description

The Linux kernel maintains a cyclic buffer, known as the Kernel Ring Buffer, that stores kernel messages and logs generated by the kernel and its modules. This buffer has a fixed size and overwrites the oldest entries as new ones are added, making it volatile. Kernel logs are essential for tracking system boot processes, hardware activity, and various kernel subsystems.

Location

- `/dev/kmsg`
- `/proc/kmsg`
- Saved logs: `/var/log/kern.log`, rotated versions of this log may include `kern.log.1`, `kern.log.2.gz`, and so on

Analysis Value

The kernel ring buffer is critical for capturing real-time kernel events, such as boot processes, hardware initialization, and errors. These logs are invaluable for reconstructing system events, analysing hardware configurations, and identifying security issues.

Interpretation Notes

Logs from the ring buffer are volatile and should be captured early in forensic investigations. Tools like `dmesg` can display current contents, but persistent logs must

be retrieved from saved files or through system daemons like rsyslog or systemd-journald.

Tools

*dmesg*⁴⁶

- **Purpose:** dmesg is a command-line tool used to display messages from the kernel ring buffer. It shows kernel boot messages, hardware events, and errors in real time.
- **Usage:**

dmesg

- **Additional Commands:**
 - `dmesg | grep error`: Search for specific errors in the kernel log.
 - `dmesg -T`: Display timestamps in a human-readable format.

*Zless*⁴⁷

- **Purpose:** zless is a pager tool used to view compressed log files, such as rotated kernel logs (`kern.log.1.gz`, `kern.log.2.gz`). It allows you to view the logs without the need to extract them first.
- **Usage:**

zless /var/log/kern.log.1.gz

- **Commands:**
 - Use arrow keys to navigate through the file.
 - Press `/` to search for keywords within the log.

journalctl

- **Purpose:** journalctl is a systemd tool that provides access to both volatile and persistent kernel logs stored by systemd's journal service. It allows for viewing logs related to kernel events, including those captured from the kernel ring buffer.
- **Usage:**

journalctl -k

- **Commands:**
 - `journalctl -k --since today`: View kernel logs from today.
 - `journalctl -k | grep error`: Search for kernel-related errors in systemd logs.

Sources

- Nikkel, B. (2021). Practical Linux Forensics: A Guide for Digital Investigators. No starch Press.
- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/managing_monitoring_and Updating_the_kernel/getting-started-with-kernel-logging_managing-monitoring-and-updating-the-kernel#role-of-printk-on-log-levels-and-kernel-logging_getting-started-with-kernel-logging

The Linux Auditing System

Description

The Linux Auditing System is a kernel feature that generates a detailed audit trail based on a set of rules. It is highly flexible, allowing the capture of security-relevant events, file access, and system calls. The `auditctl` program loads these rules into the kernel, and the `auditd` daemon writes audit records to disk.

Location

- Logs: `/var/log/audit/audit.log`
- Configuration: `/etc/audit/audit.conf`
- Rules: `/etc/audit/audit.rules`, `/etc/audit/rules.d/*.rules`

Analysis Value

The audit logs provide a detailed record of security-relevant events, including file access, system calls, and user activities. These logs are crucial for forensic analysis, helping to trace unauthorized actions, detect security violations, and reconstruct events on a system.

Interpretation Notes

Audit logs are generated based on specific rules, making them highly customizable. Analysts should be aware of the configuration and rules in place, as these determine what gets logged. Log integrity and completeness are crucial, as tampering with these logs can affect the outcomes of the forensic examination.

Tools

*auditctl*⁴⁸

- **Purpose:** `auditctl` is used to manage and control audit rules. These rules dictate what activities or events the audit system should monitor and log. The tool is critical for configuring specific audit requirements, such as tracking file modifications, system calls, or user actions.
- **Usage:**

```
auditctl -w /etc/passwd -p war -k passwd_changes
```

- **Explanation:**
 - `-w /etc/passwd`: Watch the `/etc/passwd` file.
 - `-p war`: Log write, attribute change, and read accesses.
 - `-k passwd_changes`: Assign a key (`passwd_changes`) to identify the audit event for easier searching later.

*auditd*⁴⁹

- **Purpose:** `auditd` is the daemon responsible for writing audit logs to disk. It runs in the background and collects audit data based on rules configured with `auditctl`. It is essential for maintaining a continuous log of security events and system activities.
- **Usage:**

- Start the audit daemon:

```
systemctl start auditd
```

- Check the status of the auditd service:

```
systemctl status auditd
```

*ausearch*⁵⁰

- **Purpose:** ausearch is a command-line tool used to search through audit logs. It provides a way to filter logs based on specific criteria such as event types, time ranges, users, or custom audit keys (set with auditctl). This makes it a valuable tool for forensic analysis and incident investigation.

- **Usage:**

```
ausearch -k passwd_changes
```

- **Explanation:**

- -k passwd_changes: Search logs that are associated with the passwd_changes key.
- -ts today: Search logs from today's date.

*aureport*⁵¹

- **Purpose:** aureport is used to generate summary reports from the audit logs. It can provide statistics on various events, such as user logins, file accesses, or system calls, and is useful for gaining a high-level overview of the system's audit trail.

- **Usage:**

```
aureport -au
```

- **Explanation:**

- -au: Generate a report on authentication events (e.g., user logins).

Linux crontab

Description

Linux Cron configuration files manage scheduled tasks on the system. The `/etc/crontab` file defines system-wide cron jobs, while the `/etc/cron.d/*` directory contains additional cron job files for more granular scheduling. User-specific cron jobs are stored in the `/var/spool/cron/**` directory, with each file corresponding to an individual user's scheduled tasks. These files are crucial for automating maintenance and repetitive tasks on a Linux system.

Location

- `/etc/crontab`
- `/etc/cron.d/*`
- `/var/spool/cron/**`

Analysis Value

Analysis of Cron files provides insight into scheduled tasks and automation on the system. Investigating these files can reveal patterns of system maintenance, automated updates, backups, and user-specific tasks, which is crucial for understanding system behavior and potential security vulnerabilities.

Interpretation notes

Careful interpretation is needed as the cron jobs could be altered by users or attackers to schedule malicious tasks or hide activities within the system.

Tools

Basic tools

- cat, grep - Useful for viewing, modifying and analysing artifacts

crontab⁵²

- **Purpose:** crontab is the command used to list, edit, or remove cron jobs for individual users. It can manage both system-wide cron jobs and user-specific cron jobs.
- **Usage:**

```
crontab -l      # List the current user's cron jobs
crontab -e      # Edit the current user's cron jobs
crontab -e # Edit system-wide cron jobs
```

- **Explanation:**
 - crontab -l: Lists the current user's cron jobs.
 - crontab -e: Opens the current user's cron job file in a text editor for modifications.
 - crontab -e: Allows editing of system-wide cron jobs under root privileges.

Sources

- <https://www.man7.org/linux/man-pages/man8/anacron.8.html>
- <https://www.digitalocean.com/community/tutorials/how-to-use-cron-to-automate-tasks-on-a-vps>
- <https://www.uptimia.com/learn/cron-jobs-in-linux>
- <https://cronitor.io/guides/cron-jobs>

Linux login files

Description

The Linux utmp, wtmp, and btmp files are critical for tracking user and system login information. The /var/run/utmp file records current login sessions and active users, while /var/log/wtmp logs historical events such as logins, logouts, and system reboots. The /var/log/btmp file tracks failed login attempts. Together, these files provide essential data for monitoring user activity, detecting unauthorized access, and supporting security audits.

Location

- /var/log/btmp*
- /var/log/wtmp*
- /var/run/utmp*

Analysis Value

These files are invaluable for forensic analysis, as they provide a timeline of user activity, including successful and failed login attempts, system reboots, and ongoing sessions. Analysing these logs can help in identifying suspicious behavior, such as repeated failed logins or unexpected login times. Furthermore, discrepancies or tampering in these files can indicate potential unauthorized access or malicious activity. They serve as key artifacts in reconstructing the sequence of events during an investigation.

Interpretation notes

Careful interpretation is required as these files can be modified or deleted by attackers to cover tracks. Monitoring these logs regularly and comparing them with other system logs can help in identifying inconsistencies or anomalies.

Tools

*last*⁵³

- **Purpose:** *last* reads from the *wtmp* file and displays a list of users who have logged in or out of the system, as well as system reboots. It shows the time and duration of each login session, providing insight into user activity history.
- **Usage:**

```
last
last reboot
```

- **Explanation:**
 - *last* Lists all logins recorded in */var/log/wtmp*.
 - *last reboot* Shows the history of system reboots.
 - Useful for auditing user activity and system restarts over time.

*who*⁵⁴

- **Purpose:** *who* reads from the *utmp* file to show which users are currently logged into the system. It also displays details like the terminal used, login time, and the originating IP or hostname of the connection.
- **Usage:**

```
who
```

- **Explanation:**
 - *who*: Displays information about currently logged-in users.
 - This tool is essential for live monitoring of active sessions and identifying who is logged into the system at any given moment.

*lastb*⁵⁵

- **Purpose:** *lastb* is used to read the *btmp* file, which logs failed login attempts. This tool is critical for identifying potential security threats, such as brute-force attacks or unauthorized access attempts.
- **Usage:**

```
sudo lastb
```

- **Explanation:**
 - *lastb* Shows all failed login attempts recorded in */var/log/btmp*.
 - Useful for identifying failed login patterns and potential unauthorized access attempts.

utmpdump⁵⁶

- **Purpose:** utmpdump converts the binary utmp, wtmp, and btmp log files into a human-readable format, allowing you to inspect and analyze them directly. This tool is crucial for deeper forensic analysis of these logs, especially when searching for specific events or anomalies.

- **Usage:**

```
sudo utmpdump /var/log/wtmp
sudo utmpdump /var/log/btmp
```

- **Explanation:**

- Converts binary login logs into a text format, making it easier to examine specific entries.
- This tool is especially useful when correlating log data with other system logs or in forensic investigations.

Basic tools

- grep - Useful for viewing and analysing artifacts

Sources

- <https://github.com/libyal/dtformats/blob/main/documentation/Utmp%20login%20records%20format.asciidoc>
- <https://www.man7.org/linux/man-pages/man5/utmp.5.html>
- <https://linux.die.net/man/5/wtmp>
- <https://linuxhandbook.com/utmp-wtmp-btmp/>

Metadata: superblock

Description

The superblock is a vital structure in Linux filesystems, such as ext2, ext3, and ext4, containing essential metadata about the filesystem. This metadata includes the filesystem's configuration, timestamps, unique identifiers, and the current status of the filesystem. It starts at byte offset 1024 (0x400) from the beginning of the filesystem. The presence of the magic string 0xEF53 at byte offset 1080 (0x438) confirms the filesystem type as ext2/3/4.

Location

Superblock is located at byte offset 1024 (0x400) from the start of the filesystem.

Analysis Value

The superblock holds critical information that forensic investigators can use to understand the history and status of a filesystem. Key data such as the filesystem creation timestamp, last mounted time, last written time, and the unique volume UUID can be instrumental in tracing the usage and modifications of the filesystem. This information can help in associating the filesystem with specific systems or users, validating the integrity of the filesystem, and uncovering potential evidence of tampering or unauthorized modifications.

Interpretation notes

- **Filesystem Timestamps:** The creation time, last mount time, and last write time provide a timeline of user activity and system interactions, which can be cross-referenced with other logs to identify patterns or suspicious activities.
- **UUID and Labels:** The unique volume UUID and any user-specified labels can help correlate the filesystem with particular systems or users, aiding in the attribution of actions performed on the filesystem.
- **Mount Point Information:** The directory where the filesystem was last mounted can reveal the context in which the filesystem was used, particularly in multi-user environments or when analysing external storage devices.

Tools

dumpe2fs⁵⁷

- **Purpose:** dumpe2fs is used to display detailed information about the superblock and block group information of ext2/ext3/ext4 filesystems. It provides metadata about the filesystem, such as its state, UUID, mount points, and other critical attributes.
- **Usage:**

```
dumpe2fs /dev/sda1
```

- **Features:**
 - -h: Displays only the superblock information (a summary).
 - Without -h: Outputs complete details, including block group information.

fsstat⁵⁸

- **Purpose:** Part of The Sleuth Kit (TSK), fsstat is used to provide general filesystem information, including superblock data for ext2/ext3/ext4 filesystems. It gives an overview of the filesystem layout, block sizes, inode information, and more, but with less detail than dumpe2fs. *Usage Example:* fsstat /dev/sda1
- **Usage:**

```
fsstat /dev/sda1
```

Displays metadata such as the filesystem type, block sizes, and layout without specific command options.

Sources

- https://ext4.wiki.kernel.org/index.php/Ext4_Disk_Layout
- Nikkel, B. (2021). Practical Linux Forensics: A Guide for Digital Investigators. No starch Press.

Network interface configuration

Description

This file configures network interfaces on non-systemd Linux systems (e.g., Debian). It defines how network interfaces should behave, whether they use static IP addresses or DHCP for dynamic configuration.

Location

- /etc/network/interfaces

Analysis Value

Reviewing this file helps in understanding the system's network configuration, including IP addresses, subnet masks, gateways, and DNS settings.

Tools

Basic tools

- cat - Useful for viewing and analysing artifacts

Sources

- <https://wiki.debian.org/NetworkConfiguration>
- <https://www.cyberciti.biz/faq/setting-up-an-network-interfaces-file/>

Partition tables

Description

Partition tables define how storage media are divided into partitions, providing details like partition type, size, and offset. Common partition schemes include DOS/MBR, GPT, BSD, and others. These partitions can host various filesystems, with typical Linux partitions including /boot/, /, /home/, swap, and more. Understanding partition tables is crucial in forensics to identify storage layouts and potential hidden data.

Location

Partition table entries can be found in:

- /dev/sda, /dev/sdb, etc. (for SATA/SAS drives)
- /dev/nvme0n1, etc. (for NVMe drives)
- /dev/mmcbk0, etc. (for SD cards)

Analysis Value

Partition tables offer valuable insights into how storage media are organized, making them indispensable in system analysis and digital forensics. Key values in the analysis include:

- **Storage Structure Identification:** Understanding the layout of partitions helps in determining how the storage is intended to be used, including the allocation of space for different filesystems.
- **Detection of Hidden Data:** Gaps between partitions or unusual partition layouts can indicate hidden data or deliberately obfuscated information.
- **Assessment of Misconfigurations:** Analysing partition tables can reveal misconfigured partitions that might lead to data loss, system instability, or security vulnerabilities.

Interpretation notes

Partition types may not always reflect the actual filesystem in use. Incorrect or misleading partition types can be used to hide or obfuscate data. Tools like `mmls` and `disktype` can help in analysing and interpreting partition tables.

Tools

`mmls`⁵⁹

- **Purpose:** Lists partition layouts and metadata from a disk or disk image, useful for forensic analysis.
- **Usage:**

```
mmls /dev/sda
mmls image.dd
```

- Outputs a list of partitions with details like start and end sectors.

`disktype`⁶⁰

- **Purpose:** Analyses disk contents to identify partitions and filesystems.
- **Usage:**

```
disktype /dev/sda
disktype image.dd
```

- Provides details about partitions and the filesystems they contain.

`fdisk`⁶¹

- **Purpose:** Manages partition tables on MBR disks (and GPT, with limitations).
- **Usage:**

```
fdisk /dev/sda
```

- **Features:** p (print partition table), n (new partition), d (delete partition), w (write changes).

`gdisk`⁶²

- **Purpose:** Similar to `fdisk` but designed for managing GPT partition tables.
- **Usage:**

```
gdisk /dev/sda
```

`parted`⁶³

- **Purpose:** Manages partitions on both MBR and GPT disks with a more advanced feature set.
- **Usage:**

```
parted /dev/sda
```

- **Commands:** print (show partition table), mkpart (create a partition), rm (remove a partition).

Sources

- <https://thestarman.pcministry.com/asm/mbr/PartTypes.htm>
- <https://recoverit.wondershare.com/partition-tips/what-is-partition-table.html>
- <https://itfreetraining.com/lesson/mbr-gpt/>
- Nikkel, B. (2021). Practical Linux Forensics: A Guide for Digital Investigators. No starch Press.

Syslog configuration

Description

Syslog configuration files are crucial in defining how logs are managed, stored, and transmitted in the Linux environment. These configurations specify where logs are saved, the severity levels captured, and whether logs are sent to remote servers. Understanding these configurations is essential in a forensic investigation in order to trace log data and identify potential log tampering or misconfigurations.

Location

- /etc/syslog.conf
- /etc/rsyslog.conf
- /etc/rsyslog.d/*.conf
- /etc/syslog-ng.conf
- /etc/syslog-ng/*

Analysis Value

These configuration files determine the location and granularity of log data, influencing the availability and integrity of logs for forensic analysis. They also indicate if logs are stored remotely, which is vital for cross-referencing logs and detecting discrepancies.

Interpretation Notes

Examining these files helps verify whether all relevant logs are being captured and stored securely. Any discrepancies between local and remote logs could indicate potential tampering or unauthorized modifications.

Tools

Basic tools

- cat, less, vi, nano - Useful for viewing and analysing artifacts

*logger*⁶⁴

- **Purpose:** logger is a utility that allows you to manually send test log messages to the syslog. It is particularly useful for verifying whether logs are being correctly captured and routed based on the syslog configuration. After using the logger, check the corresponding log files (e.g., /var/log/syslog) to verify that the message was logged correctly.
- **Usage:**

```
logger "Test message for syslog configuration"
```

Sources

- Nikkel, B. (2021). Practical Linux Forensics: A Guide for Digital Investigators. No starch Press.

Sources

- Nikkel, B. (2021). Practical Linux Forensics: A Guide for Digital Investigators. No starch Press.
- https://www.rsyslog.com/doc/v8-stable/configuration/basic_structure.html
- <https://www.ibm.com/docs/en/ztpf/2023?topic=daemon-syslog-configuration-file>

System logs

Description

System logs in operating system Linux, specifically `/var/log/syslog` or `/var/log/messages`, are vital for tracking general system activities, including boot processes, kernel events, service start/stop operations, and various informational messages. These logs capture a wide array of system events, crucial for system monitoring, troubleshooting, and forensic investigations.

Location

- `/var/log/syslog` (Debian/Ubuntu systems): This file logs all general system messages, including those related to system services, network activities, and non-critical errors.
- `/var/log/messages` (RedHat/CentOS systems): Similar to `syslog`, this file records system-wide events, focusing on the general operation of the system, including hardware and kernel messages.

Analysis Value

Analysing system logs is essential for understanding the overall health and performance of the system. These logs can reveal patterns of system crashes, hardware failures, or misconfigurations. They also provide insights into system boots, reboots, and shutdowns, which are crucial for constructing a timeline of system activities. System logs are invaluable for detecting anomalies, such as repeated service failures or unexpected reboots, which could indicate security incidents or system instability.

Interpretation notes

Frequent or recurring error messages in system logs can indicate underlying hardware issues or software bugs. Analysing boot messages can help identify problems that occur during system startup, such as failing services or missing dependencies. System logs should be correlated with other logs, such as authentication logs and application-specific logs, to get a complete picture of system behavior. Monitoring these logs helps in proactive system maintenance and early detection of potential issues.

Tools

Basic tools

- `less`, `grep` - Useful for viewing and analysing artifacts

logwatch

- **Purpose:** logwatch is a log monitoring and reporting tool. It scans system log files and generates daily summaries or detailed reports about system activities. It is useful for quickly reviewing key events, errors, or warnings from system logs.
- **Usage:**

```
logwatch --detail High --service all --range today
```

- **Features:**
 - `--detail High`: Provides detailed information in the report.
 - `--service <service_name>`: Focuses on specific services (e.g., sshd, postfix).
 - `--range <range>`: Specifies the time range for the logs (e.g., "today", "yesterday").

Sources

- <https://www.thegeekstuff.com/2011/08/linux-var-log-files/>
- <https://www.freedesktop.org/software/systemd/man/journalctl.html>
- Nikkel, B. (2021). Practical Linux Forensics: A Guide for Digital Investigators. No starch Press.

Systemd

Description

Contains configuration files related to systemd, which is the service and system manager used in most modern Linux distributions. It controls the startup, management, and monitoring of services through unit files such as `.service`, `.target`, and `.timer`.

Location

/usr/lib/systemd/: This directory contains the default unit files for systemd-managed services, targets, and timers. These configuration files govern how services behave across different system states, such as boot and shutdown.

Analysis Value

The files in `/usr/lib/systemd/` define how services are started, stopped, and managed. They are essential for troubleshooting system boot problems, ensuring proper service configurations, and automating system tasks.

Interpretation Notes

The files in this directory should generally not be modified directly. Instead, overrides or custom configurations should be placed in `/etc/systemd/system/`. This ensures the system can retain its original service files while allowing specific configurations to be safely overridden.

Tools

Systemctl⁶⁵

- **Usage:**
 - To view active services:

```
systemctl list-units --type=service
```

 - To start or stop a service:

```
systemctl start service-name
systemctl stop service-name
```

journalctl

- **Usage:**
 - To view systemd logs, useful for diagnosing service startup issues.

```
journalctl -xe
```

Sources

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/system_administrators_guide/chap-managing_services_with_systemd
- Nikkel, B. (2021). Practical Linux Forensics: A Guide for Digital Investigators. No starch Press.

5.6. Analysis of MacOS system artifacts

Introduction

This section discusses the artifacts that can be found on a medium running the MacOS operating system and how to interpret them during IT Forensic analysis using available tools.

DISCLAIMER

The commands presented below have been tested in the MacOS 14.5 (Sonoma) environment. In earlier versions of the system there may be certain discrepancies related to the change in the system architecture. In such cases, it is recommended to get acquainted with the system documentation.

Data format

Files containing data used during IT Forensic analysis are mostly saved in two formats:

- property list files (.plist) – they are commonly found in the Preferences directories of a MacOS operating system. These files are in the binary format, but can be also found in a text-based format. Most IT forensic software accepts the data contained in these types of files. An alternative solution is to use programs such as Apple TextEdit;
- SQLite databases (.db) - SQLite is a self-contained, cross-platform relational database management system. SQLite databases are commonly used for storing large amounts of data in MacOS. To browse such files, you can use tools such as DB Browser for SQLite as a last resort. However, it should be noted that they may negatively impact on the excess data remaining in the journal files.

Data location

The most important place is the User's Library Directory which contains resources, configuration settings, and data specific to each Mac user, so it can be extremely

valuable in forensic investigations (make sure to distinguish the User Library from local or system library catalogues, as they should be used for different purposes).

Artifacts of importance for forensic analysis are most often found in a few specific subdirectories. These include:

- Application Support - it is a repository for application-specific data and support files;
- Caches - it stores temporary files used to speed up processes and applications;
- Containers – usually used by sandboxed applications, each container isolates the application's data.
- Group Containers – this is similar to Containers, but for applications that share data between multiple apps or extensions;
- Preferences - it contains user preferences and settings for applications and the system.

Artifacts

Below is a description of system artifacts that may be useful during an IT forensic investigation:

Safari

It is a web browser created by Apple for the OS X operating system. In the main Safari folder located at the root of the user's Library several important artifacts can be found:

- **Browsing History** (artifact location: ~/Library/Safari/History.db)

The history of web browsing using the Safari browser is stored in the History.db file, which is an SQLite database. This file contains the following data:

- 1) Visited websites - URLs and titles of visited websites, the number of times each site was visited, dates and times of these visits;
- 2) Downloaded files - source URL, file name and download time;
- 3) Redirections - source URL, target URL and redirection time.

- **Browsing Preferences** (artifact location: ~/Library/Safari/PerSitePreferences.db)

This database is a source of forensic data regarding user interactions with websites on macOS and iOS devices. It stores user-configured settings for individual websites, such as notification permissions, and camera and microphone access permissions, providing a detailed view of user preferences and permissions for each site. What is more, for users signed in with their iCloud account, the database also includes information on websites visited from other devices.

- **Recently closed tabs** (artifact location: ~/Library/Safari/RecentlyClosedTabs.plist)

This file is used by the web browser to keep track of tabs that users have recently opened and then closed. It not only records which tabs were closed but also logs the specific dates when each of these tabs was closed by the user.

Mail

This natively installed macOS app can provide valuable information about user communications, including sent and received emails, attachments, and account configurations.

Mail data is stored in the directory ~/Library/Mail. It contains subdirectories for the different versions of Mail and email accounts configured on the device. Inside the

directory with the version number, there is a dedicated folder for each mail account that has been set up in the Apple Mail application. The MailData folder, which is also part of Apple Mail's directory, contains various information that is important for the operation of the Mail application and its associated accounts.

In this application, each configured mail account has specific directories associated with it. "Attachments Folder" stores all the attachments received in the mail. "Messages Folder" contains all the individual email messages in raw EMLX format, which are sequentially numbered (the number assigned to each message in the Messages folder can be used to easily locate its corresponding attachments in the Attachments folder).

The Envelope Index is a SQLite database in Apple Mail that contains key elements of email messages and their attachments. This database includes several notable tables that store different types of email-related information. These tables include information about the email addresses involved in the messages:

- Details of files attached to the emails.
- The subject lines of the emails.
- Brief summaries or previews of the email content.
- The actual text and other details of the email messages themselves.

IT forensic examiner should keep in mind that each new version of macOS may introduce changes to the way Mail stores and manages data. This means that tools and techniques that worked in one version of the system may need to be updated or adapted to handle new versions.

Messages

This is a native instant messaging application, which can be used with various different protocols. The messages are stored in the database chat.db located in user's directory ~/Library/Messages. This database is in sqlite format and includes messages and their metadata (for example identification of conversation participants, indication of whether messages were sent using iMessage or SMS, a phone number or email used for the last message and the timestamp for when the last message was read).

The Attachments subfolder contains a range of sub-folders. They store files that have been sent through Messages. The names of these sub-folders correspond to Unique IDs (UIDs) that can be found in the Messages SQLite database chat.db.

Operating system

The artifacts presented below will allow IT Forensic Examiner to obtain as much data as possible about the operating system being examined:

- /System/Library/CoreServices/SystemVersion.plist - OS version, build version.
- /private/var/root/Library/Caches/locations/cache_encryptedA.db - serial number.
- /private/var/log/install.log - OS installation date.
- /Library/Preferences/org.cups.printers.plist - installed printers list.
- /Library/Preferences/com.apple.alf.plist – Application-Level Firewall configuration.
- /Library/Preferences/SystemConfiguration/com.apple.accounts.exists.plist - the users account details and configuration
- /private/var/folders/*/<DARWIN_USER_DIR>/cache_encrypted*.db - network,
- /Library/Preferences/com.apple.MobileBluetooth.devices.plist – the list of paired Bluetooth devices.
- /Library/Preferences/com.apple.MobileBluetooth.ledevices.paired.plist -the list of paired Bluetooth low energy devices.
- ~/Library/Preferences/com.apple.iPod.plist - the list of all Apple devices that have been attached to the system.

Keychain

The keychains on a system are used to store sensitive data such as usernames, passwords, and encryption keys. The Default login.keychain-db password is the user's account password. file located in the folder /Library/Keychains/ may contain user passwords for access points, Time Machine, applications, and websites (default login.keychain-db password is the user's account password). The System.keychain file contains passwords for VPNs, access points, Time Machine, and applications. It is located in /Library/Keychains directory. What is most important, the data in Keychain is encrypted and protected by the user's password. To access this data, IT Examiner must know the user's password or access the master key from the system's memory.

Unified logs

A unified log format for all Apple products (beginning with macOS 10.12) has replaced many of the old log formats that Apple used.

Unified Log files are located in directories such as /private/var/db/diagnostics and /private/var/db/uuidtext. They consist of the following three files:

- Tracev3 files - binary files containing log entries.
- UUID files - binary files containing metadata for log entries.
- Timesync files - binary files containing time-related metadata associated with log entries.

The Unified Logs contain a large amount of valuable information such as process and library path associated with the log entry, PID, EUID, timestamp for the log entry and its type, event type, subsystem, category and log message. Each log entry is associated with the event type and, depending on the Event Type, with a Log Type (also called a Message Type). Event Type can take values such as log, activity, loss or trace.

Due to the huge amount of data that can be found here, effective browsing of Unified Logs requires that filters should be applied to the entire data set and that the time period or type of events that are of interest should be indicated. If we have access to the evidence machine from the perspective of an administrator account, in order to export data from Unified Logs on a working macOS system the following methods can be used:

- *console application* – it is built natively into the operating system. It allows you to view and export logs. You can open the Console application, select the appropriate logs, and then export them to the external device.
- *log command* - using this command in the terminal allows for exporting logs to a file. You can export all logs to a text file (using command: `log show --info --debug > unified_logs.txt`) or apply criteria such as time, process, or subsystem (example of filtering by subsystem: `log show --predicate 'subsystem == "com.apple.network"' --info --debug > network_logs.txt`).
- *third-party software* - software such as Cellebrite Digital Collector or Sumuri Recon ITR allows, when run on an evidence machine, for exporting all or a specified range of logs to an external medium using its GUI.

Time machine backups

A built-in backup feature in macOS that regularly creates backups of the system to an external or network drive. These backups include user files, system settings, and installed applications (they usually can be found in the directories like /Volumes/BackupDrive/Backups.backupdb/). They allow you to attempt to recover deleted files, access older versions of documents, and understand the states of the evidence system. Time Machine Backup files are also used to determine which files may have been deleted or changed.

Time Machine settings can be found in the file `com.apple.TimeMachine.plist` located in the directory `/Library/Preferences`. This file contains various Time Machine settings and preferences, such as:

- *AutoBackup* - enables automatic backups.
- *BackupAlias* - alias for the backup location.
- *BackupAliasVolumeName* - name of the backup volume.
- *Destinations* - list of backup destinations, including IDs and snapshot dates.
- *SkipPaths* - paths that are excluded from backups.

Unified logs reveal when the backup started and finished, highlight networks or local locations of backup, volume name backed up, amount of data backed up, and deletion of old backups (to show backup info use `log show -info -predicate 'process = "backupd"`). Once extracted, the contents of Time Machine backups can be analysed using both commercial (Magnet Axiom) and open source (macosac) tools.

Spotlight

This tool indexes files and metadata on macOS, storing the data in a hidden database containing information on file paths, creation dates, and content snippets.

Spotlight stores its indexes in a metadata database located in the directory called `/private/var/db/Spotlight-V100`. This database contains information about files, such as names, paths, modification dates, file types, and other attributes. Spotlight database analysis can help recover metadata from files that have been deleted because even if a file has been deleted, its metadata may still be present for some time.

After connecting an external drive, Spotlight automatically starts indexing its contents as long as the drive is not excluded from indexing. The indexing includes files, folders and their metadata. Spotlight indexes for external drives are stored on the drive itself in a hidden folder called `/Spotlight-V100`.

Trash

Any files or folders deleted by the user are saved into a hidden Trash folder in the root of that user's home directory (`~/Trash`). Some trashed files can be restored using the "Put Back" option. If the file has this option, the data can be found in the `.DS_Store` file in Trash (it is a binary, hidden file). There is also a system Trash folder for files deleted from external drives and other volumes `/Volumes/[VolumeName]/Trashes/[UserID]`.

Volumes

These are the logical units of data storage, which can include both internal and external drives. Volume analysis provides information about file structure and user activity. The macOS operating system has implemented several mechanisms related to their operation, which can provide valuable data from the standpoint of IT forensics.

1. **Fsevents** - File System Events Store Database stores file system changes and includes events such as file/folder creation, renaming actions, unzipping of files, item deletion, Trash being emptied, and volumes being mounted and unmounted. This information is stored in the `/fsevents` folder on each volume (local or external). Fsevents files are compressed and store file system events in the binary format. Each file contains multiple event records that are sorted alphabetically by full file path. Each event record consists of three main components such as full path, event ID and record flags (the type of object and the nature of the change). We should remember that it can be wiped during a system crash or a hard power failure, and tracks only changes on HFS and APFS.

volumes although we may see a directory on FAT volumes (it is worthwhile to mention that events do not have associated timestamps).

2. **Document versions** - it automatically backups certain types of documents or restores documents after a system crash. Versions are created when a document is saved, opened, every hour a document is open, and when it is frequently being edited. This feature is only supported by certain applications. The version data for documents is stored in hidden directories within the user's Library folder. Specifically, versions are stored in `/System/Volume/Data/DocumentRevisions-V100/`. Historical versions of files are saved in Chunk Storage but Document Versions are only found on HFS+ and APFS-formatted volumes.
3. **Recent access** - when it comes to artifacts related to opening folders and files, the following can be indicated here:
 - *"Recent Items"* - where the list of items recently accessed by the user account is stored, per application. The list contain both native and third-party applications and it is located in this place `~/Library/Application Support/com.apple.sharedfilelist/* .sfl2`.
 - *"Recent Folders"* - where the list of folders recently accessed by the user account is stored. It is located in `~/Library/Preferences/com.apple.finder.plist` subfolder. The related information is stored inside by `FXRecentFolders` key. It contains the name of the folder, its full path and the date and time when the folder was last accessed.
 - *"Most Recently Used"* - the MRU plist files are used to store information about the most recently accessed files, applications, documents, servers, and other items. MRU files are typically located in the user's directory. They can be in the XML or binary format and contain arrays of dictionaries, where each dictionary represents a recently accessed item. Each dictionary typically includes keys such as the name of the item, its path or URL, the date and time when the items were last accessed and additional metadata about the item.
 - *"File Last Used"* - This extended attribute is updated when a file is used in the Finder window or if the file is opened using the "open" command in the Terminal. We can find this information by viewing extended attributes for a specific file using the command `xattr -xl <file name>`.
 - *"USB Mass Storage Class"* - this identifier can be used to find the USB device connections in the Unified Logs, including the device's serial number, vendor, and product information. We can use the command `log show --predicate "eventMessage contains 'USBMSC'" --info` to see what kind of external USB devices was attached to the operating system being examined as evidence.
 - *"Mounted volumes"* - there are various locations where you can find the logs which record what volumes were mounted on the system and can include the device file the volume is using, volume size, name, and mount point. You can search for `/Volumes/` to find any volumes mounted under the default mount point and also search system.log and unified logs for `apfs`, `hfs`, `mounted`, `unmounted`, or `disk#s#`. What is more, the daily logs record located in `/var/log/daily.out` file reveals what volumes were mounted on the system when the daily maintenance script was running.

Forensic Tools

As regards computer forensics software that can help us analyze the above-mentioned MacOS artifacts, the following commercial solutions are available: **Magnet Axiom** and **Belkasoft Evidence Center X** (both enable the analysis of data collected from devices working under the control of other operating systems like Windows and Linux). Among open-source solutions, of particular interest is **Autopsy** and a tool called **mac_apt**, which

is basically a set of python scripts prepared to process Apple computer full disk images and extract data useful for forensic investigation.

5.7. Analysis of mobile artifacts

Introduction

This material presents guidance on how to approach the process of acquisition and analysis of mobile devices. The main objective is to collect data related to user activity, avoiding interference with evidence to the greatest extent possible. Currently, the market is dominated by devices running under Android and iOS operating systems, and information about artifacts that can be found on these devices is given below.

DISCLAIMER

This instruction is based on the assumption that IT Forensic Examiner has a PIN code giving full access to the device whose memory contents were to be analysed, and that the device itself was made available to him in accordance with the legal provisions in force in an organisational unit.

The information below applies to Android 14 and iOS 16 operating system environments. In earlier or later versions of the system, there may be discrepancies related to the change in the system architecture. In such cases, it is recommended to get acquainted with the system documentation.

Preparation

First, it is important that the evidence device should be isolated forthwith from the networks to which it is usually connected. This action prevents new data from modifying the existing evidence and allows for the data to be preserved. There is a risk that if the device remains connected to the network, the user data contained in it may be destroyed, because a mobile device can be remotely wiped using a remote signal.

The easiest way to isolate an evidence device is to switch to the airplane mode — this disables wireless and cellular connections. Alternatively, the device can be placed in a Faraday bag, a container made of a metal shield that blocks radio frequencies used by cellular networks, Bluetooth, and Wi-Fi networks. In this case, it is important to keep the battery in the device charged to ensure it should not switch off during storage or transport.

The device should remain in the state in which it was when transferred. If the device was turned off, it should remain that way. However, if it was transferred on, every effort should be made to ensure that it remains in that state. This is especially important if the device is password protected and the investigator does not have the code: when the device switches off, it will go from AFU (After First Unlock) to BFU (Before First Unlock) state, thus rendering data collection much more difficult, and in some cases impossible.

After successfully seizing and isolating the device, utmost care should be taken to maintain the device's chain of custody and prepare photo documentation of the device (technical condition, main user account details, installed applications). Detailed

instructions on how to proceed in this case are provided in the Digital Evidence Acquisition chapter.

Type of extraction

Acquisition is the process of obtaining information and evidence from a device.

- **logical acquisition** - is usually the easiest and fastest way to extract the most common types of evidence from a mobile device. The main limitation in a logical acquisition is the depth and quality of the extracted data ; although it will include most user-generated content, such as text messages, call logs, photos, and notes, and will not include other precious artifacts, such as emails, location data, third-party apps, and precise system logs.
- **full filesystem acquisition** – it gives investigators access to much more user data on mobile devices than in a standard logical extraction. By acquiring the full filesystem, IT Forensic Investigators will also gain access to many of the system artifacts and databases in which the operating system or third-party applications store key information (in this scenario recovery of some deleted artifacts can also be attempted).
- **physical acquisition** - is a bit-by-bit copy of the mobile device's memory. This is the best method, but it is not always possible. Since data on mobile devices has been encrypted, using traditional techniques such as imaging the physical volume would result in the acquisition of encrypted data, which would be completely useless (in the case of devices running iOS, this is currently virtually impossible).

The state of the device is of great importance in the context of the scope of data that can be read from the evidence mobile device. Generally, two states are distinguished:

- **BFU (Before First Unlock)** – it is the initial state when the device boots. This state refers to a phone that has been powered off or reset and has not been signed in again using the screen lock passcode. Most data is still securely encrypted;
- **AFU (After First Unlock)** - it is the state of the device after the first password entry. Although data is always stored in a securely encrypted form in the device's memory, most of the data is now available for decryption using cryptographic keys that were loaded into memory after the correct password or PIN code was entered.

Considering the above information, the optimal solution from the perspective of computer forensics is to secure the mobile device in the AFU state, which together with the PIN code or password will allow for the full filesystem extraction. As already mentioned above, it should allow for securing as much data as possible for further examination.

Type of data

In general, artifacts that can be exposed in the memory of a mobile device fall into the following categories:

- **SQLite databases** -it is a standalone, self-contained database that can store just about any kind of data, including binary BLOBs. SQLite databases are the primary source of storage for applications and system data.
- **property lists** - Plists are structured files that are used by iOS and applications to store, organize, and access data on the device. plists can be stored in two different

formats (an XML-based format or binary-encoded format, also known as Binary-PLIST (BPLIST)).

- *protocol buffers* - are a language-agnostic method developed by Google to serialize data structures to the binary format and are sometimes used as an alternative to XML files or plists.
- *XML file* – it is a type of a text file that stores data in a text-based format readable to both humans and the device.
- *log file* – it is a text file containing a chronological record containing information about events and activities related to users or operating system activity.

It is worthwhile to mention that the last two types of data are essentially text files and as such can be analysed using any text editor, the remaining files are typically stored in a binary format and require specific tools to analyse them.

Tools

Currently, there are several commercially available tools that allow for collecting data from the memory of mobile devices. They support a wide range of mobile devices, from smartphones to tablets, and thanks to regular updates allow for the extraction of data from subsequent versions of mobile operating systems that appear on the market. Examples of such solutions include Cellebrite UFED/Inseyets, MSAB XRY and Oxygen Forensic Detective. Unfortunately, the costs of purchasing and then maintaining this type of software are high whilst open-source alternatives (for example UFADE) do not cover all aspects of data acquisitions.

IOS

System

The starting point for any iOS study will be the `/private/var/mobile/Library` folder which contains artifacts related to device information as well as user account data. The subfolder “Preferences” contains device configuration files and user-defined settings, such as language, device name, time settings. In turn, the “SystemConfiguration” subfolder contains cellular information and network settings, including WiFi and VPN data. Artifacts relating to the SIM card installed in the evidence device such as the phone number, IMEI, and network carrier can be found at `/private/var/wireless/Library/Preferences`.

Another place where interesting information can be found about the operation of the mobile device itself is the `/device` and `/app` directories. In the first one, you find information such as the charge level of the device's battery (`/device/batteryPercentage`), whether the device was plugged in and charging or not (`/device/isPluggedIn`) or is passcode-locked or not (`/device/isLocked`). In the second one you find information which application is active and running in the foreground (`/app/inFocus`) and whether an internet connection (`/app/webUsage`) is used. With this information, the investigator will be able to identify which applications were used on the device during a given time, for how long, and to what extent they used data traffic.

The `applicationState.db` database located at `/private/var/mobile/Library/ FrontBoard/`, keeps track of Bundle IDs for any application that is installed on the evidence device. It also includes deleted applications, which may indicate the further direction of investigations in the the evidence analysis.

Another equally interesting database that can help an IT Forensic Examiner understand what processes were running on a device at a given time is the DataUsage.sqlite database, in a /private/var/wireless/Library/Databases directory. It stores network usage statistics for each process, which can help determine not only which processes were using the network, but also whether the subject was using the application at a given time.

If the examiner wants to know whether the evidence device has been switched to the airplane mode, they can check the contents of the com.apple.radios.plist file, w located in the "/private/var/preferences/SystemConfiguration/" directory. In the same folder, two other, equally interesting files can be found. The MAC address of the wireless network interface within the device can be found by analysing the NetworkInterfaces.plist file, while a list of known wireless networks is stored in the com.apple.wifi.plist file.

As regards the data on the location of the evidence device, in this case we must rely on the information contained in two databases located in the "/private/var/mobile/Library/Caches" folder, the first of them in the "/com.apple.routined" subfolder and is named "Cache.sqlite". It contains several useful records, such as GPS coordinates with timestamps and a list of Wi-Fi networks that the device has scanned. The second subfolder called "cache_encryptedB.db" from "/location" contains location data such as radio cells and Wi-Fi networks.

Communication and user activity

The primary source of information about both user and device activity is a database called KnowledgeC.db. It is in "/private/var/mobile/Library/CoreDuet/Knowledge/" directory on iOS devices. It contains many records about events related to the phone's operation and user activity but can only be accessed through a full filesystem acquisition.

Another interesting artifact is the SQLite database interactionC.db, which stores information about interactions between the device owner and their contacts, such as text messages or calls (most often from the last six months). Although interactionC.db does not contain the actual content of the conversation (that is located elsewhere, as will be explained shortly), it does contain metadata that can be correlated with other artifacts revealed in the evidence system. The database itself can be found in "/private/var/mobile/Library/CoreDuet/People".

Another group of artifacts that are of importance during the investigation are records related to communication via text messages and voice calls. All phone calls and FaceTime calls that have been placed, received, or missed are logged in the CallHistory.storedata database, located at "/private/var/mobile/Library/CallHistoryDB" subdirectory. It contains, inter alia, information such as: the unique identifier of the call, type, duration and the date/time when it was made, the direction of the call: whether it is an incoming or outgoing call and its participants.

The artifacts related to SMS/iMessage messaging are in the "/private/var/mobile/Library" folder. In the subfolder named "/SMS" two SQLite databases can be found that are important for IT Forensics. The "sms.db" database stores all metadata and contents of all messages (SMS and iMessage). This database stores most of the data, such as the date and time of sending or receiving messages, message contents, handler identifiers, statuses (received, sent, recalled, deleted) and additional information about the message, such as attached files. Messages received Before First Unlock (BFU) are stored in a temporary database, called "sms-temp.db".

The `"/private/var/mobile/Library/SMS/Attachments/"` folder, contains attachments that were sent through MMS or iMessage, such as photos, videos, and audio recordings whilst, the subfolder named `"/Drafts"` stores drafts messages or pending messages that the user has sent but that iOS failed to send due to the lack of connectivity.

There are two property lists related to message analysis, located in a directory called `"/private/var/mobile/Library/Preferences"`. The one called `"com.apple.madrid.plist"` is responsible for storing properties and user preferences related to SMS, iMessage, and FaceTime like all the aliases connected to the iMessage account, such as phone numbers and email addresses. The other, called `"com.apple.MobileSMS.plist"`, contains preferences related to SMS and iMessage indicating, for example, the retention period (set to Forever by default).

Web browsing history is an equally important issue from the perspective of IT Forensics. Typically, users of devices running the iOS operating system use the natively installed Apple Safari browser. Most of its artifacts are stored in the `"/private/var/mobile/Library/Safari"` subdirectory. The History.db database stores Safari browsing history which can be found on the following path `"/private/var/mobile/Library/Safari"`. It has two important tables which contain the browsing history related information of the Safari User: `"history_visits"` and `"history_items"`. The `"history_visits"` table stores information about each visit to a website, including the information like a unique identifier for each visit, the time at which the website was visited or the title of the web page visited. The `"history_items"` contains a unique identifier for each visit and the URL of the visited web page. Combining these two tables yields detailed information about the history of websites visited.

Safari Bookmarks and other artifacts such as Downloads, Recently Closed Tabs and Reading List are stored in plist files located in `"/private/var/mobile/Library/Safari"` folder. The Bookmarks.db database contains the titles and URLs of web pages that the user has bookmarked in Safari. It is worthwhile to mention that web pages that are added to the Reading List are also stored in this file. The BrowserState.db database contains a table that describes the current state of open tabs in Safari (title, URL, and `"last_viewed_time"` attribute) as well as session data, such as previously browsed pages within that tab - the private tabs will also be stored in this table (`"private_browsing"` flag is set).

In the operating system itself other artifacts can be found that are related to the Safari web browser and which may contain traces of user activity, such as session cookies and cache. They can be found in the folder `"/private/var/mobile/Containers/Data/Application/GUID/Library"`. The subfolder `"/Caches/com.apple.mobilesafari"` contains any data that was cached by the Safari web browser. The cache.db database contains traces of user navigation, even when the browsing history was deleted. Cached data is typically stored within the database as a BPLIST. The cookies can be found in the `"/Cookies"`. Safari cookies are stored in `"Cookies.binarycookies"` file, in the binary format. Because of their format, the use of computer forensics software or a dedicated script will be necessary.

When it comes to analysing additional artifacts related to the operation of the Safari web browser, of particular importance is the subfolder `"/Safari"` in the above-listed directory. It ought to be analysed because it contains several useful files, such as a list of downloaded files (Downloads.plist - containing, inter alia, the name of the downloaded file and its destination directory) or auto-fill data (`"Form Values"` plist file).

Apple Mail artifacts can be found in a folder located in `"/private/var/mobile/Library/Mail"` (a full filesystem extraction is required to access this resource). For each email account configured on the device under investigation, a corresponding subfolder is created in the Mail folder (the folder name refers to the unique account identifier). Each of these

contains additional subfolders, such as Inbox, Sent, Spam, and Drafts, which contain actual emails, stored as email files (EML). Additionally, attention should be given to two subfolders: “/Envelope Index” containing a SQLite database which stores metadata related to each email account that was configured on the device and “/Protected Index” which contains the actual data related to emails, such as a summary of the email's body, subjects, and addresses.

Android

System

In the Android operating system, its configuration data is stored both using XML, TXT files and in SQLite databases. The text file named “build.prop” located in the “/system” subdirectory contains key-value pairs that define system-wide properties, such as the device model, Android version, and network settings. The other files called “global_settings.xml” and “system_settings.xml” found in the “/data/system/users/0” subfolder store settings that affect the entire device, such as network preferences, display settings, and system behaviors. For example, Wi-Fi and mobile data configurations, display Settings like Brightness levels, screen timeout durations or system Preferences: default language, time zone, and other global preferences.

The “secure_settings.xml” file in Android is used to store secure system settings that are critical for the device's operation and security, including the Android ID or Bluetooth name. Typically, they are found in the directory “/data/system/users/0”. This file contains settings that are considered secure and are not easily modifiable by regular applications. Information is to be found there about the type of screen lock (PIN, pattern, password) and its configuration, details about whether the device storage is encrypted and the encryption status or certain developer settings that can impact the security of the device, such as USB debugging.

The “accounts_ce.db” and “accounts_de.db” databases are used by the Android system to manage user accounts and authentication tokens. They contain details about the accounts configured on the device, including Google accounts, email accounts, and other third-party service accounts. They can be found in the folder “/data/system_ce/0/”. One has to bear in mind, however, that there can be more than one local user. The first user is referred to as user 0, and when the “/system/users/” folder is examined, more local users can be revealed. These databases store account information such as the account name (e.g., email address), account type (e.g., Google, Microsoft), and user ID. We can also find there the tokens that applications use to authenticate with services without requiring the user to re-enter credentials as well as information about synchronization settings for each account, such as whether contacts, emails, or calendar events are synced. Third-party applications aren't required to store information in the accounts_ce.db or accounts_en.db. These files must therefore be considered as the starting point.

The “telephony.db” file is used by the Android system to manage and store information related to telephony services, including call logs, SMS, and MMS message. Usually it should be found in “/user_de/0/com.android.providers.telephony/databases/” subfolder. It contains information about the SIM card (or cards) used on the device, including ICCID and IMSI number or name of the network operator.

Two text files called “bt_config.conf” and “bt_config.bak” are stored in the userdata partition under the “/misc/bluedroid/” directory. They contain information about the Bluetooth configuration of the device and the history of connected Bluetooth devices

Communication and user activity

The `callog.db` database in Android contains detailed records of call history. This database is typically located at: `"/data/data/com.android.providers.contacts/databases"`. There we find the call Entries where each record includes information about outgoing, incoming, and missed calls. This database also contains details such as the phone number, call duration, call type (incoming, outgoing, missed), and timestamps for when the calls were made or received.

The `mmssms.db` database in the Android system contains SMS and MMS message data. It can be found in the following folder: `"/data/data/com.android.providers.telephony/databases"`. Each record contains information such as the message content, the time it was sent/received, and the phone numbers of the sender and recipient. In addition to the text, records may include multimedia attachments for MMS messages, such as images, videos, and sounds. The database also contains information about conversation threads that group messages by the contacts or groups, as well as details about individual parts of MMS messages.

The `download.db` database is of importance in the context of IT forensics as it contains records of files downloaded to the device. This database, typically located at subfolder: `"/data/data/com.android.providers.downloads/databases"`, stores information about files that have been downloaded (file name, file size, and download status). There we also find the details such as the URL from which the file was downloaded, the timestamp of the download, and the destination path on the device.

Google Chrome is a most popular web browser among Android users and it stores a variety of artifacts which provide valuable insights into the user's browsing behavior and activities. They are typically stored in SQLite databases within the Chrome app's data directory `"/data/data/com.android.chrome/app_chrome/Default"`. Bookmarks and Browser settings are stored in the JSON file with the same name. Chrome browsing history is stored in the 'History' SQLite database. It contains a table of URLs visited, along with timestamps, visit counts and details about each visit, including the time of visit and the referring URL. In the same database we find information about Searches which made by user (they are stored within the 'keyword_search_terms' table). Chrome Cookies are stored in the 'Cookies' SQLite database (located in the Network folder) which contains session data and other information used by websites.

The artifacts of Gmail email client provide the insights into the user's email activities and interaction. This data is mainly stored in a database called „bigTopDataDB.-<User_ID>” stored in the userdata partition under the `"/data/com.google.android.gm/databases"`. This database structure is such that it contains several tables, the key ones including individual email records, providing information about the email content and metadata; information about email threads, grouping related emails together and labels applied to emails, such as "Inbox," "Sent," "Spam," etc. It also contains detailed information about each email, such as the sender, recipient, subject, and body.

Forensic Tools

Selected examples of tools e used in the process of collecting digital evidence are discussed below:

- **commercial** - Cellebrite UFED⁶⁶ and MSAB XRY⁶⁷. Among commercially available tools, these are the two most popular solutions which enable the collection of data from mobile devices at an advanced level and then the analysis of thus obtained artifacts.. Their main drawback is high price of purchasing and

maintaining a license, as well as the lack of flexibility in terms of adapting to the individual needs of the user.

- **opensource** - UFADE (Universal Forensic Apple Device Extractor) and iLEAPP⁶⁸. The first one is basically a script written in Python, whose function it is to collect the data from Apple mobile devices. The other tool allows for the analysis of thus acquired data.. Another solution which is worthwhile to mention is the Avilla Forensic.. It is a free mobile forensic tool designed to assist investigators in collecting information and evidence from various mobile devices.

5.8. Memory artifacts analysis

Memory analysis procedure

Memory analysis is an integral aspect of the forensic investigator's job. Since everything within an operating system passes through the operating memory, it is repository of a great deal of useful information, including: :

- running and completed processes,
- open files,
- network activity, including network connections,
- user-generated content (e.g. passwords, temporarily stored content),
- system configuration, including software and hardware,
- memory mapping, etc.

Memory analysis tool

Volatility

A widely used framework in memory analysis is the **Volatility tool**⁶⁹, enabling us to extract forensic artifacts from an operational memory image and then to analyze them. It can be applied to handle memory images seized from Windows, Mac and Linux operating systems. Each operating system has a different memory structure and this tool has the ability to understand all of them. It is accomplished through the use of the operating system-specific profile that helps to understand the correct data structures of the memory image being examined and select the right tools for forensic analysis. .

The basic functionality of this tool involves the use of plugins. Each plugin performs specific tasks with an image of the operating memory. Within the Volatility tool we find ready-made plugins, but it is also possible to program your own plugin. Volatility is written in the popular Python language, which simplifies the plugin programming. .

Volatility 2 is an older version of this tool, written in Python 2, which leads to compatibility issues with newer operating systems. Volatility 3 is a redesigned version, written in Python 3, featuring a more modular architecture and providing better support for various memory formats. Volatility 3 offers faster performance, a more flexible plugin system, and improved support for new operating systems. However, some older plugins and features are missing in version 3, so in certain cases it is necessary to resort to Volatility 2.

Volatility supports the following images of RAM or other files that hold all or part of RAM.

- Raw/Padded Physical Memory

- Firewire (IEEE 1394)
- Expert Witness (EWF)
- 32- and 64-bit Windows Crash Dump
- 32- and 64-bit Windows Hibernation
- 32- and 64-bit MachO files
- Virtualbox Core Dumps
- VMware Saved State (.vmss) and Snapshot (.vmsn)
- HPAK Format (FastDump)
- LiME (Linux Memory Extractor)
- QEMU VM memory dumps

The volatility tool is used as follows:

```
volatility [-h] [-c CONFIG] [--parallelism [{processes,threads,off}]] [-e EXTEND] [-p
PLUGIN_DIRS] [-s SYMBOL_DIRS] [-v] [-l LOG] [-o OUTPUT_DIR] [-q] [-r RENDERER] [-f
FILE] [--write-config] [--clear-cache] [--single-location SINGLE_LOCATION] [--stackers
[STACKERS [STACKERS ...]]] [--single-swap-locations [SINGLE_SWAP_LOCATIONS
[SINGLE_SWAP_LOCATIONS ...]]] plugin
```

Bulk extractor

Bulk Extractor is a tool that extracts data such as email addresses, credit card numbers, phone numbers, as well as network packets from memory images. It is a most useful tool that provides the following features:

- finding email addresses, URLs, and credit card numbers that other tools may miss,
- the ability to process compressed data (e.g., ZIP, PDF, and GZIP files), and incomplete or partially corrupted data,
- creating lists of words found in the data, even those in compressed files that are in unallocated space,
- allowing data to be processed across multiple threads, and
- creates histograms showing the most used email addresses, URLs, domains, search terms, and other types of information.

This tool manages disk images, memory images, files or file directories. It extracts useful information without analyzing the file system or file structure. The input is divided into pages and processed by one or more scanners. The results are stored in separate files that can be easily inspected, analyzed or processed by other automated tools.

Operating system information

imageinfo. Information about the operating system can be retrieved from the memory image using the imageinfo plug-in.

Using Volatility 2: `vol.py -f <ImageFile> imageinfo`

Using Volatility 3: `vol.py -f "/path/to/file" windows.info`

```

C:\Users\XXX\volatility3-1.0.0>vol.py -f memdump.mem windows.info
Volatility 3 Framework 1.0.0
Progress: 100.00      PDB scanning finished
Variable      Value

Kernel Base    0x82636000
DTB            0x185000
Symbols file:///C:/Users/XXX/volatility3-1.0.0/volatility3/symbols/windows/ntkrpamp.pdb/92D32EE7188A4CB3AB23EDA0CB0F9D7B-2.json.xz
Is64Bit        False
IsPAE          True
primary        0 WindowsIntelPAE
memory_layer    1 FileLayer
KdDebuggerDataBlock 0x82762c78
NTBuildLab     7601.23915.x86fre.win7sp1_ldr.17
CSDVersion     1
KdVersionBlock 0x82762c50
Major/Minor    15.7601
MachineType    332
KeNumberProcessors 1
SystemTime     2020-09-28 11:53:33
NtSystemRoot   C:\Windows
NtProductType  NtProductWinNt
NtMajorVersion 6
NtMinorVersion 1
PE MajorOperatingSystemVersion 6
PE MinorOperatingSystemVersion 1
PE Machine     332
PE TimeDateStamp Wed Sep 13 14:47:57 2017

```

Figure 54: imageinfo

Process information

All processes are listed in the memory of the running Windows operating system in a single structure called `_EPROCESS`. Each entry in the `_PROCESS` structure contains one process with its metadata:

- process name,
- its executable path,
- the parent process,
- start time, and
- in some cases, the time of termination.

pslist. To display a list of processes in the operating system we use the `pslist` plugin which displays the process name, process ID, parent process ID, number of threads, number of handles, and the start and end timestamp of the process. This plugin does not detect hidden or unconnected processes (the `psscan` plugin can do that).

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> pslist`

Using Volatility 3: `vol.py -f "/path/to/file" windows.pslist`

```

C:\Users\XXX\volatility3-1.0.0>vol.py -f memdump.mem windows.pslist
Volatility 3 Framework 1.0.0
Progress: 100.00      PDB scanning finished

```

PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime	ExitTime	File output
4	0	System	0x83d33ba0 85	506	N/A	False	2020-09-28 20:50:28.000000	N/A	Disabled	
252	4	smss.exe	0x84427ac0 2	29	N/A	False	2020-09-28 20:50:28.000000	N/A	Disabled	
320	312	csrss.exe	0x84a385a8 8	445	0	False	2020-09-28 20:50:29.000000	N/A	Disabled	
356	312	wininit.exe	0x84a55030 3	78	0	False	2020-09-28 20:50:29.000000	N/A	Disabled	
364	348	csrss.exe	0x84a55518 8	275	1	False	2020-09-28 20:50:29.000000	N/A	Disabled	
392	348	winlogon.exe	0x84a63d28 4	112	1	False	2020-09-28 20:50:29.000000	N/A	Disabled	
452	356	services.exe	0x84a88d28 21	232	0	False	2020-09-28 20:50:29.000000	N/A	Disabled	
460	356	lsass.exe	0x84a919d8 7	614	0	False	2020-09-28 20:50:29.000000	N/A	Disabled	
468	356	lsass.exe	0x84a8db20 10	145	0	False	2020-09-28 20:50:29.000000	N/A	Disabled	
560	452	svchost.exe	0x84aa58b8 12	352	0	False	2020-09-28 20:50:30.000000	N/A	Disabled	
620	452	VBoxService.exe	0x84ace1b8 13	116	0	False	2020-09-28 20:50:30.000000	N/A	Disabled	
672	452	svchost.exe	0x84ad7a20 7	263	0	False	2020-09-28 11:50:32.000000	N/A	Disabled	
768	452	svchost.exe	0x84afd708 20	450	0	False	2020-09-28 11:50:32.000000	N/A	Disabled	
820	452	svchost.exe	0x84b0e4f8 18	367	0	False	2020-09-28 11:50:32.000000	N/A	Disabled	
848	452	svchost.exe	0x84b156f0 19	331	0	False	2020-09-28 11:50:32.000000	N/A	Disabled	

Figure 55: pslist.

psscan. When we need to display a list of processes by scanning the area markers (`_POOL_HEADER`), the `psscan` plugin should be used. This will also reveal processes that

were previously terminated (inactive) and those that were hidden or disconnected by a malicious code (rootkit).

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> psscan`

Using Volatility 3: `vol.py -f "/path/to/file" windows.psscan`

```
C:\Users\XXX\volatility3-1.0.0>vol.py -f memdump.mem windows.psscan
Volatility 3 Framework 1.0.0
Progress: 100.00
PDB scanning finished
```

PID	PPID	ImageFileName	Offset	Threads	Handles	SessionId	Wow64	CreateTime	ExitTime	File output
2632	452	SearchIndexer.exe	0x3ea2e4d0	16	639	0	False	2020-09-28 11:50:41.000000	N/A	Disabled
516	452	svchost.exe	0x3ec23988	6	93	0	False	2020-09-28 11:50:37.000000	N/A	Disabled
1916	1360	calc.exe	0x3ec24428	3	71	1	False	2020-09-28 11:52:33.000000	N/A	Disabled
1516	452	sppsv.exe	0x3ec25780	5	146	0	False	2020-09-28 11:50:36.000000	N/A	Disabled
3164	560	WmiPrvSE.exe	0x3ece15a8	9	176	0	False	2020-09-28 11:53:25.000000	N/A	Disabled
2320	2068	msedge.exe	0x3ece4d28	0	-	1	False	2020-09-28 11:50:39.000000	2020-09-28 11:52:11.000000	Disabled
1964	1844	cygrunsrv.exe	0x3ee30030	0	-	0	False	2020-09-28 11:50:35.000000	2020-09-28 11:50:35.000000	Disabled
1572	452	svchost.exe	0x3ee45bc0	11	144	0	False	2020-09-28 11:50:34.000000	N/A	Disabled
1624	1360	VBoxTray.exe	0x3ee64030	16	241	1	False	2020-09-28 11:50:34.000000	N/A	Disabled
1656	452	svchost.exe	0x3ee71358	15	230	0	False	2020-09-28 11:50:35.000000	N/A	Disabled
2068	1360	msedge.exe	0x3ee7a030	0	-	1	False	2020-09-28 11:50:37.000000	2020-09-28 11:52:11.000000	Disabled

Figure 56: psscan.

pstree. The pstree plugin lists processes in a tree view, identifying parent and child processes. It lists the process using the same method used by the pslist plugin whilst hidden or unbound processes will not be detected.

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> pstree`

Using Volatility 3: `vol.py -f "/path/to/file" windows.pslist`

```
C:\Users\XXX\volatility3-1.0.0>vol.py -f memdump.mem windows.pstree
Volatility 3 Framework 1.0.0
Progress: 100.00
PDB scanning finished
```

PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime	ExitTime
4	0	System	0x84b2f598	85	506	N/A	False	2020-09-28 20:50:28.000000	N/A
* 252	4	smss.exe	0x84b2f598	2	29	N/A	False	2020-09-28 20:50:28.000000	N/A
320	312	csrss.exe	0x84b2f598	8	445	0	False	2020-09-28 20:50:29.000000	N/A
* 2004	320	conhost.exe	0x84b2f598	2	33	0	False	2020-09-28 11:50:35.000000	N/A
356	312	wininit.exe	0x84b2f598	3	78	0	False	2020-09-28 20:50:29.000000	N/A
* 468	356	lsass.exe	0x84b2f598	10	145	0	False	2020-09-28 20:50:29.000000	N/A
* 460	356	services.exe	0x84b2f598	7	614	0	False	2020-09-28 20:50:29.000000	N/A
* 452	356	svchost.exe	0x84b2f598	21	232	0	False	2020-09-28 20:50:29.000000	N/A
** 768	452	svchost.exe	0x84b2f598	20	450	0	False	2020-09-28 11:50:32.000000	N/A
** 516	452	svchost.exe	0x84b2f598	6	93	0	False	2020-09-28 11:50:37.000000	N/A
** 1424	452	svchost.exe	0x84b2f598	19	313	0	False	2020-09-28 11:50:34.000000	N/A
** 3100	452	svchost.exe	0x84b2f598	17	377	0	False	2020-09-28 11:52:38.000000	N/A
** 672	452	svchost.exe	0x84b2f598	7	263	0	False	2020-09-28 11:50:32.000000	N/A
** 1572	452	svchost.exe	0x84b2f598	11	144	0	False	2020-09-28 11:50:34.000000	N/A
** 560	452	svchost.exe	0x84b2f598	12	352	0	False	2020-09-28 20:50:30.000000	N/A
*** 3524	560	dllhost.exe	0x84b2f598	6	88	1	False	2020-09-28 11:53:34.000000	N/A
*** 3164	560	WmiPrvSE.exe	0x84b2f598	9	176	0	False	2020-09-28 11:53:25.000000	N/A
** 820	452	svchost.exe	0x84b2f598	18	367	0	False	2020-09-28 11:50:32.000000	N/A

Figure 57: pstree.

procdump. Use the procdump command to dump the process executable. Optionally, passing the --unsafe or -u flags bypasses certain sanity checks that are used when parsing the PE header. Some malware will intentionally create size fields in the PE header to cause memory flushing tools to fail. Use --memory to insert spaces between portions of the PE that are not aligned on the page. Without --memory, you get a file that more closely resembles the file on disk until the parts are expanded.

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> procdump -D dump/ -p PID`

Using Volatility 3: `vol.py -f "/path/to/file" -o "/path/to/dir" windows.dumpfiles -- pid <PID>`

```
C:\Users\XXX\volatility3-1.0.0>vol.py -f memdump.mem -o output\procdump windows.dumpfiles --pid 1916
Volatility 3 Framework 1.0.0
Progress: 100.00 PDB scanning finished
Cache FileObject FileName Result
DataSectionObject 0x84c22758 calc.exe.mui file.0x84c22758.0x8449b868.DataSectionObject.calc.exe.mui.dat
DataSectionObject 0x84c22520 StaticCache.dat file.0x84c22520.0x84b4f188.DataSectionObject.StaticCache.dat.dat
SharedCacheMap 0x84c22520 StaticCache.dat file.0x84c22520.0x84b4f220.SharedCacheMap.StaticCache.dat.vacb
ImageSectionObject 0x84b0c8e8 GdiPlus.dll file.0x84b0c8e8.0x84b0eef0.ImageSectionObject.GdiPlus.dll.img
DataSectionObject 0x84e25f80 calc.exe.mui file.0x84e25f80.0x8449b868.DataSectionObject.calc.exe.mui.dat
ImageSectionObject 0x84e25b38 calc.exe file.0x84e25b38.0x84dfbdc0.ImageSectionObject.calc.exe.img
DataSectionObject 0x84c768e8 oleaccrc.dll file.0x84c768e8.0x84c868c8.DataSectionObject.oleaccrc.dll.dat
ImageSectionObject 0x84c85248 winmm.dll file.0x84c85248.0x84c87ba8.ImageSectionObject.winmm.dll.img
DataSectionObject 0x84b4db38 StaticCache.dat file.0x84b4db38.0x84b4f188.DataSectionObject.StaticCache.dat.dat
SharedCacheMap 0x84b4db38 StaticCache.dat file.0x84b4db38.0x84b4f220.SharedCacheMap.StaticCache.dat.vacb
ImageSectionObject 0x84b36778 WindowsCodecs.dll file.0x84b36778.0x84b37910.ImageSectionObject.WindowsCodecs.dll.img
ImageSectionObject 0x84c74e38 oleacc.dll file.0x84c74e38.0x84c74cf0.ImageSectionObject.oleacc.dll.img
ImageSectionObject 0x8449e038 msvcrt.dll file.0x8449e038.0x8449b008.ImageSectionObject.msvcrt.dll.img
ImageSectionObject 0x844a2b78 KernelBase.dll file.0x844a2b78.0x844ab440.ImageSectionObject.KernelBase.dll.img
```

Figure 58: procdump.

handles. To display the open "handle", use the handles plugin. Handles include files, registry keys, mutexes, named channels, events, workspaces, threads, and all other types of secured objects.

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> handles -p <optionalPID>`

Using Volatility 3: `vol.py -f "/path/to/file" windows.handles -- pid <PID>`

```
C:\Users\XXX\volatility3-1.0.0>vol.py -f memdump.mem windows.handles --pid 796
Volatility 3 Framework 1.0.0
Progress: 100.00 PDB scanning finished
PID Process Offset HandleValue Type GrantedAccess Name
796 FTK Imager.exe 0x8a1d3368 0x4 Directory 0x3 KnownDlls
796 FTK Imager.exe 0x83e4e4f8 0x8 File 0x100020 \Device\HarddiskVolume1\Users\IEUser\Desktop\Imager_Lite_3.1.1
796 FTK Imager.exe 0x84e6e420 0xc File 0x100020 \Device\HarddiskVolume1\Windows\winsxs\x86_microsoft.windows.common-controls
601.18837 none_ec86b8d6858ec0bc
796 FTK Imager.exe 0x9981a4f8 0x10 Key 0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\NLS\SORTING\VERSIONS
796 FTK Imager.exe 0x84b2d038 0x14 ALPC Port 0x1f0001
796 FTK Imager.exe 0x84f065a8 0x18 Mutant 0x1f0001
796 FTK Imager.exe 0x84e93458 0x1c Mutant 0x1f0001
796 FTK Imager.exe 0x961e9368 0x20 Key 0xf003f MACHINE
796 FTK Imager.exe 0x85026018 0x24 Event 0x1f0003
796 FTK Imager.exe 0x84c25158 0x28 Semaphore 0x100003
796 FTK Imager.exe 0x8503f158 0x2c Semaphore 0x100003
796 FTK Imager.exe 0x84c20170 0x30 Semaphore 0x100003
796 FTK Imager.exe 0x84c23a38 0x34 Semaphore 0x100003
796 FTK Imager.exe 0x84fb6ad0 0x38 Semaphore 0x100003
796 FTK Imager.exe 0x84456ba8 0x3c Semaphore 0x100003
796 FTK Imager.exe 0x99839380 0x40 Key 0x1 MACHINE\SYSTEM\CONTROLSET001\CONTROL\SESSION MANAGER
796 FTK Imager.exe 0x8502bd48 0x44 EtwRegistration 0x804
796 FTK Imager.exe 0x84ba8aa0 0x48 Event 0x21f0003
796 FTK Imager.exe 0x84a64530 0x4c WindowStation 0xf037f WinSta0
```

Figure 59: handles.

psxview. The psxview plugin compares the results of different process tracing techniques, such as pslist and psscan, which helps to detect hidden and unbound processes from a single output window.

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> psxview`

```
Volatility Foundation Volatility Framework 2.3.1
Offset(P) Name PID pslist psscan thrdproc pspcid csrss session deskthrd
-----
0x3de63340 SearchIndexer 2940 True True True True True True True
0x3df72a40 taskhost.exe 1184 True True True True True True True
0x3df12a08 VSSVC.exe 516 True True True True True True True
0x3dcd28e0 SearchProtocol 3088 True True True True True True True
```

Figure 60: psxview.

cmdline. A plugin for Volatility used to display the command arguments of a process. This plugin can be used to detect whether the process was started by a malicious command or not.

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> cmdline`

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> dlllist -p <OptionalPID> --offset=<PhysicalOffsetOfTheProcessFromPsscan>`

Using Volatility 3: `vol.py -f "/path/to/file" windows.dlllist -- pid <PID>`

```
C:\Users\XXX\volatility3-1.0.0>vol.py -f memdump.mem windows.dlllist --pid 796
Volatility 3 Framework 1.0.0
Progress: 100.00 PDB scanning finished
PID Process Base Size Name Path LoadTime File output
796 FTK Imager.exe 0x400000 0xa97000 FTK Imager.exe C:\Users\IEUser\Desktop\Imager_Lite_3.1.1\FTK Imager.exe N/A Disabled
796 FTK Imager.exe 0x7700000 0x142000 ntdll.dll C:\Windows\SYSTEM32\ntdll.dll N/A Disabled
796 FTK Imager.exe 0x76d00000 0xd5000 kernel32.dll C:\Windows\system32\kernel32.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x750e0000 0x4b000 KERNELBASE.dll C:\Windows\system32\KERNELBASE.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x6f600000 0x314000 LMS.dll C:\Users\IEUser\Desktop\Imager_Lite_3.1.1\LMS.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x76ae0000 0xa1000 ADVAPI32.dll C:\Windows\system32\ADVAPI32.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x76520000 0xac000 msvcrt.dll C:\Windows\system32\msvcrt.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x75310000 0x19000 sechost.dll C:\Windows\SYSTEM32\sechost.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x77210000 0xa2000 RPCRT4.dll C:\Windows\system32\RPCRT4.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x725e0000 0x69000 MSVCP100.dll C:\Users\IEUser\Desktop\Imager_Lite_3.1.1\MSVCP100.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x71f20000 0xbe000 MSVCR100.dll C:\Users\IEUser\Desktop\Imager_Lite_3.1.1\MSVCR100.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x72ea0000 0x10000 boost_date_time-vc100-mt-1_49.dll C:\Users\IEUser\Desktop\Imager_Lite_3.1.1\boost_date_time-vc100-mt-1_49.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x72e40000 0x13000 boost_thread-vc100-mt-1_49.dll C:\Users\IEUser\Desktop\Imager_Lite_3.1.1\boost_thread-vc100-mt-1_49.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x74520000 0x9000 VERSION.dll C:\Windows\system32\VERSION.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x733b0000 0x11000 NETAPI32.dll C:\Windows\system32\NETAPI32.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x73370000 0x9000 netutils.dll C:\Windows\system32\netutils.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x74b30000 0x19000 srvccli.dll C:\Windows\system32\srvccli.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x73340000 0xf000 wksccli.dll C:\Windows\system32\wksccli.dll 2020-09-28 11:52:26.000000 Disabled
796 FTK Imager.exe 0x6f580000 0x101000 LIBEAY32.dll C:\Users\IEUser\Desktop\Imager_Lite_3.1.1\LIBEAY32.dll 2020-09-28 11:52:26.000000 Disabled
```

Figure 63: `dlllist`.

Network connections analysis

Generally, network activity will leave traces in the operating memory. For example, examination of network connections can reveal a hidden connection created by a malicious code (malware). These connections can be hidden from the common list building tools in the same manner as processes.

netscan. The netscan plugin can be used to scan connections.

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> netscan`

Using Volatility 3: `vol.py -f "/path/to/file" windows.netscan`

```
C:\Users\XXX\volatility3-1.0.0>vol.py -f memdump.mem windows.netscan
Volatility 3 Framework 1.0.0
Progress: 100.00 PDB scanning finished
Offset Proto LocalAddr LocalPort ForeignAddr ForeignPort State PID Owner Created
0x159733a0 UDPv4 0.0.0.0 0 * 0 516 svchost.exe 2020-09-28 11:50:37.000000
0x16355bc0 TCPv4 0.0.0.0 22 0.0.0.0 0 LISTENING 2040 sshd.exe N/A
0x1aed6178 UDPv4 0.0.0.0 3702 * 0 1656 svchost.exe 2020-09-28 11:50:41.000000
0x1aed6178 UDPv6 :: 3702 * 0 1656 svchost.exe 2020-09-28 11:50:41.000000
0x27204bc0 TCPv4 0.0.0.0 22 0.0.0.0 0 LISTENING 2040 sshd.exe N/A
0x27dda3a0 UDPv4 0.0.0.0 0 * 0 516 svchost.exe 2020-09-28 11:50:37.000000
0x2b1dfbc0 TCPv4 0.0.0.0 22 0.0.0.0 0 LISTENING 2040 sshd.exe N/A
0x2fd40178 UDPv4 0.0.0.0 3702 * 0 1656 svchost.exe 2020-09-28 11:50:41.000000
0x2fd40178 UDPv6 :: 3702 * 0 1656 svchost.exe 2020-09-28 11:50:41.000000
0x328163a0 UDPv4 0.0.0.0 0 * 0 516 svchost.exe 2020-09-28 11:50:37.000000
0x3e867bc0 TCPv4 0.0.0.0 22 0.0.0.0 0 LISTENING 2040 sshd.exe N/A
```

Figure 64: `netscan`.

netstat. This plugin can only be used in Volatility version 2.

Using Volatility 2: `vol.py -f "/path/to/file" --profile <profile> netstat`

Registry analysis

The Windows operating system registries are discussed in more detail in earlier sections, highlighting the methods of extraction from running and shutdown systems. One has to bear in mind, however, that the current version of the Windows operating system

registry is contained in the operating memory. In the case of the memory seizure, that enables us to extract the entire registry or only specific keys from memory.

hivescan. To look up the physical addresses of the hive registers in RAM, use the hivescan command. This plugin is not particularly useful in itself. Actually, it is meant to be inherited by other plugins (such as the hivelist plugin) that generate and interpret the information found by this plugin.

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> hivescan`

Using Volatility 3: `vol.py -f "/path/to/file" windows.registry.hivescan`

```
C:\Users\XXX\volatility3-1.0.0>vol.py -f memdump.mem windows.registry.hivescan
Volatility 3 Framework 1.0.0
Progress: 100.00          PDB scanning finished
Offset
0x2807da5
0x5d21400
0xb21c008
0xd6c3948
0xf20b9c8
0xf3d69c8
0xf72a9c8
0x10353756
0x11604948
```

Figure 65: hivescan.

hivelist. We use the hivelist plugin to find the virtual addresses of the hive registry in RAM and the full paths to the corresponding hive on a disk. To print values from a particular subregister, first run this command to see the address of the subregisters.

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> hivelist`

Using Volatility 3: `vol.py -f "/path/to/file" windows.registry.hivelist`

```
C:\Users\XXX\volatility3-1.0.0>vol.py -f memdump.mem windows.registry.hivelist
Volatility 3 Framework 1.0.0
Progress: 100.00          PDB scanning finished
Offset FileFullPath      File output
0x8780c6b8          Disabled
0x8781c008          \REGISTRY\MACHINE\SYSTEM      Disabled
0x8783a218          \REGISTRY\MACHINE\HARDWARE    Disabled
0x8a1bf648          \Device\HarddiskVolume1\Boot\BCD      Disabled
0x8a1e0008          \SystemRoot\System32\Config\SOFTWARE      Disabled
0x8b0195c0          \SystemRoot\System32\Config\SAM      Disabled
0x8b059948          \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT      Disabled
0x8b0a09c8          \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT      Disabled
```

Figure 66: hivelist.

printkey. To view the subkeys, values, data, and data types contained in the specified registry key, use the printkey command. By default, printkey scans all hives and prints the key information (if found) for the desired key.

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> printkey -o <OptionalVirtualOffsetOfTheHiveFile>`

Using Volatility 3: `vol.py -f "/path/to/file" windows.registry.printkey`

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> mftparser --output=<body|text> --output-file=<OutputFile>`

evtlogs. The evtlogs command extracts and parses binary event logs from memory. The binary event logs are found on Windows XP and 2003 machines, so this plugin works on those architectures only. These files are extracted from the services.exe VAD process, parsed, and dumped to a specified location.

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> evtlogs`

userassist. To get the UserAssist keys from a sample, use the userassist plugin. For more information see Gleeda's Volatility UserAssist plugin post.

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> userassist`

shellbags. This plugin parses and prints Shellbag (pdf) information obtained from the registry. For more information see Shellbags in Memory, SetRegTime, and TrueCrypt Volumes. Two output options are available: verbose (default) and bodyfile format.

Using Volatility 2: `vol.py -f <ImageFile> --profile=<ImageProfile> shellbags`

shimcache. This plugin parses the Application Compatibility Shim Cache registry key.

Using Volatility 2: `vol.py -f win7.vmem --profile = <ImageProfile> shimcache`

Hibernation file

A hibernation file (hiberfil.sys) is a file that the Windows operating system uses by default to store the state of a device as part of the hibernation process. The operating system also keeps an open handle for this file so that no user, including the administrator, should read the file when the system is running, the file must be extracted from a disk dump or using dedicated tools.

imagecopy. Using the imagecopy plugin, the hiberfil.sys can be converted into a raw memory image.

Using Volatility 2: `vol.py imagecopy -f hiberfil.sys -O hiber.img --profile=<ImageProfile>`

Page file

The Windows operating system uses a paging file (pagefile.sys) to store page-sized blocks that do not currently fit in physical memory. This file, stored in %SystemDrive%\pagefile.sys, is a hidden system file and can never be read or accessed by the user, including the administrator. Unlike hypertext files, page files cannot be processed by the Volatility tool.

- Finding all paths in a file: `strings pagefile.sys | grep -i "[a-z]:\\\\" | sort | uniq | less`
- Finding environment variables: `strings pagefile.sys | grep -i "[a-zA-Z09_]*=.*" | sort -u | uniq | less`
- Finding URLs: `strings pagefile.sys | egrep "^https?://" | sort | uniq | less`
- Finding email addresses: `strings pagefile.sys | egrep '([[:alnum:]]_-[1,64]+@[[:alnum:]]_-[2,255]+?\\[[:alpha:]]{2,4})'`

References

- [1] <https://blog.onfvp.com/post/volatility-cheatsheet/>

- [2] <https://book.hacktricks.xyz/generic-methodologies-and-resources/basic-forensic-methodology/memory-dump-analysis/volatility-cheatsheet>
- [3] <https://volatilityfoundation.org/>
- [4] <https://andreafortuna.org/2019/05/15/how-to-read-windows-hibernation-file-hiberfil-sys-to-extract-forensic-data/>
- [5] <https://andreafortuna.org/2019/04/17/how-to-extract-forensic-artifacts-from-pagefile-sys/>

5.9. Network forensic artifact analysis

This part of the DFIR toolbox focuses on network traffic data recapping the monitoring approaches, standard analysis tools, and analytical scenarios represented as Jupyter Notebooks. The reader will not only get an overview of the available data but also clear procedures for handling these data to resolve common incident response cases.

Types of network traffic artifacts and their collection

The network traffic artifacts are mostly obtained from a network or directly on a selected endpoint. The choice of where to collect data always depends on the availability of the resource and the purpose for which the data is monitored. In the case of continuous network traffic monitoring, it is necessary to use network monitoring through selected network elements or probes. In the case of specific incidents, it may be helpful to collect network traffic directly from a given host. The following overview summarizes the detailed data description report by Jirsík et al.⁷⁰.

Network-based Data

Active and passive monitoring are two basic approaches to network monitoring. Active network monitoring actively injects probing traffic into the network and it infers the network status, based on the network response. Active approaches are used for network topology discovery, quality of service measurement, availability checking, or determination of connection properties. The type of the data is usually textual, and they are usually stored in unstructured files or, in some cases, in the form of structured data, such as JSON records.

Passive approaches rely on observation of network traffic as it passes through observation points within the network without any negative impacts on network functionality and traffic delivery. There are two generally accepted types of traffic monitoring access methods: in-line and mirroring methods⁷¹. Additionally, a third type of access method can be considered – the built-in method that emerged thanks to the improved capabilities of network switching and routing devices. All three types of access monitoring methods are described below:

- **In-line** traffic access methods observe network traffic using a capture device placed to a monitored line between two network elements. Capture devices are additional hardware installed to a line, such as bridging hosts or network traffic access points (TAP)⁷².
- **Mirroring** traffic access methods leverage a feature present in most of the packet forwarding devices – packets mirroring from one or more ports to another port. This feature is referred to as a port mirroring or Switched Port ANalyser (SPAN) session. Port mirroring introduces additional timing difference artifacts, changes traffic stream and reorders packets. Despite the mentioned issues, traffic mirroring access remains a widely used and popular method for network traffic monitoring due to its wide availability and low cost.

- **Built-in** traffic access methods represent network monitoring software implemented directly to network switching devices. For example, in a software-defined networking (SDN), devices provide (built-in) network monitoring software such as NetFlow export in Open vSwitch. This approach is usually only a complementary feature to the primary purpose of such devices, which is network switching. In the case of high traffic loads, a device allocates resources to fulfil its main function and the network monitoring resources are reduced.

The main approaches to network traffic monitoring can be categorized by the level of abstraction of information collected from network traffic in the following three categories: volumetric statistics, IP flow monitoring, and deep packet inspection (DPI). The level of abstraction is determined by details from a packet retrieved in a monitoring process and is connected to other characteristics of the network traffic monitoring approach, mainly throughput and resource requirements. The comparison of basic approaches to network traffic monitoring is depicted in Figure 69⁷³.

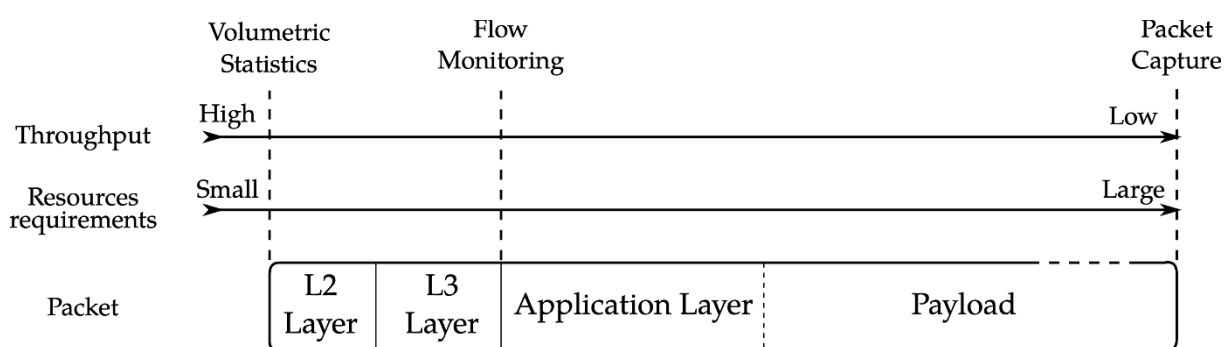


Figure 69: Levels of abstraction of the network monitoring data

- **Volumetric statistics** approaches do not require analyzing packets at all. Instead, these statistics are usually collected directly from network devices using a simple network management protocol SNMP (RFC3413) or remote network monitoring RMON systems (RFC3577). Thus, collected information provides only volumetric data on the network traffic including the number of transmitted or received packets and bytes or number of errors. These statistics provide insufficient information on the nature and content of network traffic and usually serve mainly for basic network accounting/management.
- **Deep packet inspection** analyses the entire packet information, i.e. both header and data parts of packets. The advantage of this approach is that the level of detail of information extracted from network traffic is the highest among the listed approaches. However, such high visibility comes at the cost of a lower throughput of the DPI approach and its high resource requirements. Hence, DPI is suitable rather for the analysis of targeted parts of network traffic than continuous monitoring of the whole network.
- **IP flow monitoring** is a balanced trade-off between the level of detail available and the scale of analysis. In general, IP flow monitoring analyses only information from packet headers, which, on one hand, allows for high throughput and relatively low resource requirements. On the other hand, the level of detail is reduced to basic metadata about network connections. The IP flow monitoring process for large-scale Internet measurement was originally described by Paxson et al.⁷⁴. An IP flow is an abstraction of a single-direction communication. All packets belonging to a particular IP flow have a set of common properties called flow keys (RFC7011). Apart from the flow keys, an IP flow contains several statistics about the connection and extracted application data.

Endpoint Data

Endpoint-based cybersecurity data mainly includes various types of logs and streaming information related to endpoint operations. Endpoint logs are recognized as important sources for gaining detailed visibility into the behavior of monitored resources – be it an operating system, a web server, a proprietary application, or a hardware appliance⁷⁵. According to Oliner and Stearley⁷⁶, logging is a pervasive practice in software development, which is likely the reason why log data represents an extremely rich source of valuable information about a vast range of elements of any modern IT infrastructure. From the standpoint of network traffic analysis, the main focus is on network logs. These logs represent events captured at the endpoint that relate to the endpoint's network activity. These logs provide information on the processes happening in a network infrastructure. Among the most common network logs are DNS logs, logs from DHCP, VPN access logs, HTTP, DLP, proxy logs, or routing information.

In addition to logs, it is possible to collect data directly from the endpoint at various levels of abstraction, as described above. The advantage over monitoring directly on the network is the specific focus on a given endpoint. Thus, in certain cases, network data can be linked to other artifacts from a given endpoint to gain greater insight into the analyzed behavior.

Network Data Processing

Data from active network monitoring is obtained by an ever-growing variety of tools. Among the by-the-time-proved tools are tools such as Ping, Traceroute, Nmap or ZMap. For a common network, active network monitoring is usually executed on-demand, and the results are stored for future use. The subsequent data processing steps include parsing the results, uploading the results to a network asset management tool such as Netbox⁷⁷ or SNIPE-IT⁷⁸.

Passive approaches to data processing involve the following steps: data capture, data storage, and data querying. For DPI, these data are captured from a network interface, usually using the libpcap library, and stored in PCAP files. Depending on the tool used, we are able to set different options for the storage of the PCAP files. For example, we can set the rotation of the files based on the size of the file or the duration of the capture. There are several tools for PCAP file merging (mergecap), editing (editcap and the Scapy library for Python), and querying or analysis (Wireshark, tshark). Capture and analysis of PCAP files are usually done on-demand, mainly because of the required computational resources.

A general workflow for IP flow monitoring has been proposed in RFC 5470, involving the process steps depicted in **Błąd! Nie można odnaleźć źródła odwołania.**⁷⁹. Packets are observed at one or more observation points. A device called exporter handles the creation and export of IP flow records. A collector receives the IP flow records from one or more exporters and stores them for further analysis. Exporters and collectors are often dedicated devices aimed to achieve a high processing performance. Nevertheless, these two devices can be integrated into a single device to save hardware costs. The scheme provided here covers the majority of IP flow monitoring installations.

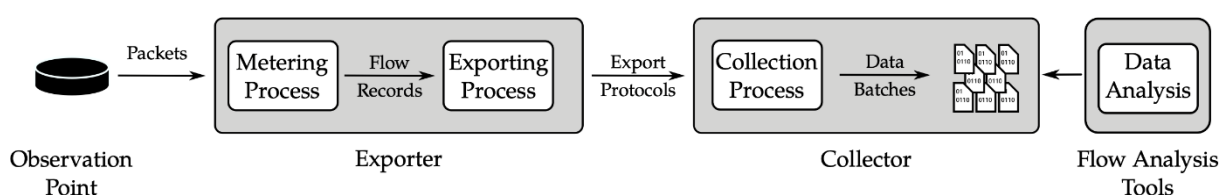


Figure 70: IP Flow data processing workflow

The analysis of data from passive monitoring approaches will typically serve the following purposes:

- **Network analysis and reporting fulfil** the original purpose of network traffic monitoring. Reports created, for example, from IP flow records, can provide information on the network. Such report can contain a sum of flows and data volume transferred by/to a customer or a number of connected devices.
- **Network performance monitoring:** The goal of performance monitoring is to watch the status of the services in a network. Frequently used information for performance assessment is, for example, round-trip time (RTT), delay, response time, and packet loss. Network-based performance monitoring enables central, one-point performance observations without accessing the devices directly.
- **Attack and anomaly detection:** Network traffic data can be naturally used for network forensics and incident response. For example, observing communication pairs in combination with IP reputation lists or blacklists allows for detecting infected devices in a network. Communication with an IP address that is listed on a blacklist means that the communicating host is likely being infected.

A fundamental, comprehensive, and detailed overview of network anomaly detection methods is provided by Bhuyan et al.⁸⁰. A basic categorization of these techniques is shown in **Błąd! Nie można odnaleźć źródła odwołania.** 71 with taxonomy proposed by Umer et al.⁸¹. They divide anomaly and intrusion detection techniques into three general categories: statistical, machine learning, and others. Statistical techniques are based on functions utilizing features of monitored network traffic to create an application, host, or network profile. A deviation of newly observed traffic and created profile is the underlying principle of the detection method. Machine learning techniques aim to automate data analysis and overcome the limits of statistical and other approaches, that require ample knowledge about anomalies and intrusions that should be detected. Other categories include the approaches, such as simple monitoring of thresholds or specific traffic patterns, that do not fit in the previously-listed categories and utilize prior knowledge about anomalies and intrusions.

Common analytical tools for network traffic captures

This section introduces three common tools for analyzing network traffic captures: Wireshark, Arkime, and Zui. These tools are chosen to offer the analysts different views of the data, as each represents unique functionalities that help to monitor network activity and identify suspicious patterns. Wireshark is renowned for its user-friendly interface and comprehensive protocol support, providing detailed insights into individual packets. Arkime focuses on extensive data retention and enables quick searches across historical records, whilst Zui enhances analysis with modern visualization techniques, making it easier to interpret complex network data. Together, these tools equip forensic analysts with the capabilities to understand and evaluate network events. However, these tools cannot handle the cases of network traffic analysis of larger networks, typically using IP flow data. For better clarity, this issue is addressed in the next section.

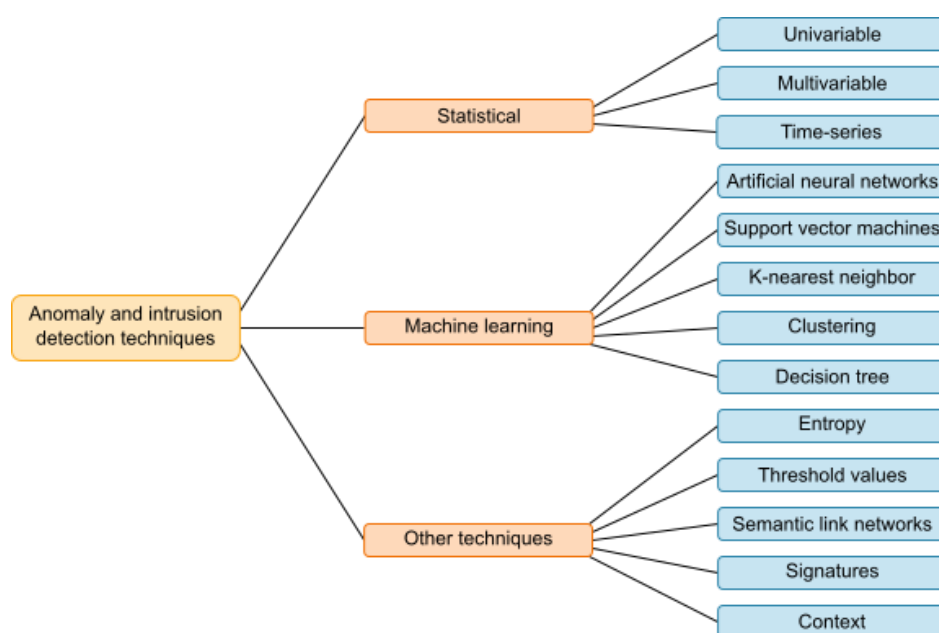


Figure 71: Taxonomy of anomaly and intrusion detection techniques

Wireshark

Wireshark⁸² is a widely used open-source network protocol analyzer that allows for the capture and in-depth examination of network traffic at the packet level. It provides the capability to intercept and log real-time data by traversing a network, offering detailed visibility into communication between devices. Wireshark also has a command line variant called tshark⁸³, which provides similar functionality. Professionals use Wireshark for network troubleshooting, performance diagnostics, and security analysis, as it enables the identification of vulnerabilities, misconfigurations, or network inefficiencies. By dissecting protocols across multiple layers of the OSI model, Wireshark supports the

analysis of both common and proprietary protocols, making it an indispensable tool for network engineers, cybersecurity specialists, and IT professionals seeking to optimize or secure network infrastructure.

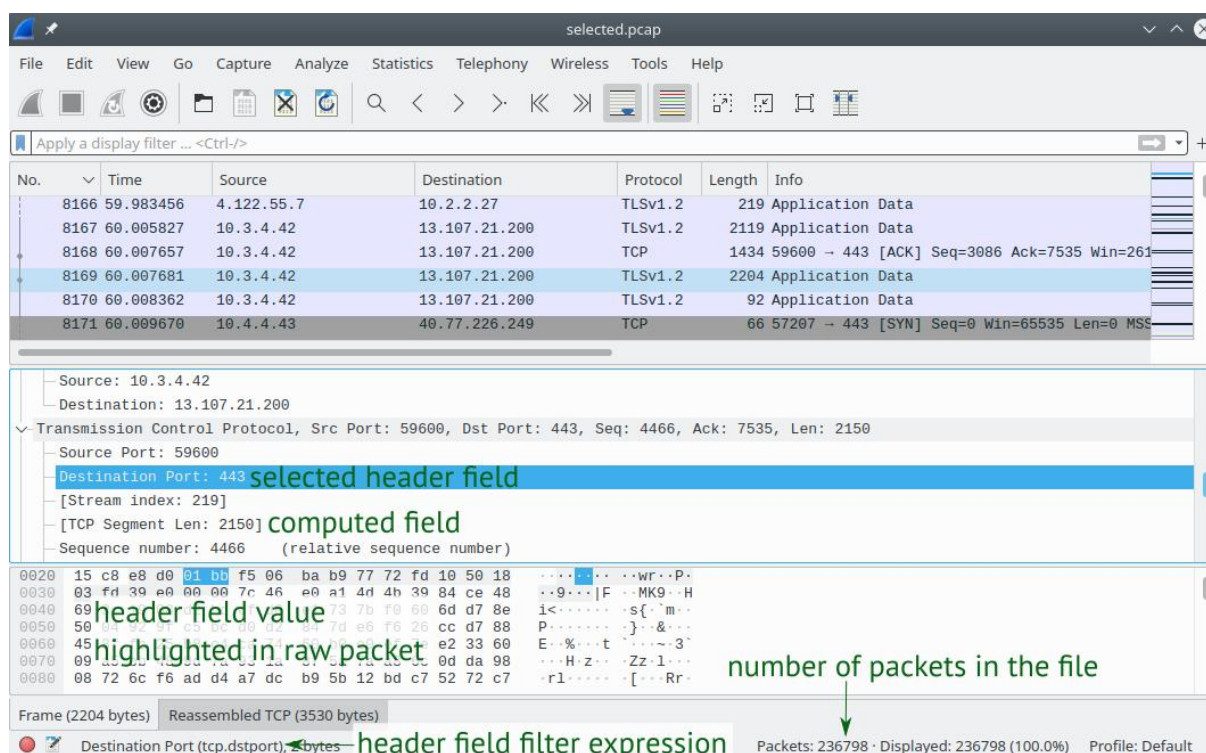


Figure 72: Wireshark main window

The main window of the application is shown in Figure 72. It consists of four main parts: menu with filter, packets list, packet dissection, and packet hexadecimal view. From the standpoint of forensic analysis, the most interesting part of the menu is Statistics information. The main sections of the menu are highlighted in Figure 73, and the information contained is summarized below:

- Capture File Properties (capture time, snaplen, packets, bytes, ...)
- Protocol Hierarchy (tree of observed protocols)
- Conversations (list of connections on L2-L4 with statistics)
- Endpoints (list of L2-L4 endpoints with traffic volume)
- Packet Lengths (packet length statistics)
- I/O Graph (throughput in time, custom selections and filters)

- Flow Graph (Communication diagram of selected traffic)
- HTTP (analysis of requests, server statistics)

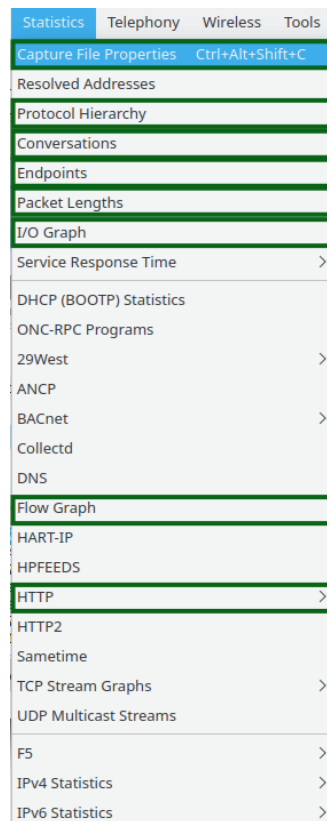


Figure 73: Wireshark Statistics menu

Wireshark provides a comprehensive functionality to filter displayed flows. The main part is the Filter bar that allows the analyst to write the filter directly using Wireshark display filter syntax⁸⁴ or by the right mouse click on the packet field, as shown in Figure 74. The filters used can be saved to ease further analysis. They can also be used to highlight selected packets instead of filtering them from the packet list. Besides the filtering options, Wireshark also provides a full-text search functionality, as shown in Figure 75. It allows the user to search in dissected packet details or packet payload, which the user may use as input string, regular expressions, or hex values.

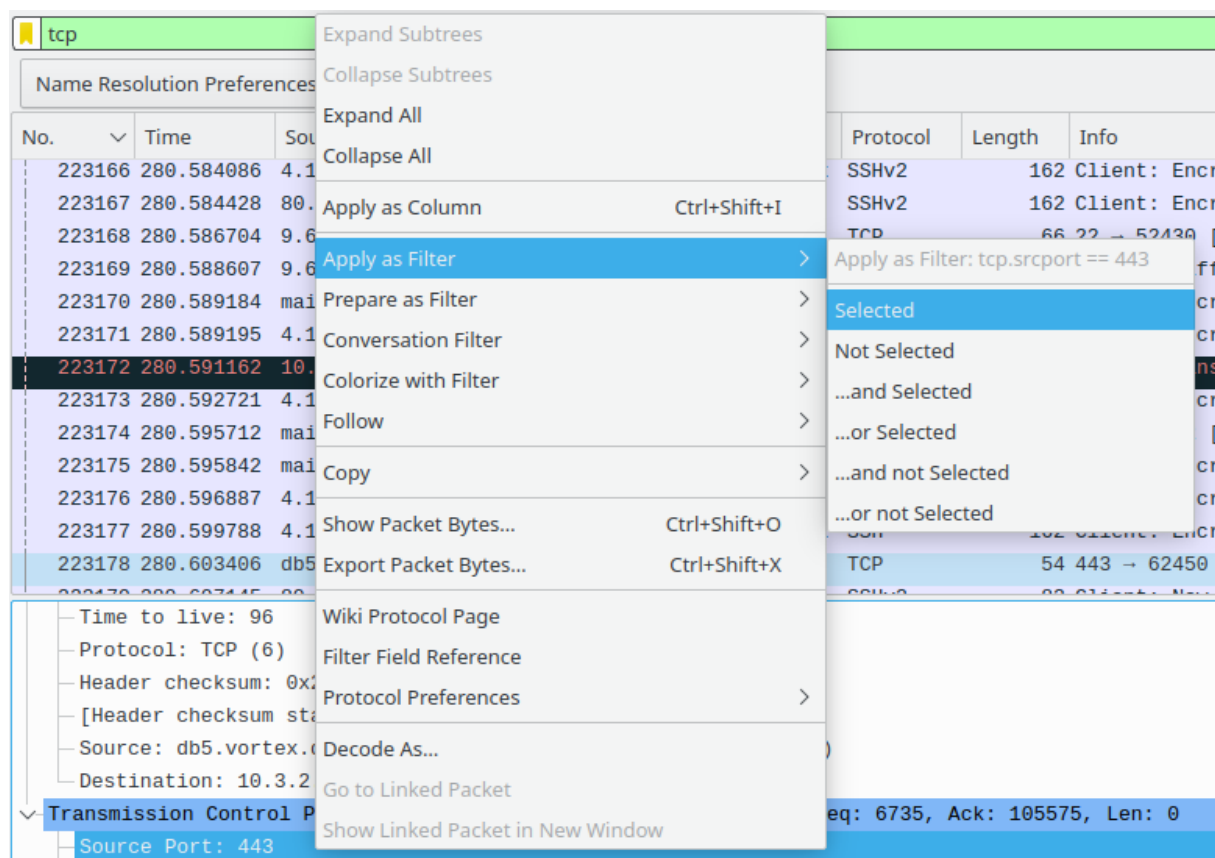


Figure 74: Wireshark interactive filter selection

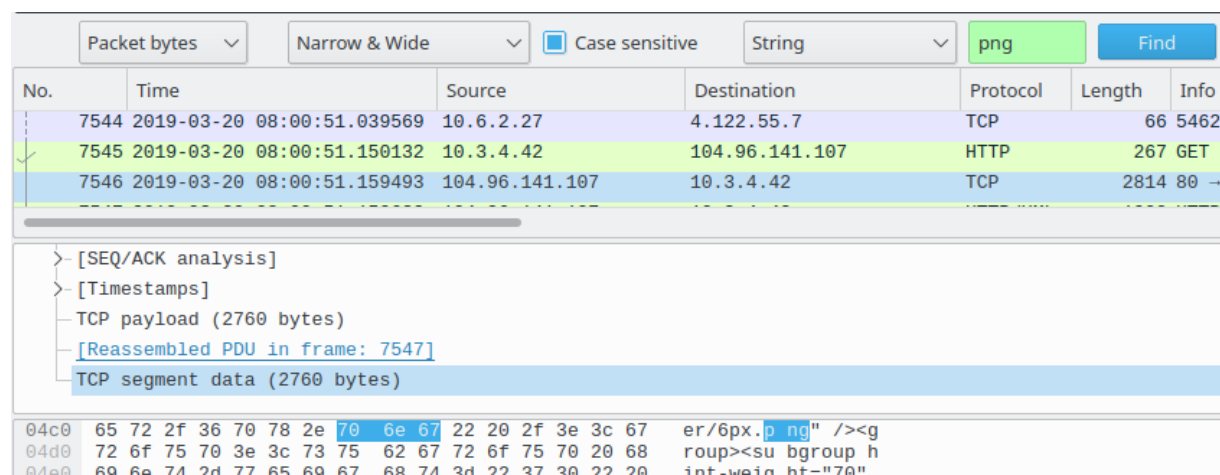


Figure 75: Wireshark full text search

The Packet list is the Wireshark window's main part, which displays an ordered list of all observed or filtered network packets. Each packet has an assigned number and contains a timestamp with information on when the packet was observed. By default, the timestamp is depicted as a number of seconds from the beginning of the packet capture. Although this can be useful in some cases, using the UTC time display format is better for easing the correlation of observed events with other data (this can be changed in Menu -> View -> Time display format). For better clarity, it is also recommended to use Name resolution and enable the option Resolve Network Addresses (as depicted in Figure 76) which shows hostnames instead of IP addresses in the packet list. By default, this resolution is done using data from the packet capture, but in the settings (Edit

Resolved Name option), it is also possible to define domain name servers and use active names resolution.

Another important functionality offered by Wireshark is the Follow TCP Stream (or Follow UDP Stream) option, which appears when you right-click on a packet. It filters out single

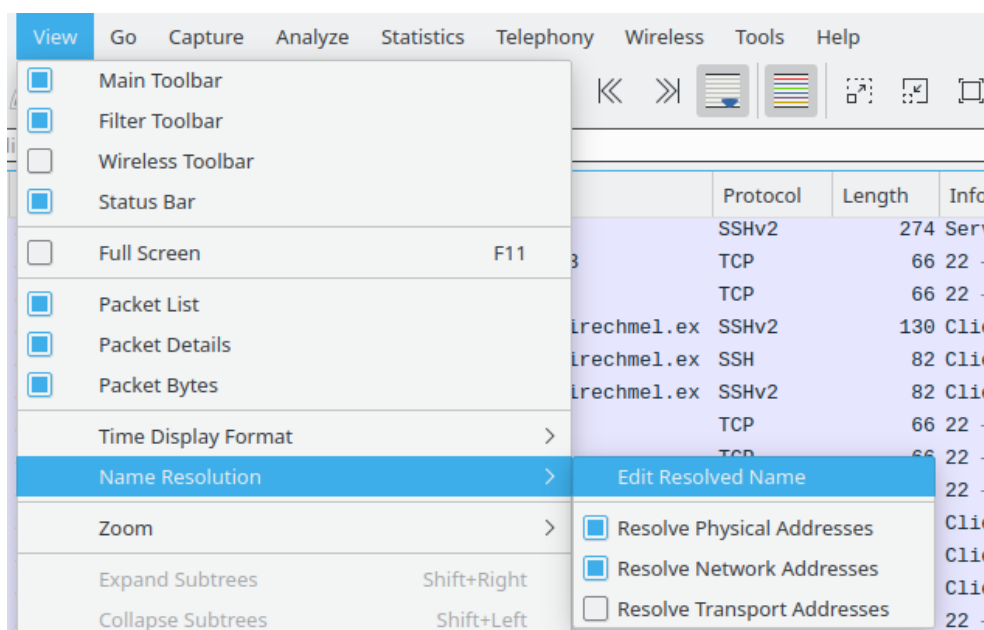


Figure 76: Wireshark name resolution options

TCP connections and displays payloads that can also be exported (see Figure 77). It is a very useful functionality, especially in unencrypted connections, to see transferred data.

Overall, Wireshark offers a great deal of functionality, beyond the scope of this guide. We therefore recommend using the official website or other professional publications to better understand all the functionalities provided.

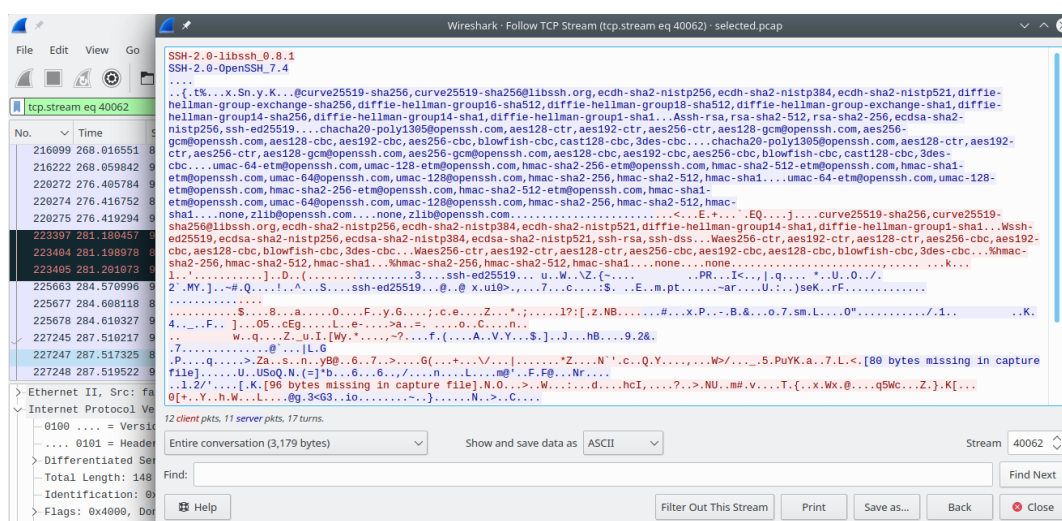


Figure 77: Wireshark Follow TCP Stream functionality

Arkime

Arkime⁸⁵ is a large scale, open source, indexed packet capture and search tool. It enables the capture and indexing of network data, focusing on efficiently managing and analyzing large volumes of packet data. Arkime offers a robust user interface for visualization and searching, facilitating identification and analysis of suspicious activities within the network. It supports various network protocols and provides advanced features, such as real-time querying and the ability to integrate with other cybersecurity tools. Arkime's key features include large-scale analysis of packet trace data, extraction and advanced search of information, export as CSV or PCAP, and integration of the CyberChef tool⁸⁶. By virtue of its flexible architecture and scalability, Arkime is a valuable asset for security analysts and network administrators who are required to conduct thorough network traffic analysis.

Arkime's graphical interface is a web application comprising six main pages: Sessions, SPIView (and SPIGraph), Connections, Hunt, and Cont3xt. The Files page displays the files that have been uploaded⁸⁷, as Arkime supports both the import of network traffic captures and the monitoring of network interfaces. The Sessions page (see Figure 78) presents a list of indexed sessions for the selected time window and search expressions. It features a timeline graph and displays extracted packet data for each observed connection (see Figure 79). The SPIView page (see Figure 80) focuses on the extracted packet data, showing their counts and enabling searches within network connections. This data can be further analyzed on the SPIGraph page, which includes statistical information and visualizations. The Connections page (see Figure 80) visualizes filtered connections through a network graph. Full-text search capabilities, including searches of packet content and payload data, are available on the Hunt page. Finally, the Cont3xt pages automates gathering contextual intelligence data from popular Cyber Threat Intelligence (CTI) services.

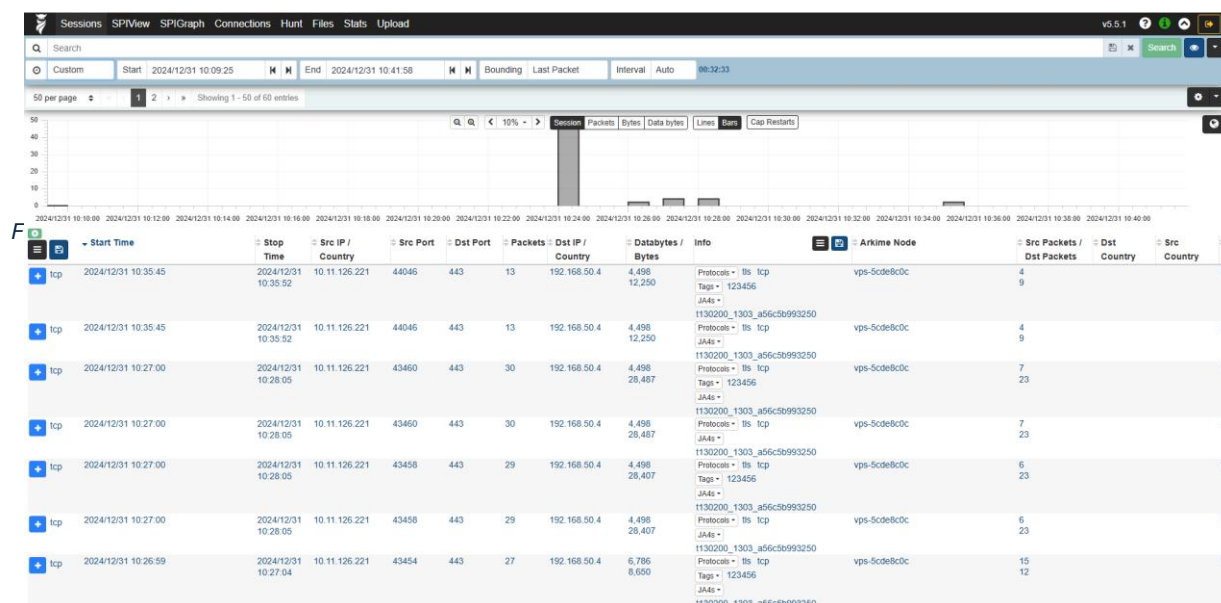


Figure 78: Arkime Sessions page

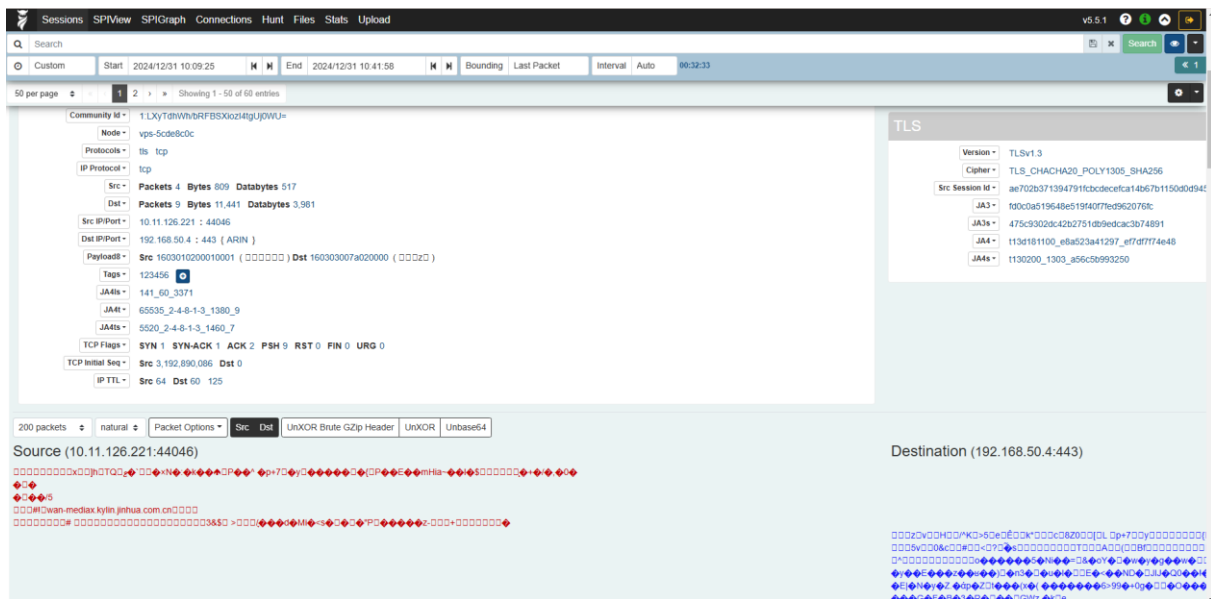


Figure 79: Arkime connection details

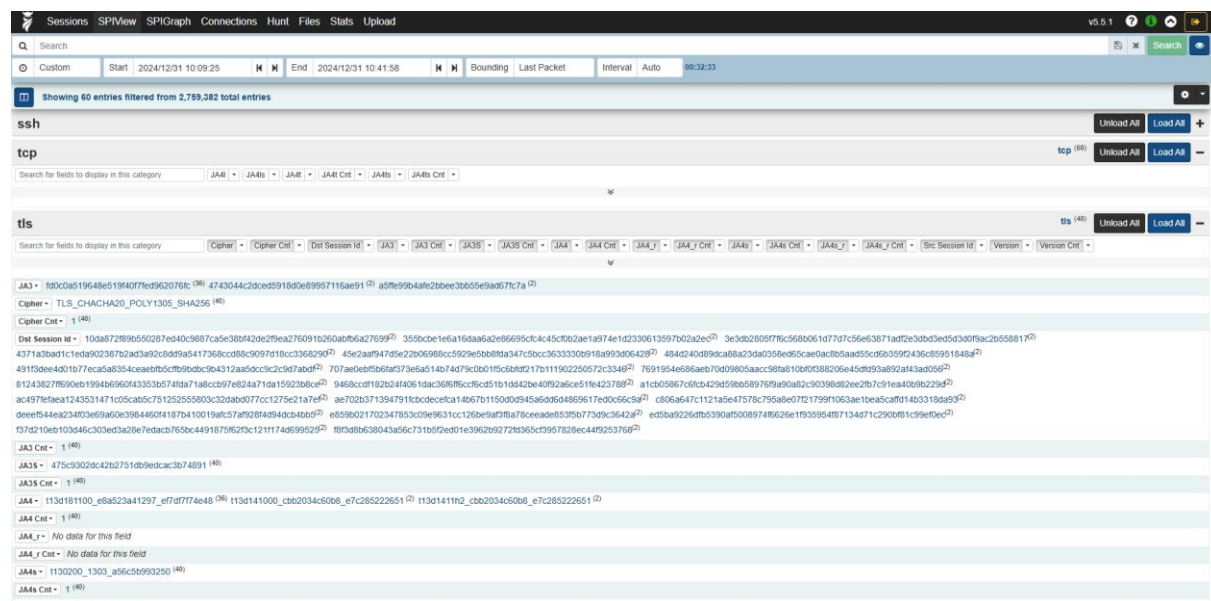


Figure 80: Arkime SPIView page

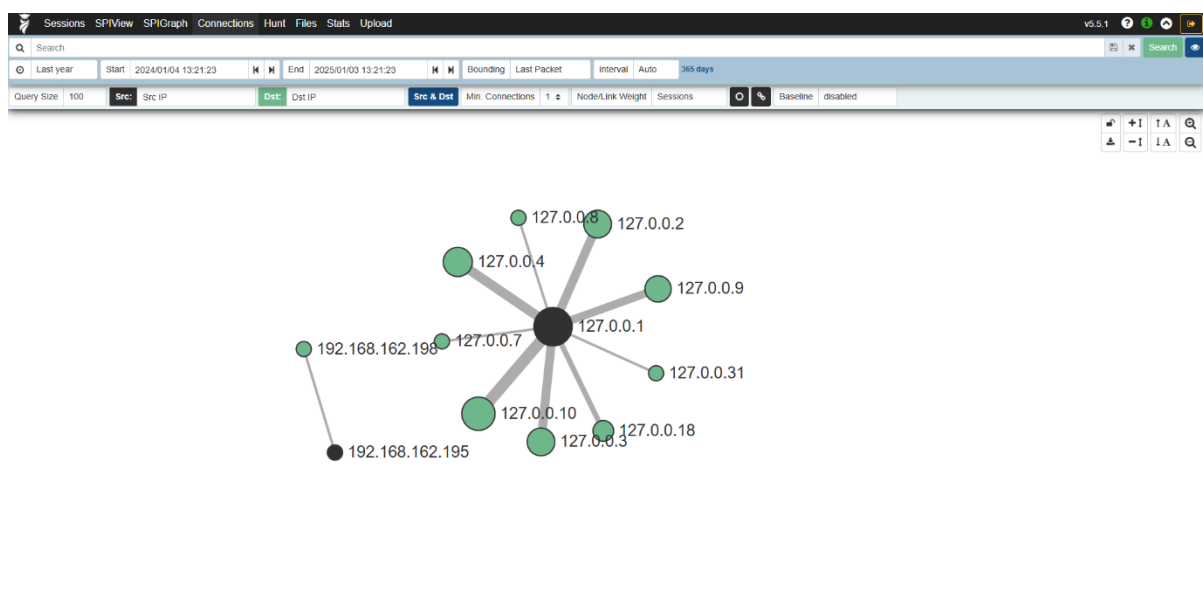


Figure 81: Arkime Connections page

Zui

Zui⁸⁸, formerly Brim, is an open-source desktop tool dedicated to analyzing and visualizing network traffic and events. Unlike Wireshark and Arkime, Zui does not allow for monitoring the network interface and is thus only used to analyze network traffic captures. The core of the tool is Zeek logs⁸⁹ that are extracted from the packet capture and stored in the Zed database (proprietary open-source database)⁹⁰. The extracted data can be analyzed forthwith, but it is recommended to wait until all data processing is complete. After the data is extracted and loaded, the Suricata IDS is used to detect anomalies and attacks that are further linked to the original data. The data can be queried using the Zed language⁹¹ whereas the query may include multiple data sources and splitting operations where multiple paths run in parallel and paths can be combined, merged by one or more sort keys, or joined using relational style join logic. Examples of such queries are listed below:

- Activity Overview: `count() by _path | sort -r`
- Unique DSN Queries: `_path=="dns" | count() by query | sort -r`
- Windows Networking Activity: `_path matches smb* OR _path=="dce_rpc"`
- HTTP Requests: `_path=="http" | cut id.orig_h, id.resp_h, id.resp_p, method, host, uri | uniq -c`
- Unique Network Connections: `_path=="conn" | cut id.orig_h, id.resp_p, id.resp_h | sort | uniq`
- Connection Received Data: `_path=="conn" | put total_bytes := orig_bytes + resp_bytes | sort -r total_bytes | cut uid, id, orig_bytes, resp_bytes, total_bytes`
- File Activity: `filename!=null | cut _path, tx_hosts, rx_hosts, conn_uids, mime_type, filename, md5, sha1`
- Show IP Subnets: `_path=="conn" | put classnet := network_of(id.resp_h) | cut classnet | count() by classnet | sort -r`
- Suricata Alerts by Category: `event_type=="alert" | count() by alert.severity, alert.category | sort count`
- Suricata Alerts by Source and Destination: `event_type=="alert" | alerts := union(alert.category) by src_ip, dest_ip`

- Suricata Alerts by Subnet: `event_type=="alert" | alerts := union(alert.category) by network_of(dest_ip)`

Zui⁴, formerly Brim, is an open-source desktop tool dedicated to analyzing and visualizing network traffic and events. Unlike Wireshark and Arkime, Zui does not allow for monitoring the network interface and is thus only used to analyze network traffic captures. The core of the tool is Zeek logs⁵ that are extracted from the packet capture and stored in the Zed database (proprietary open-source database)⁶. The extracted data can be analyzed forthwith, but it is recommended to wait until all data processing is complete. After the data is extracted and loaded, the Suricata IDS is used to detect anomalies and attacks that are further linked to the original data. The data can be queried using the Zed language⁷ whereas the query may include multiple data sources and splitting operations where multiple paths run in parallel and paths can be combined, merged by one or more sort keys, or joined using relational style join logic. Examples of such queries are listed below:

- Activity Overview: `count() by _path | sort -r`
- Unique DSN Queries: `_path=="dns" | count() by query | sort -r`
- Windows Networking Activity: `_path matches smb* OR _path=="dce_rpc"`
- HTTP Requests: `_path=="http" | cut id.orig_h, id.resp_h, id.resp_p, method, host, uri | uniq -c`
- Unique Network Connections: `_path=="conn" | cut id.orig_h, id.resp_p, id.resp_h | sort | uniq`
- Connection Received Data: `_path=="conn" | put total_bytes := orig_bytes + resp_bytes | sort -r total_bytes | cut uid, id, orig_bytes, resp_bytes, total_bytes`
- File Activity: `filename!=null | cut _path, tx_hosts, rx_hosts, conn_uids, mime_type, filename, md5, sha1`
- Show IP Subnets: `_path=="conn" | put classnet := network_of(id.resp_h) | cut classnet | count() by classnet | sort -r`
- Suricata Alerts by Category: `event_type=="alert" | count() by alert.severity,alert.category | sort count`
- Suricata Alerts by Source and Destination: `event_type=="alert" | alerts := union(alert.category) by src_ip, dest_ip`
- Suricata Alerts by Subnet: `event_type=="alert" | alerts := union(alert.category) by network_of(dest_ip)`

Zui stores data in Pools, functioning as a data lake, ensuring the data remains accessible even after the tool is closed. Users can use the main window's IMPORT FILES field to load traffic captures or drag and drop the selected PCAP files. Upon successful import, an initial overview is presented (see Figure 82), which includes filters, statistics of observed log types over time, and a list of Zeek log entries organized by connection time. By clicking on a selected row, users can view detailed information about the related connection in a new modal window (see Figure 83). This modal provides additional functionalities for specific log types, such as checking exported data using VirusTotal or exporting the connection as a PCAP file for analysis in Wireshark. When more complex queries and filters are applied, the data is displayed in a table format, with columns corresponding to the query parameters. The query language facilitates common statistical operations on the data, providing a comprehensive overview of the analyzed information.

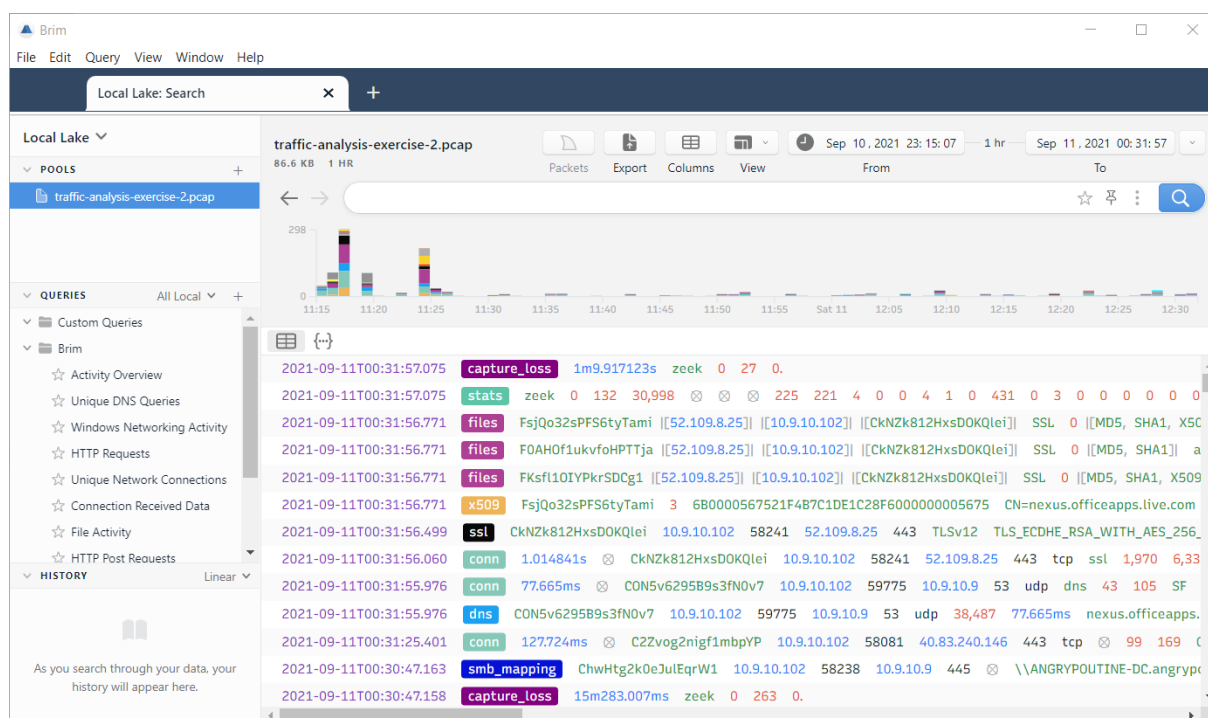


Figure 82: Zui initial overview

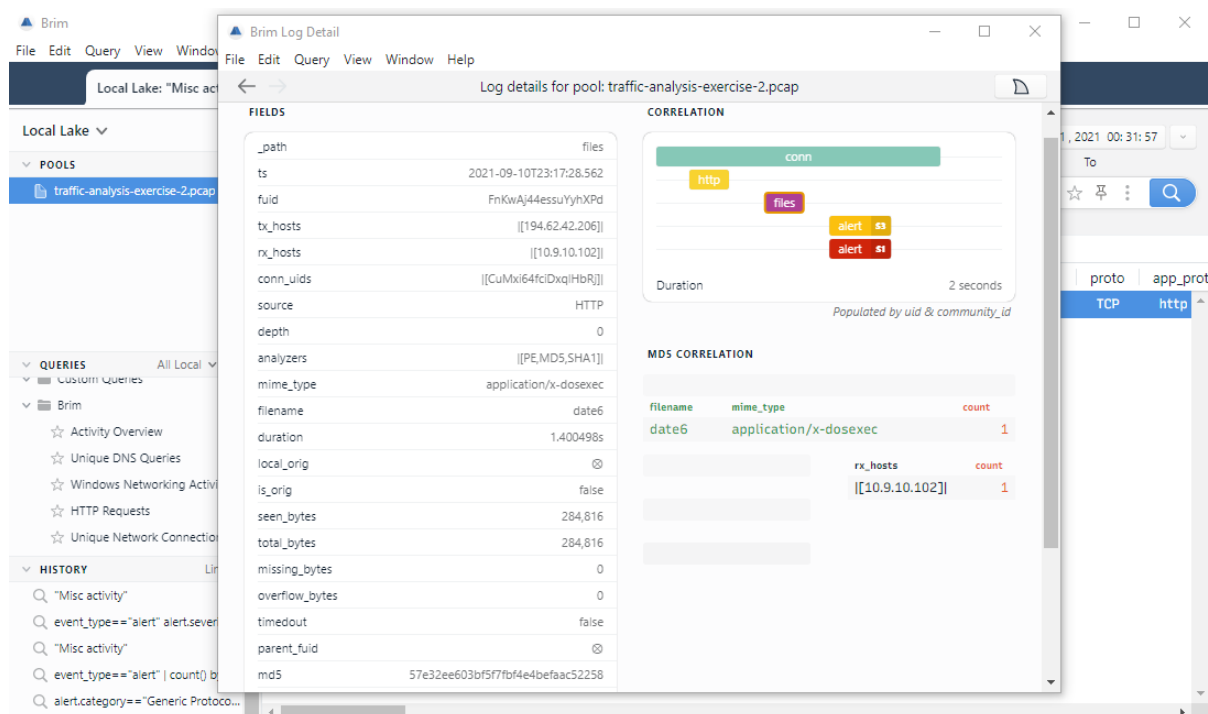


Figure 83: Zui connection detail

Analytical scenarios (Jupyter notebook)

Jupyter notebooks can be adopted as a practical tool for structured analysis and reporting in certain Digital Forensics and Incident Response (DFIR) processes. Their versatility makes them particularly valuable for addressing common incidents and use cases.

This section introduces nine Jupyter notebooks developed to handle frequent scenarios critical for incident response and analysis, offering efficient tools to streamline workflows. Five notebooks primarily rely on IP flows as their main data source but also incorporate additional sources such as system logs, accounting records, and Cyber Threat Intelligence (CTI), similar to the remaining four notebooks, enabling more comprehensive incident analysis.

These nine notebooks have proven to be highly effective for our incident response team thus far. This section outlines their capabilities and demonstrates how they can be applied effectively in various incident response scenarios.

The Jupyter Project

The Jupyter Project⁹², an open-source platform, has become a valuable tool across many fields, including Digital Forensics and Incident Response (DFIR). Developed initially from the IPython project, Jupyter allows users to create and share interactive documents that combine live code, data visualizations, and explanatory texts. Supporting multiple programming languages, Jupyter is highly versatile, though Python is most used in DFIR contexts.

Jupyter's key advantage lies in its ability to conduct analysis by combining code execution, data processing, and documentation within a single, easily shared Jupyter notebook. For DFIR teams, this means investigators can perform live data analysis—such as parsing logs, analyzing network flows, or correlating threat intelligence—while documenting findings and insights in real-time. This enhances both transparency and reproducibility, as the entire workflow is captured and easily shared with others.

In our practice, in collaboration with incident response (IR), Jupyter notebooks have proven to be highly effective tools for resolving incidents. The process should be designed so that analysts create, maintain, and refine these notebooks based on feedback from IR teams to ensure their continued relevance and effectiveness. While analysts may use these tools as needed, the notebooks they develop are usually more specific, designed for particular incidents. These notebooks are also often kept flexible rather than moved into full production, allowing adaptability to various situations.

Shared Notebook Properties

Based on our experience, we have identified the following post-processing mechanisms as the essential features of Jupyter notebooks designed for working with network flows. The presenting notebooks include and share these elements that extend their functionality, typically through buttons or options menus. The following is a list of these core features:

- **IP Flow Types:** Allows switching between two types of IP flows: uniflows and biflows. *Uniflows* return data in its original form as stored, while *biflows* merge the flows programmatically.
- **IP Filtering:** The *IP flows filtering* tab allows filters to be applied to the obtained IP flows. The user can specify the column, choose an operation, and input the desired value, e.g., *destination port == 443*. Also, multiple values can be entered

and separated by commas. Filters are evaluated as “AND” between lines and “OR” within a line.

- **Geolocation:** Post-processing of flows entering country codes and visualized as country flags are used in graphs to display geolocation data. When downloading data, only country codes are included.
- **Hostname Resolution:** The *Hostname resolution* tab allows IP addresses to be translated into hostnames. Two methods are available: *Passive resolution*, based on data from HTTP, DNS, or TLS within the obtained flows, and *Active resolution*, which uses reverse DNS queries. Active resolution should be used for a limited number of flows. If both methods are selected, passive resolution is applied first, with active resolution filling in the remaining addresses. A colored dot next to each hostname indicates the resolution method as seen in Figure 84.
- **IP Flow Export and Storage:** Data can be exported in the CSV or JSON format from the initial query or the most recent filtered result.
- **Aggregation Functions:** These functions offer a statistical overview of the obtained IP flows, providing sums of relevant statistics. Aggregation is based on protocol, source and destination IP addresses, and ports, offering insights into the overall network activity.

The screenshot shows the 'Hostname resolution' tab in a web interface. It includes a section to 'Select hostname resolution approach:' with two options: 'Passive resolution' (checked) and 'Active resolution' (unchecked). Below this is a 'Resolve NOW' button and a 'Resolve automatically' checkbox. At the bottom, there is a table of resolved IP flows.

Below the resolution controls, there is a 'Show 25 entries' dropdown and a 'Search:' input field. The table below displays the results:

SOURCE IP ADDRESS	SOURCE PORT	DESTINATION IP ADDRESS	DESTINATION PORT
147.251.163.136	44382	www.topky.sk	443
195.113.158.133	58614	195.113.144.194	53

Figure 84: Passive and active hostname resolution detail

kNotebooks with IP Flows as the Primary Data Source

The following notebooks demonstrate analytical processes essential for daily incident response and network monitoring, offering a structured approach to handling routine security tasks. Their main strength lies in the depth of information IP flows provide, while they are enhanced with additional tools and data sources.

Analysis of Connections Between Hosts

One of the most critical capabilities for any incident response team is the ability to search, filter, and analyze the database of network flows collected across the organization, extracting only the relevant flows based on key parameters. This analytical playbook focuses on analyzing network connections for selected hosts, identified by their IP addresses, within a specific time window.

For querying, it is important to provide the flexibility of identifying both destination and source IPs, whether as individual addresses, lists, or relevant CIDR blocks. Additionally, users should be able to choose and combine different network flow data channels, as seen in this notebook (Figure 85, option “Channel”), depending on the situation—such as probes located on the network perimeter or within internal segments.

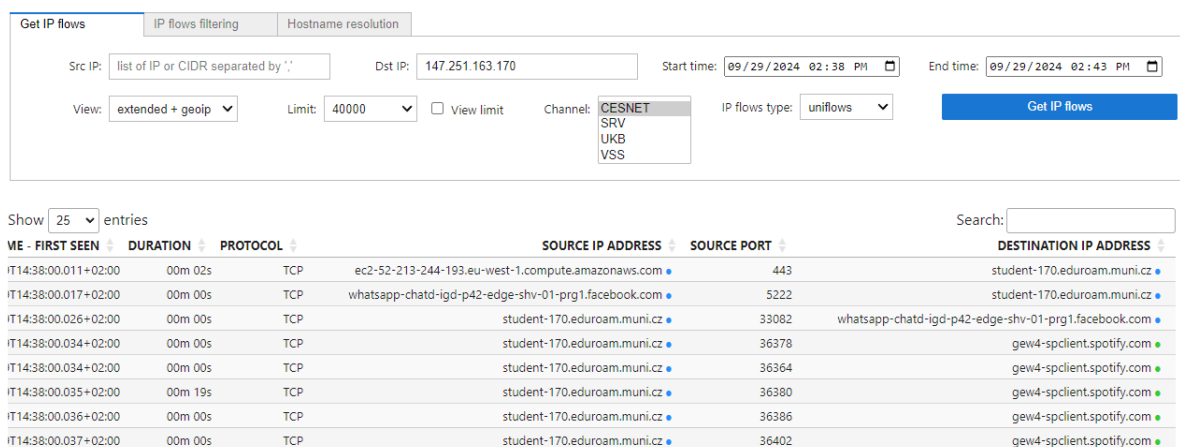


Figure 85: The network connections of the defined IP address within the time window

A common challenge with many tools is the large amount of information returned in query results, which can overwhelm rather than assist the analyst. This is particularly true when working with the IPFIX standard, a universal protocol for exporting network flow information from routers, probes, and other devices, which covers a broad range of protocols.

Based on our experience, we have streamlined the output to focus on the most relevant data for incident response, organized into multiple views:

- The **Basic view** provides key network flow attributes such as start time, duration, protocol, source and destination ports and addresses, TCP flags, and the number of transferred packets and bytes.
- The **Extended view** builds on the basic one by adding protocol-specific details for HTTP, TLS, and DNS, such as HTTP host, TLS Server Name Indication (SNI), and DNS Query Name (QNAME). For clarity, Figure 85 shows only a portion of the output table.

The Jupyter notebook implementation allows additional views to be added as needed, though in practice these have proven to be sufficient.

Finally, the **View limit** checkbox controls how much data is displayed. When enabled, all IP flow data is retrieved, but only the first N results based on the limit value are shown. If unchecked, the limit is applied in the query itself, restricting the data retrieved.

Analysis of Connections with Domains

This playbook is an adaptation of the previous one, designed for analyzing network connections by specifying domain names or lists of domains instead of destination IP addresses, within a given time window. It enables the analysis of communications not just to a specific server but to any service hosted on it. This is particularly useful for investigating access to malicious domains, such as during a large-scale phishing attack on an organization.

For domain specification, the wildcard (%) can be used to represent zero, one, or multiple characters, allowing flexible domain matching. As with the previous playbook, the output is streamlined and controlled through different views (see Figure 86).

Get IP flows

IP flows filtering

Hostname resolution

Domains: cloudfare.com, google.com, facebook.com, apple.com, microsoft.com

Start time: 17. 09. 2024 12:00

End time: 17. 09. 2024 14:00

View: extended + geopl

Limit: 40000

☐ View limit

Channel: CESNET

Get IP flows

SRV

UKB

VSS

Show 25 entries

SOURCE IP ADDRESS	SOURCE PORT	DESTINATION IP ADDRESS	DESTINATION PORT	TCP FLAGS	PACKETS	BYTES	DETECTED PROTOCOL	HTTP HOST	HTTP URL
147.251.106.162	51770	20.70.246.20	80	.AP.S.	4	236	HTTP	microsoft.com	
147.251.106.162	51793	20.70.246.20	80	.AP.S.	4	236	HTTP	microsoft.com	
147.251.106.162	51803	20.70.246.20	80	.AP.S.	4	236	HTTP	microsoft.com	
147.251.106.162	51820	20.70.246.20	80	.AP.S.	4	236	HTTP	microsoft.com	
147.251.106.162	51831	20.70.246.20	80	.AP.S.	4	236	HTTP	microsoft.com	
147.251.106.162	51831	20.70.246.20	80	.AP..	2	120	HTTP	microsoft.com	
147.251.106.162	51772	142.251.36.110	80	.AP.S.	4	233	HTTP	google.com	

Figure 86: Output for connections to query the specified domains in each time window

Domain Activity Analysis for Selected Hosts

Domains are one of the most frequent that CTIs analysts encounter in most types of forensic data fragments while dealing with incidents. Therefore, evaluation of user and machine interactions with various domains becomes one of the most common and important tasks.

To quickly assess the behavior of a specific host (user or machine), especially when malicious activity is suspected, analysts need to collect two types of information from multiple, sometimes external, systems: who the host (in this case, their IP address) is communicating with and whether these communications are legitimate or suspicious.

This analytical playbook offers an approach to gather domain data linked to specified IP addresses and evaluate those domains using Cyber Threat Intelligence (CTI) tools. The process is divided into two main stages: data gathering and domain evaluation.

Data Gathering:

All (sub)domains in DNS, TLS, or HTTP(S) associated with the specified IP address(es) are collected through network flows within the user-specified time frame.

Domain Evaluation:

- **Step 1 - Benign Domain Check:** In the first step, the collected domains are compared against popular Internet (sub)domain lists like SimilarWeb⁹³ and Tranco⁹⁴ to identify domains that are most likely benign, see Figure 87)
- **Step 2 - Malicious Domain Analysis:** For domains not marked as benign in the previous step, the process assesses their reputation using free tools such as Cloudflare, DNS0_EU, Google Safe Browsing, and PhishingArmy to evaluate potential maliciousness.
- **Step 3 - In-Depth Malicious Domain Analysis:** Any domain flagged as potentially malicious is subject to further investigation with VirusTotal for a deeper threat assessment. VirusTotal⁹⁵ is used at this stage due to its query limits, making it a more resource-efficient choice after initial checks.

Gathers all (sub)domains observed in communication (DNS, TLS and HTTPS) of the selected IP address and evaluates their presence in top (sub)domain lists:

- SimilarWeb list of top 10K domains;
- Tranco list of top 1M domains;
- Tranco complete list of subdomains.

IP: Start time: End time:

DOMAIN	SUBDOMAIN	DOMAIN IN TOP 10K LIST	DOMAIN IN TOP 1M LIST	SUBDOMAIN IN TOP LIST
qq.com	qq.com	true	true	true
n64.top	c1.147-251-16-195.gpuav.ipv4.n64.top	false	true	false
oracle.com	oracle.com	true	true	true
163.com	www.163.com	false	true	true
cyberresilience.io	cyberresilience.io	false	true	true
higi.com	higi.com	false	true	true
shadowserver.org	dnsscan.shadowserver.org	false	true	true
youporn.com	www.youporn.com	true	true	true
ta6.ch	ta6.ch	false	true	true
google.com	www.google.com	true	true	true
collectd.org	collectd.org	false	true	true
parrotdns.com	ip.parrotdns.com	false	true	true

Figure 87: Evaluation of the benignity of domains interacting with the specified IP address

Analysis of Connections to Services

For humans, the easiest way to gain insight into a large amount of data is through visualization, even at the cost of reducing fidelity. Therefore, we are gradually incorporating it into our notebooks, as demonstrated in this example.

This notebook serves as a playground for analyzing and visualizing host interactions with specific services, defined by a list of IP addresses within the configuration file. Its primary objective is to identify anomalies in data transfer volumes during communication with various services, which may suggest potential account compromise or other irregularities, depending on the context and the type of applications involved.

Its functionality is divided into two main parts:

- **Flow Filtering and Statistics:** The notebook takes as input a file containing filtered network traffic, where the analyzed source IP address(es) communicates with the services of interest defined in the configuration file. It then calculates various statistics about the traffic, such as the amount of uploaded and downloaded data and the percentage of total traffic represented by these interactions. This is the first step in finding something suspicious (see Figure 88).

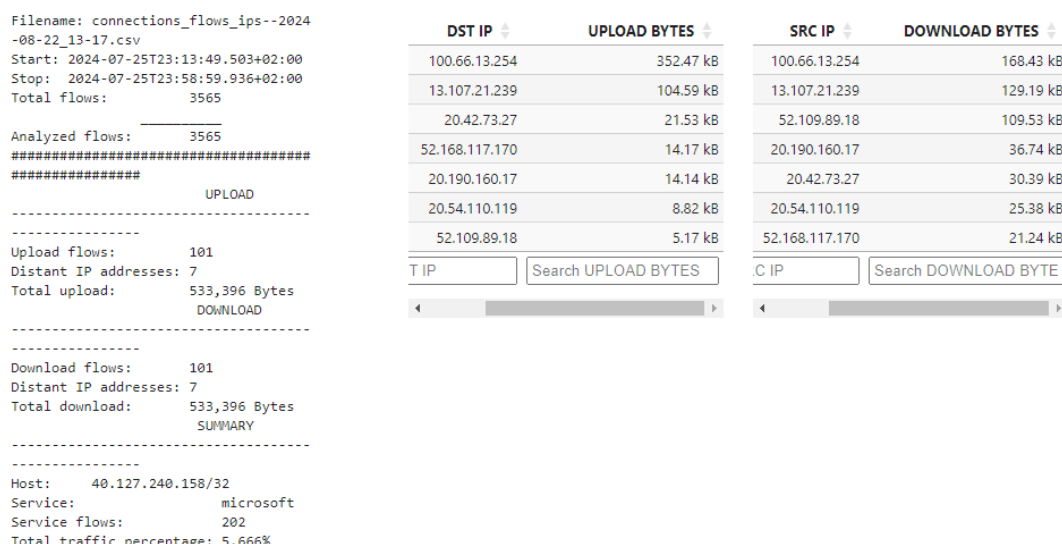


Figure 88: The example of statistical output

- **Visualization:** The filtered data is visualized to provide a clear understanding of the communication between the host and services, making it easier to identify patterns or anomalies in network traffic (see Figure 89).

For simplicity, the notebook operates exclusively with uniflows, as they are adequate for the intended analysis. The network flows required as the input file (in CSV or JSON format) can be obtained from a previously referenced notebook in this section.

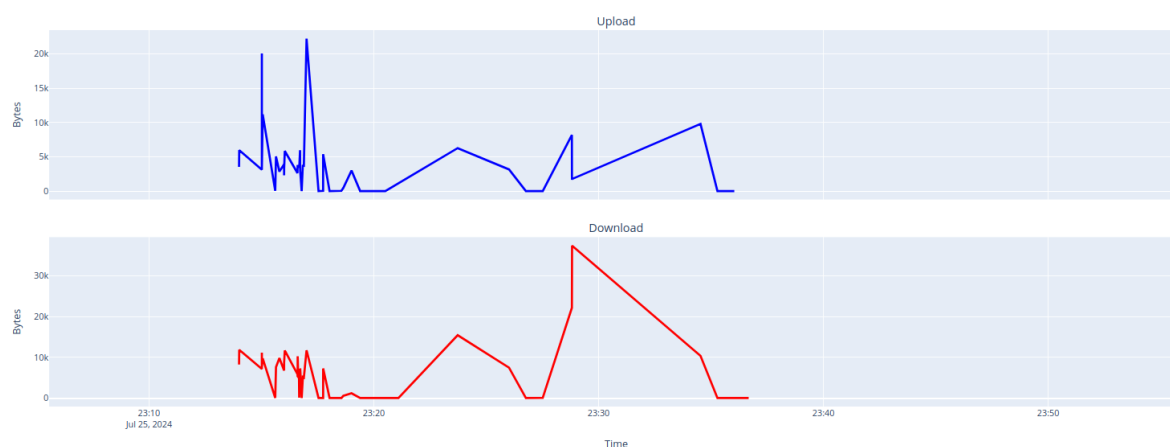


Figure 89: Visualization of data uploads and downloads to service-specific servers

Botnet Communication Detection

Jupyter notebooks provide an excellent platform for adapting existing algorithms or detection methods into tools that can be tested by incident response (IR) or analytics teams using an organization's own, often highly specific data before full deployment into production.

This notebook is based on the AC&CD⁹⁶ algorithm, which detects malicious beaconing activity by analyzing time delta and data size distributions between the same source and destination. What makes the algorithm unique is its focus on **biflows (bi-directional network traffic)** and its method of adjusting the data score based on the volume of received bytes. Specifically, if the maximum received bytes are below 20,000, the algorithm reduces the score, indicating a lack of substantial data transfer. This reflects

real-world attack patterns, where malicious commands or tools typically involve larger amounts of data, helping distinguish legitimate traffic from potential beaconing activity.

It identifies significant interactive phases (command execution) by analyzing interactivity duration, connection count, and detection score. The default parameter values (duration: 3600s, count: 24, score: 0.85) are set to avoid false positives, as illustrated in the following figure.

Botnet Communication Detection

Jupyter notebooks provide an excellent platform for adapting existing algorithms or detection methods into tools that can be tested by incident response (IR) or analytics teams using an organization's own, often highly specific data before full deployment into production.

This notebook is based on the AC&CD⁸ algorithm, which detects malicious beaconing activity by analyzing time delta and data size distributions between the same source and destination. What makes the algorithm unique is its focus on **biflows (bi-directional network traffic)** and its method of adjusting the data score based on the volume of received bytes. Specifically, if the maximum received bytes are below 20,000, the algorithm reduces the score, indicating a lack of substantial data transfer. This reflects real-world attack patterns, where malicious commands or tools typically involve larger amounts of data, helping distinguish legitimate traffic from potential beaconing activity.

It identifies significant interactive phases (command execution) by analyzing interactivity duration, connection count, and detection score. The default parameter values (duration: 3600s, count: 24, score: 0.85) are set to avoid false positives, as illustrated in the following figure.

Data load

The function checks all IP flow data in the `../data` directory and loads them for further processing. Note that this notebook is able to **process biflows only**.

Check for data files

Load selected data files

Select data files to load:

```
../data/connections_flows_ips/connections_flows_ips--2024-09-19_09-21.csv
../data/connections_flows_ips/connections_flows_ips--2024-08-22_13-52.csv
../data/connections_flows_ips/connections_flows_ips--2024-08-22_13-52.json
../data/connections_flows_ips/connections_flows_ips--2024-08-22_13-17.json
../data/connections_flows_ips/sample_beacon_dataset.csv
../data/connections_flows_ips/connections_flows_ips--2024-08-22_13-17.csv
../data/connections_flows_ips/ipynb_checkpoints/connections_flows_ips--2024-08-22_13-17-checkpoint.csv
```

```
2024-09-19 09:30:53 Botnet_Communications [INFO]: Loading selected data files...
2024-09-19 09:30:53 Botnet_Communications [INFO]: Selected data files successfully loaded
```

Data processing

Minimal interactivity duration (s):

Minimal interactivity connection count:

Minimal detection score:

Process loaded IP biflows

```
2024-09-19 09:30:56 Botnet_Communications [INFO]: IP biflows processing...
2024-09-19 09:30:59 Botnet_Communications [INFO]: No beaconing data observed
```

Domains analysis

Analyze related domains

Figure 90: The looks of playbook implementing the detection algorithm

While this algorithm could be developed as a standalone detection tool, implementing it within a Jupyter notebook was more efficient given the need for occasional analysis of a specific network segment. Developing a full detection system would have required unnecessary effort for this one-off task. The notebook format allowed for quick execution and testing of the algorithm without significant development overhead (see Figure 90).

Combining CTIs, Asset Management, and Accounting

In the previous section, we introduced several scenarios focused on the analysis of IP flows, with their primary strength derived from network flow data. This section builds on that foundation by presenting analytical processes that incorporate additional data sources, such as Cyber Threat Intelligence (CTI), asset management, and accounting data to enhance analytical capabilities.

CTI-Enriched IP Data Analysis

To enable analysts to quickly assess the risk level of observed IP addresses and identify potential threats or malicious activity, integrating various CTI data throughout the analytical process is essential. This approach enhances analysts' decision-making and provides a more complete context by correlating network behavior with the known threat intelligence. Therefore, we have prepared a playbook that integrates multiple data sources, including asset management systems and CTI platforms, consolidating relevant information into a unified resource (see Figure 91).

The following are the primary sources we have used until now:

- **Netbox:** Netbox¹, as the solution for modeling and documenting modern networks, is used in our network for asset management. This playbook retrieves from our organization's Netbox instance basic details about assets like Tenant, Description, and Parent information for IP addresses associated with our network. The extracted data includes hypertext links to the corresponding Netbox pages for easy navigation.

While this algorithm could be developed as a standalone detection tool, implementing it within a Jupyter notebook was more efficient given the need for occasional analysis of a specific network segment. Developing a full detection system would have required unnecessary effort for this one-off task. The notebook format allowed for quick execution and testing of the algorithm without significant development overhead (see Figure 90).

Combining CTIs, Asset Management, and Accounting

In the previous section, we introduced several scenarios focused on the analysis of IP flows, with their primary strength derived from network flow data. This section builds on that foundation by presenting analytical processes that incorporate additional data sources, such as Cyber Threat Intelligence (CTI), asset management, and accounting data to enhance analytical capabilities.

CTI-Enriched IP Data Analysis

To enable analysts to quickly assess the risk level of observed IP addresses and identify potential threats or malicious activity, integrating various CTI data throughout the analytical process is essential. This approach enhances analysts' decision-making and provides a more complete context by correlating network behavior with the known threat intelligence. Therefore, we have prepared a playbook that integrates multiple data

sources, including asset management systems and CTI platforms, consolidating relevant information into a unified resource (see Figure 91).

The following are the primary sources we have used until now:

- **Netbox:** Netbox⁹⁷, as the solution for modeling and documenting modern networks, is used in our network for asset management. This playbook retrieves from our organization's Netbox instance basic details about assets like Tenant, Description, and Parent information for IP addresses associated with our network. The extracted data includes hypertext links to the corresponding Netbox pages for easy navigation.

▼ INFO

Classic_DNS

157.240.30.35 -> edge-star-mini-shv-01-prg1.facebook.com

TOR - TorProject

Tor: False

IP address details - IPAPI & Robtex

as	AS32934 Facebook, Inc.
isp	Facebook, Inc.
lat	50.0883
lon	14.4124
org	Facebook, Inc.
zip	110 00
city	Prague
region	10
country	Czechia
timezone	Europe/Prague
regionName	Prague
countryCode	CZ

as	32934
city	Menlo Park
asname	Facebook Facebook
country	United States
bgproute	157.240.30.0/24
routedesc	Facebook, Inc.
whoisdesc	Facebook, Inc. (THEFA-3)

► PASSIVE DNS

► WHOIS (RDAP)

Figure 91: Overview of IP address CTI data collected

- **IntelOwl:** IntelOwl⁹⁸ is an open-source platform designed to collect and analyze intelligence data about files, IP addresses, domains, and other indicators of compromise (IoCs) from multiple sources. In this Jupyter notebook, we use five chosen sources provided by IntelOwl that are sufficient for our needs.
- **Whois/RDAP:** Retrieves Whois records using the RDAP protocol, a modern alternative to the traditional Whois protocol, to provide ownership and administrative details of the IP address.

Analyzing observables to extract IoCs

In Incident Response (IR), extracting Indicators of Compromise (IoCs) is a fundamental step in detecting and addressing security threats. IoCs can be derived from various sources, including phishing emails, network traffic, system logs, and malware samples. These indicators—such as IP addresses, domains, file hashes, and URLs—provide

essential information for identifying and mitigating threats. The next critical step is to analyze these IoCs to uncover related indicators, expanding the investigation and revealing other hidden threats.

Once extracted, IoCs can be used to block malicious IPs, restrict access to phishing domains, or filter emails containing harmful file hashes. Additionally, IoCs can be cross-referenced with network traffic to identify communication with fraudulent domains or IP addresses exhibiting suspicious activity.

VirusTotal is a widely used tool for initial IoC analysis and validation. However, manual searching of VirusTotal can be time-consuming, especially when handling multiple daily incidents, making automation essential for efficiency.

This playbook is designed for incident response workflows, enabling IoCs analysis and extraction for observables such as IP addresses, domains, and file hashes. It seamlessly integrates with our IR team's workflows, allowing precomputed file hashes from email attachments, which are already done by the RTIR system used by our team, to be directly used for further investigation. Moreover, it automates the search for extracted IoCs within network traffic.

In summary, the implemented process involves three key steps:

1. The observable is analyzed using VirusTotal, displaying the number of analyzers marking it as malicious or undetected, with a link to the full report. This step confirms the malicious nature of the content (see Figure 92).
2. IoCs such as IP addresses, domains, and URLs associated with the observable are extracted, including indirect IoCs (one more pre-click in VirusTotal) if a file hash is involved.
3. The extracted IoCs are searched within the network traffic database over a specified period. IPs are matched with source/destination fields, domains with DNS QNAME, TLS SNI, and HTTP HOST fields, and URLs are parsed into domain (HTTP HOST) and path (HTTP URL) components (see Figure 93).

VirusTotal observable analysis

Analyses provided observable (hash, domain, or IP address) with VirusTotal and displays basic analysis results (amount of analyzers that classify the observable as malicious/undetected) with link to the full report.

Observable: Analyse observable

Analysis results:
MALICIOUS: 44/77
UNDETECTED: 21/77

LINK TO FULL REPORT: <https://www.virustotal.com/gui/search/93b4864fa57c5eb739676566b444a0b94f31c98bad882116d6b8680fa7a30337>

Figure 92: Analyzer assessment of the harmfulness of the IoCs in question

Select IP addresses:
 Select domains:
 Select URLs:

View: Limit: ☐ View limit Channel: IP flows type:

Start time: End time: Search IOCs in IP flows

Show entries

START TIME - FIRST SEEN	DURATION	PROTOCOL	SOURCE IP ADDRESS	SOURCE PORT	DESTINATION IP ADDRESS	DESTINATION PORT	TCP FLAGS	PACKETS	BYTES	DETECTED PROTOCOL	HTTP HOST	HTTP UR
2024-09-19T10:19:01.301+02:00	00m 00s	UDP	147.251.4.33	53	147.251.149.156	62877	---	1	326	DNS		

Figure 93: Screening network flows for the presence of extracted IoCs

M365 Defender Logs: User Activity Analysis

Within our organization, Microsoft M365 services are extensively utilized for email, data backup, and file sharing across teams and employees' projects. Leveraging M365 enables access to user activity logs, which are valuable for incident investigations.

In previous scenarios, we highlighted the importance of analyzing user behavior on the network during incident investigations. By incorporating both network flow and M365 logs, the detection of malicious activities, such as account compromise, is greatly enhanced. However, the M365 system generates a wide array of logs with differing syntax and context, which are not directly integrated. This complexity can hinder an analytical process.

In this playbook, these challenges are addressed by providing a unified visualization of user events from M365 Defender logs based on a specified user ID used within the organization. It consolidates logs from various M365 interfaces into a single view, streamlining the investigation of potential account compromises.

The notebook is divided into four key sections:

1. **Event Overview:** Obtaining user events from four M365 sources—SharePoint (file sharing), Sign-ins, General, and Exchange (email)—displaying the count of occurrences for each event type and source workload. This offers a snapshot of the user's overall activity.
2. **Category-Specific Events:** Focuses on events tied to specific categories, such as:
 - File events (from SharePoint)
 - Teams' events (from General with workload set to Microsoft Teams)
 - Mail events (from Exchange, see Figure 95)
 - Sharing events (from SharePoint, with keywords like "Sharing," "Access"—excluding "Accessed"—and "Link").
- This breakdown enables a more granular analysis of activity in each category, providing relevant attributes for each event type.
3. **Sign-In Events:** Displays sign-in data for the selected user, including key attributes like the time of login, device used, and location, offering insight into access patterns and potential unauthorized logins.
4. **User Indicators:** Extracts key indicators associated with the user's activities:
 - Objects found in the *objectid* field (files, URLs, etc.).
 - URLs from the *siteurl* field, listing URLs the user interacted with.
 - IP addresses from the *clientip* field show the user's IP address, except in cases where the field logs the IP of a host performing actions on behalf of the user.

Get events of selected category

Displays events of selected category with attributes that are relevant for given category. Categories are defined as follows:

- **File events** are events from the Opensearch MS365 `SharePoint` index pattern.
- **Teams events** are events from the Opensearch MS365 `General` index pattern, with workload set to `MicrosoftTeams`.
- **Mail events** are events from the Opensearch MS365 `Exchange` index pattern.
- **Sharing events** are events from the Opensearch MS365 `SharePoint` index pattern, with their operation containing one of the keywords `Sharing`, `Access` (but not `Accessed`!) or `Link`.

User: Start time: End time:

Get sign-ins

Displays sign-in events (with corresponding attributes) for selected user.

User: Start time: End time:

Figure 94: Illustrative use cases for the analysis of M365 logs

Accounting Logs User Traffic Analysis

The final scenario, similar to the preceding one, centers on the analysis of user behavior using Eduroam, VPN, and other LAN networks across the university, leveraging accounting logs as the principal data source.

The playbook is organized into two primary stages:

1. **User Session Retrieval:** This step identifies all user sessions via Eduroam, VPN or LAN within a designated time frame, utilizing the user's identifier specific to the university's environment. The logs are sourced from various systems, including RADIUS/VPN servers, routers, and other network devices, and are stored in a centralized database. These logs are retrieved via connectors, enabling their subsequent analysis within the playbook (see Figure 95).
2. **IP Flow Analysis:** In this stage, the accounting logs are correlated with the associated IP flows for the retrieved sessions, allowing for a comprehensive analysis. This section employs shared analytical mechanisms, such as filtering, aggregation, and hostname resolution, which are standard across other playbooks working with IP flows. This facilitates a thorough examination of network activity (see Figure 96).

accountStartTime	accountStopTime	framedIpAddress	sourceNetwork	data
2024-09-18 08:04:10+02:00	2024-09-18 08:24:58+02:00	100.66.76.41	EDUROAM	{'chargeableUserIdentity': 'ryccpNyEbLjyCB142Ry4f+Bouiw', 'eapIdentity': None, 'username': '445323@muni.cz'}

Figure 95: User session retrieval detail

START TIME - FIRST SEEN	DURATION	PROTOCOL	SOURCE IP ADDRESS	SOURCE PORT	DIRECTION	DESTINATION IP ADDRESS	DESTINATION PORT	TCP FLAGS	SENT
2024-09-18 08:04:10.526000+02:00	00m 00s	TCP	100.66.76.41	33484	→	18.133.140.134	80	.AP.SF	
2024-09-18 08:04:10.551000+02:00	00m 00s	TCP	100.66.76.41	33484	→	18.133.140.134	80	.AP.SF	
2024-09-18 08:09:10.651000+02:00	00m 00s	TCP	100.66.76.41	60306	→	18.133.140.134	80	.AP.SF	
2024-09-18 08:09:10.675000+02:00	00m 00s	TCP	100.66.76.41	60306	→	18.133.140.134	80	.AP.SF	
2024-09-18 08:14:10.651000+02:00	00m 00s	TCP	100.66.76.41	42708	⇄	18.159.254.57	80	.AP.SF	
2024-09-18 08:19:10.650000+02:00	00m 00s	TCP	100.66.76.41	57806	⇄	38.145.60.20	80	.AP.SF	
2024-09-18 08:23:40.731000+02:00	00m 00s	TCP	100.66.76.41	48992	→	18.133.140.134	80	S.	
2024-09-18 08:23:40.755000+02:00	00m 03s	TCP	100.66.76.41	48992	→	18.133.140.134	80	.A.S.	

Figure 96: The subsequent network flow analysis corresponding to retrieved session

5.10. Digital forensics and incident response on Microsoft Entra ID

Reacting to compromised user accounts

When the user is confirmed compromised, it is recommended to revoke user access in the shortest time possible. To fully revoke user access, it's required to either change password or disable user account, as well as revoke refresh tokens and session cookies.

Disabling user account or changing user password does not revoke current session, so it's strongly recommended to both revoke sessions and disable account / change user password.

Disabling user account via GUI

Prerequisites:

- A user account with access to the Entra ID admin panel, with permissions to disable user accounts, e.g. Global Administrator.

To disable user account, use the Microsoft Entra ID for workforce admin panel, navigate to "Users", select the required user account and follow these specific steps:

1. Click on "Edit properties"

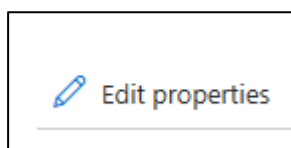


Figure 97: Disabling user account step 1

2. Click on "Settings"



Figure 98: Disabling user account step 2

3. Uncheck "Account enabled"

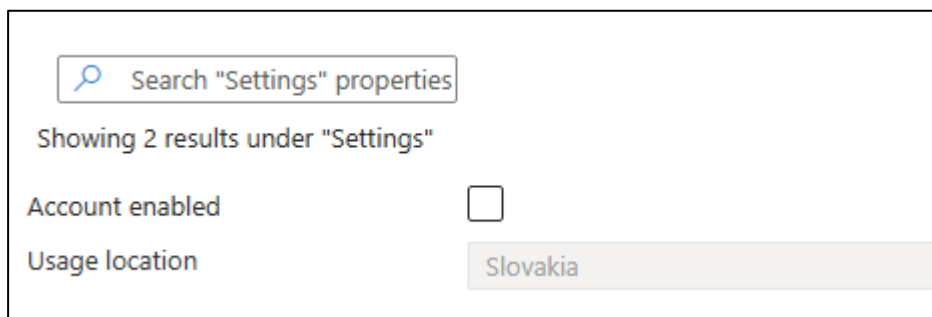


Figure 99: Disabling user account step 3

Disabling user account via PowerShell

Prerequisites:

- Microsoft.Graph PowerShell Module (tested on version 2.17.0)
- A privileged user account with permission to accept application requests (e.g. Global Administrator)

To programmatically disable user account, connect via the following command:

```
Connect-MgGraph -Scopes "User.ReadWrite.All"
```

using the Microsoft Graph PowerShell module and use the following command to disable a user account:

```
$User = Get-MgUser -Search UserPrincipalName:'johndoe@testdomain.com' -ConsistencyLevel eventual
```

```
Update-MgUser -UserId $User.Id -AccountEnabled:$false
```

Changing user password via GUI

Prerequisites:

- A user account with access to the Entra ID admin panel, with permissions to change user passwords, e.g. Global Administrator.

To change user password, use the Microsoft Entra ID for workforce admin panel, navigate to “Users”, select the required user account and follow these specific steps:

1. Click on “Reset password”

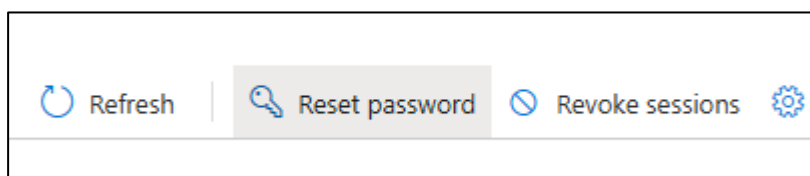


Figure 100: Changing user password step 1

2. In the pop-up side panel, click on “Reset password”.

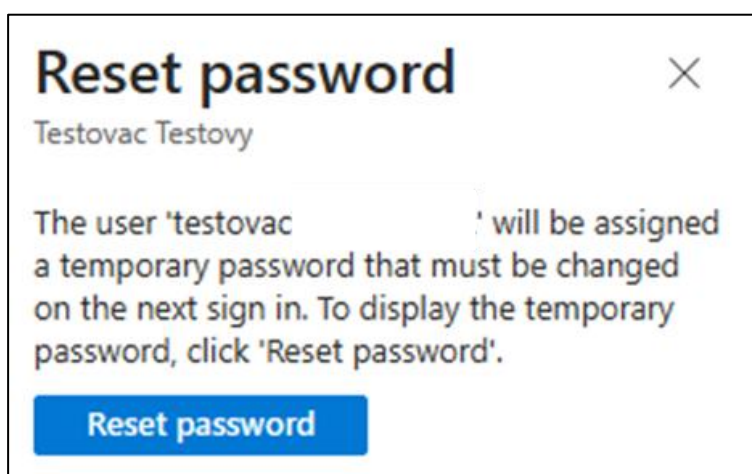


Figure 101: Changing user password step 2

3. A new, randomly generated password will be set. You are unable to modify this password or choose an arbitrary password.

Changing user password via PowerShell

Prerequisites:

- AzureAD PowerShell Module (tested on version 2.0.2.182)
- A privileged user account with permission reset passwords (e.g. Global Administrator)

To programmatically change the user password, connect via the following command using the Microsoft Graph Module:

```
Connect-MgGraph -Scopes "User.ReadWrite.All"
```

then use the following commands:

```
$Password = ConvertTo-SecureString -String "ExamplePassword123" -Force -AsPlainText
```

```
Set-AzureADUserPassword -ObjectId "usersObjectID" -Password $password
```

To get ObjectID of a specific user, use the following command:

```
Get-AzureADUser -ObjectId username@testdomain.com
```

Revoking sessions via GUI

Prerequisites:

- A user account with access to the Entra ID admin panel, with permissions to revoke sessions, e.g. Global Administrator.

To revoke user sessions, use the Microsoft Entra ID for workforce admin panel, navigate to “Users”, select the required user account and follow these specific steps:

1. Click on “Revoke sessions”

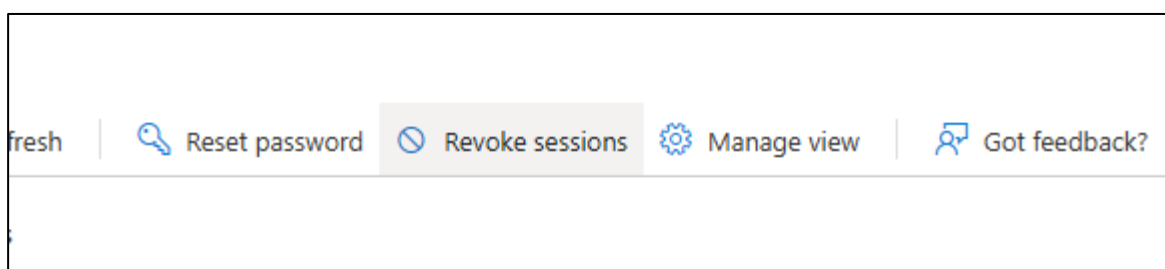


Figure 102: Revoking sessions step 1

2. Confirm the action

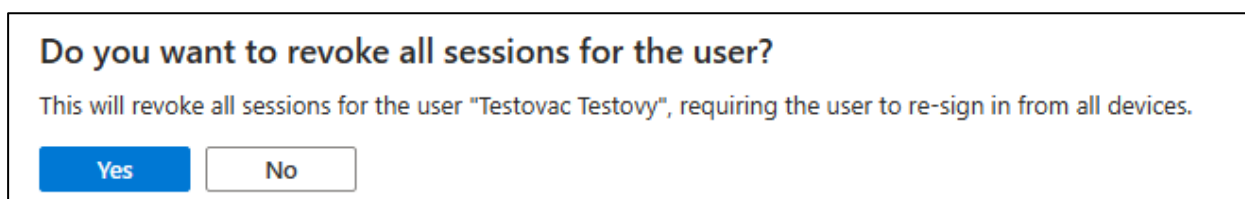


Figure 103: Revoking sessions step 2

Revoking sessions via PowerShell

Prerequisites:

- Microsoft.Graph PowerShell Module (tested on version 2.17.0)
- A privileged user account with permission to accept application requests (e.g. Global Administrator)

To programmatically revoke user sessions, connect via the following command using the Microsoft Graph Module:

```
Connect-MgGraph -Scopes "User.ReadWrite.All"
```

and use the following command:

```
Revoke-MgUserSignInSession -UserId name.surname@exampledomain.com
```

Azure / Entra ID forensics

During account investigation in Entra ID, it is recommended to extract tenant-wide sign-in logs, as well as activity logs of specific users.

User activity logs, contained within the unified activity logs, contain detailed information about both user and administrator activities in Entra ID, as well as Office 365 applications. Forensically relevant logged operations include:

- Application-related operations, such as registering or approving a new application.
- Changing tenant-wide policies with administrator privileges.
- Running eDiscovery or audit log searches.
- Modifying mailbox properties, such as creating new mailbox rules, or changing mailbox permissions.
- Accessing, modifying, or deleting files within SharePoint.

Full list of logged operations is available here: <https://learn.microsoft.com/en-us/purview/audit-log-activities?view=o365-worldwide>

Extracting sign-in logs using PowerShell

Warning: We recommend extracting sign-in logs via GUI, as console commands currently supported by Microsoft do not extract all relevant available information from the Entra ID.

Prerequisites:

- Microsoft.Graph PowerShell Module (tested on version 2.17.0)
- A privileged user account with permission to accept application requests (e.g. Global Administrator)

To programmatically extract tenant-wide sign-in logs, connect via the following command:

```
Connect-MgGraph -Scopes "AuditLog.Read.All"
```

using the Microsoft Graph PowerShell module and use the following command to extract sign-in logs:

```
Get-MgAuditLogSignIn
```

Example use:

```
Get-MgAuditLogSignIn -Filter "UserPrincipalName eq 'uset@testdomain.com'"
Get-MgAuditLogSignIn | Format-Table CreatedDateTime, UserPrincipalName,
AppDisplayName, ResourceDisplayName, IPAddress
```

See Microsoft' documentation for more information: <https://learn.microsoft.com/en-us/powershell/module/microsoft.graph.reports/get-mgauditlogsignin?view=graph-powershell-1.0>

Extracting sign-in logs using GUI

To extract tenant-wide sign-in logs via GUI, use the Microsoft Entra ID for workforce admin panel, navigate to “Sign-in logs” logs under “Monitoring” and follow these specific steps:

1. Select the desired date interval using the Date filter:

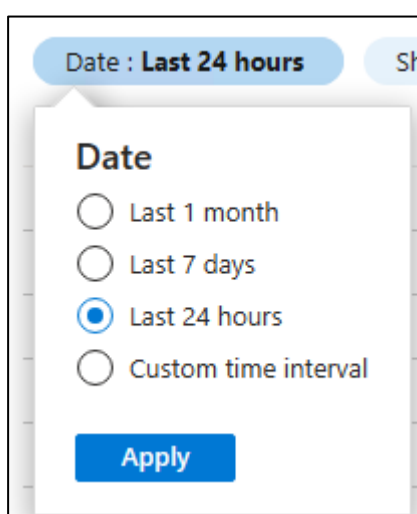


Figure 104: Extracting sign-in logs step 1

2. Click on Download.

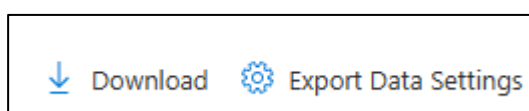


Figure 105: Extracting sign-in logs step 2

3. Select the desired format and click on Download.

To extract sign-in logs of a specific user, first navigate to “Users” and select the required user account.

Extracting audit using GUI

To extract tenant-wide Entra ID audit logs via GUI, use the Microsoft Entra ID for workforce admin panel, navigate to “Audit logs” logs under “Monitoring” and follow these specific steps:

1. Select the desired date interval using the Date filter:

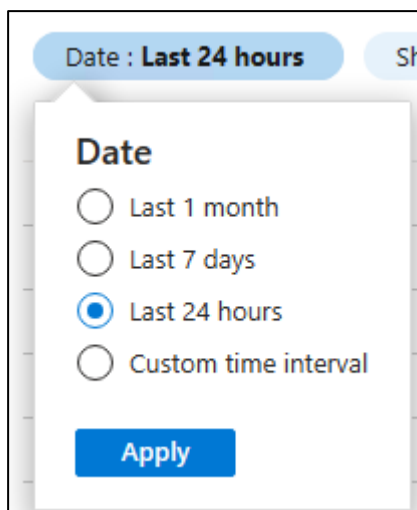


Figure 106: Extracting audit step 1

2. Click on Download.



Figure 107: Extracting audit step 2

3. Select the desired format and click on Download.

To extract audit logs of a specific user, first navigate to “Users” and select the required user account.

Extracting audit using PowerShell

Prerequisites:

- Microsoft.Graph PowerShell Module (tested on version 2.17.0)
- A privileged user account with permission to accept application requests (e.g. Global Administrator)

To programmatically extract tenant-wide audit logs, connect via the following command:

```
Connect-MgGraph -Scopes "AuditLog.Read.All"
```

using the Microsoft Graph PowerShell module and use the following command to extract audit logs:

```
Get-MgAuditLogDirectoryAudit
```

Example use:

```
Get-MgAuditLogDirectoryAudit -Filter "UserPrincipalName eq 'uset@testdomain.com'"
```

See Microsoft' documentation for more information: <https://learn.microsoft.com/en-us/powershell/module/microsoft.graph.reports/get-mgauditlogdirectoryaudit?view=graph-powershell-1.0>

Extracting activity (unified audit logs) logs using GUI

Activity logs can be extracted from the compliance.microsoft.com administrative portal following these specific steps:

1. Log in to the compliance.microsoft.com portal as a privileged user.
2. Click on “Audit” under “Solutions” in the side panel

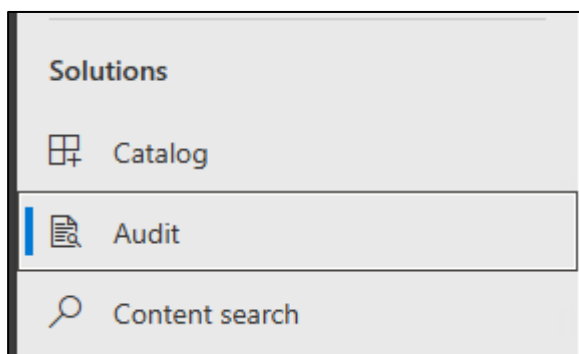


Figure 108: Extracting activity step 2

3. Specify a date and time range.

Audit

New Search

Audit retention policies

Searches completed | Active searches | Active unfiltered searches

1

0

0

Date and time range (UTC) *

Start	Apr 20 2024	00:00
End	Apr 22 2024	00:00

Figure 109: Extracting activity step 3

4. We also recommend that the user should be specified, especially in larger environments.

The screenshot shows a web interface for extracting activity. It contains three main sections, each with a text input field:

- Users:** The input field contains the text "user.user@testdomain.com".
- File, folder, or site:** The input field has a placeholder text "Enter all or a part of the name of a file, website, or folder".
- Workloads:** The input field has a placeholder text "Enter the workloads to search for" and a small downward arrow icon on the right side.

Figure 110: Extracting activity step 4

5. Click on Search.
6. To export a search, click on a completed search and click on "Export".

Extracting activity logs using PowerShell

Prerequisites:

- ExchangeOnlineManagement PowerShell Module (tested on version 3.1.0)
- A privileged user account with Unified Audit Log read permission (e.g. Global Administrator)

To programmatically extract per-user activity logs, connect via the following command:

Connect-ExchangeOnline

and use the following command to extract activity logs:

```
Search-UnifiedAuditLog -StartDate 9/1/2024 -EndDate 9/8/2024 -
SessionCommand ReturnLargeSet -UserIds "test.user@testdomain.sk"
```

Example use:

```
Search-UnifiedAuditLog -StartDate 9/1/2024 -EndDate 9/8/2024 -
SessionCommand ReturnLargeSet -UserIds test.user@testdomain.sk | Export-Csv
-Path C:\Users\User\Downloads\log.csv
```

Refer to Microsoft documentation for more information: <https://learn.microsoft.com/en-us/powershell/module/exchange/search-unifiedauditlog>

Alternatively, use a PowerShell script provided by Microsoft to programmatically extract activity logs. Refer to the Microsoft documentation for more information: <https://learn.microsoft.com/en-us/purview/audit-log-search-script>

Investigating compromised Entra ID accounts

When investigating compromised Entra ID / Office 365 accounts, the following artifacts are typically analyzed.

Azure Sign-in logs

Azure Sign-in logs contain information about all successful, interrupted or unsuccessful sign-ins, including Conditional Access policies results and geolocation for each sign-in. Sign-in logs can be exported in CSV or JSON formats for greater flexibility in regards to data analysis.

While sign-in logs are available for all tenants, Azure Sign-in logs are stored for 7 days for Microsoft Entra ID free licenses and for 30 days for Entra ID P1 and P2.

Note: Longer time windows of sign-in logs are available in Unified Audit Logs.

Sign-in logs are divided into 4 categories:

- Interactive sign-ins – logins performed by a user, e.g. login via online GUI
- Non-interactive sign-ins – logins performed by applications, e.g. login by OneDrive client
- Service principal sign-ins – sign-ins of a non-user service accounts, e.g. applications registered within Azure
- Managed identity sign-ins – logins by resources that used managed Azure credentials, e.g. VM with managed credentials

Out of the 4 log types, Interactive sign-ins and non-interactive sign-ins are most relevant when investigating compromised Office 365 accounts.

When analyzing sign-in logs of a compromised account, if possible, we recommend verifying each IP used for successful or unsuccessful sign-ins. In our experience, BEC gangs tend to use VPN connections as close as possible to the victim's real geographic location so relying purely on geolocation anomalies is not a reliable method of identifying illegitimate sign-ins. Suspicious logins can be identified using ISP attribution, whereas VPN and data center logins are usually considered suspicious.

You can also use log-ins to specific applications, such as applications user for Azure/Entra ID/Office 365 administration, as an indicator of suspicious activity.

Audit Logs (Entra ID audit logs, not Office 365 Unified Audit Logs)

Entra ID audit logs contain information about changes to objects within Entra ID – users, applications, or groups.

When investigating compromised accounts, the most relevant data included in the Audit Log include:

- Password change information, including self-service password resets and resets initiated by a privileged user.
- Account restrictions, such as revoking sessions and disabling user account.
- Other changes to account properties
- Billing or license changes
- Multi-factor authentication changes
- Azure application consents

Note: Some actions in Azure Logs are logged with Microsoft IP instead of the real client IP of the user who initiated the action.

Unified Audit Logs

Unified audit logs are centralized records that track user and admin activities across various Office 365 applications, such as Exchange Online, SharePoint, Teams, and OneDrive. These logs provide a log of events such as file access, user logins, and administrative changes. They are extracted per-mailbox.

Unified audit logs contain relevant information, including:

- SharePoint or OneDrive file access
- E-mail events, such as e-mail sent, deleted, created, or accessed
- User log-in events, including failed events
- Mailbox rule created, or deleted

A comprehensive list of all events included in the Unified Audit Logs is available in documentation provided by Microsoft: <https://learn.microsoft.com/en-us/purview/audit-log-activities>

Certain events, such as MailItemsAccess, require a deeper understanding of Exchange Online synchronization to fully understand the attacker's activities. Sync access is evidence of an automated access, likely by a desktop e-mail client. Bind access is evidence of individual access to specific e-mails. See Microsoft's documentation for more information regarding MailItemsAccess: <https://learn.microsoft.com/en-us/purview/audit-log-investigate-accounts>

Note: Audit Log Ingestion must be enabled in the tenant. To verify the Audit Log Ingestion status, use the following commands:

```
Connect-ExchangeOnline
```

```
Get-AdminAuditLogConfig | FL UnifiedAuditLogIngestionEnabled
```

eDiscovery Content Search

eDiscovery content search is used in forensic investigations to locate specific emails and files by scanning digital data for relevant keywords, metadata, or file types. It allows investigators to filter results by date, sender, or recipient, ensuring targeted and efficient extraction of relevant evidence for legal or investigative purposes.

To run an eDiscovery Content search, follow these steps:

1. Navigate to the following link: <https://compliance.microsoft.com/contentsearchv2> and log in as a user privileged enough to create new content search and export them, e.g. Compliance Administrator.
2. Click on "New search"

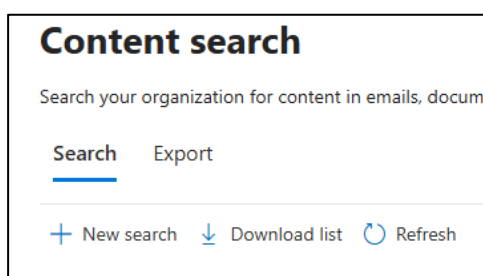


Figure 111: eDiscovery Content step 2

3. Fill in a name for the search and click on Next

Figure 112: eDiscovery Content step 3

4. Select locations for the search, usually Exchange or SharePoint.

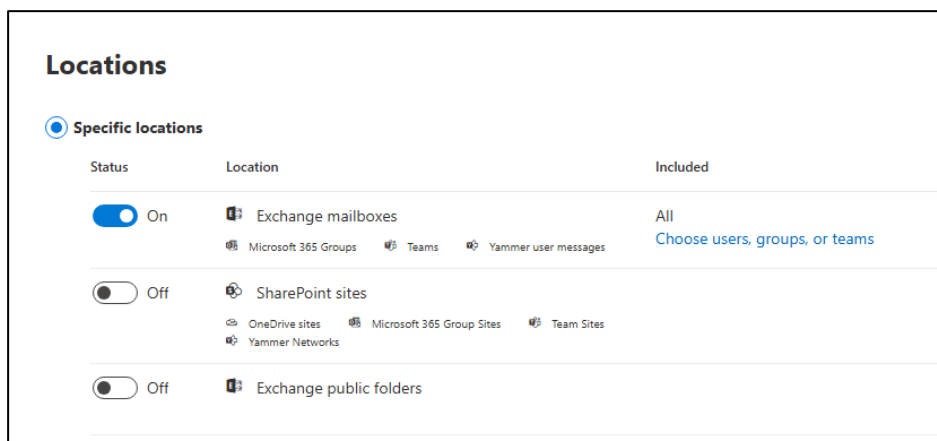


Figure 113: eDiscovery Content step 4

5. Build a query based on the search requirements.

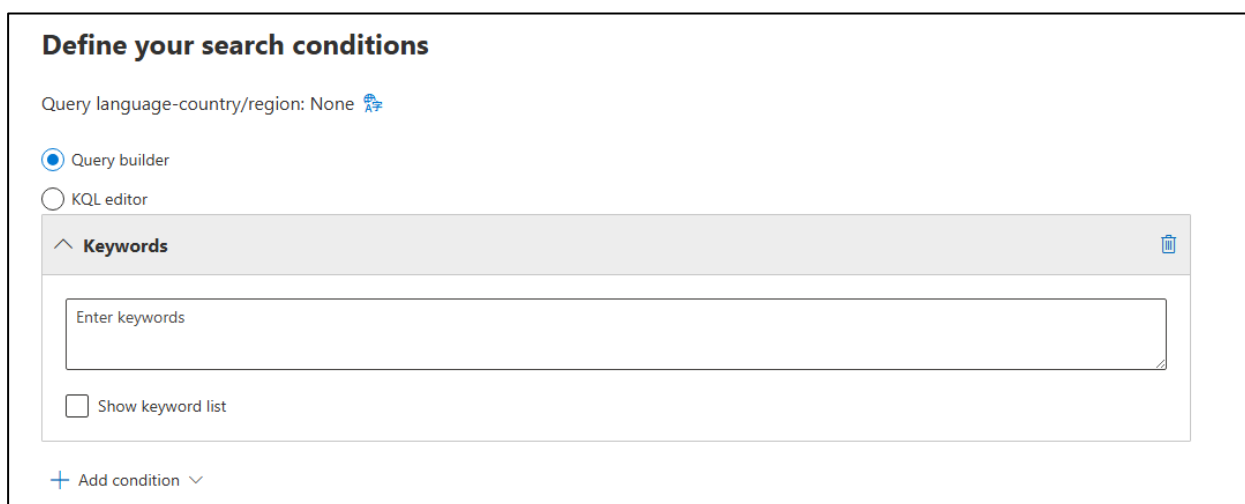


Figure 114: eDiscovery Content step 5

6. Submit search.

To download search results, follow these steps.

1. Click on a finished search, then click on “Actions” and next on “Export results”.

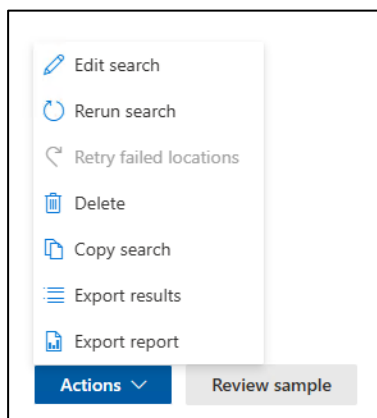


Figure 115: eDiscovery Content – download search results step 1

2. Select export parameters, usually “All items, excluding ones that have an unrecognized format, are encrypted, or weren’t indexed for other reasons” and “One PST file for each mailbox” and click on Export.
3. After a while switch the “Export” tab and click on the finished export.

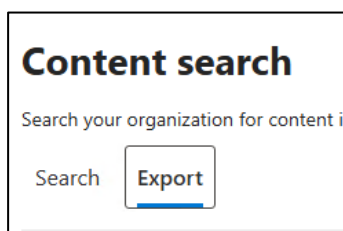


Figure 116: eDiscovery Content – download search results step 3

4. Click on “Download results.”

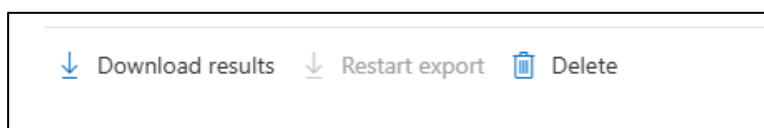


Figure 117: eDiscovery Content – download search results step 4

5. Follow the instruction to download and run the eDiscovery export tool. Enter location and export keys per export results.

Security monitoring

We recommend monitoring Entra ID infrastructure for logins to suspicious applications and/or from suspicious locations.

Suspicious applications are such applications, that are typically not used/accessed by a vast majority of users. They include, though are not limited to, administrative consoles and portals, such as:

- Azure Portal
- Microsoft Azure CLI
- Azure Active Directory PowerShell
- Microsoft Azure PowerShell
- Graph Explorer
- Azure DevOps ACOM Azure Website

We also recommend verifying logins from uncommon or unusual locations, such as logins from different continents and certain, high-risk countries.

Sign-in logs can be viewed and extracted in CSV format in the Microsoft Entra ID for workforce admin panel:

https://aad.portal.azure.com/#view/Microsoft_AAD_UsersAndTenants/UserManagementMenuBlade/~/_/SignInLogs

Tenants with Microsoft 365 Cloud App Security, or Microsoft Defender for Office 365 can also create and investigate automated or custom detection rules. Cloud App Security detection rules can be managed and viewed in the unified security.microsoft.com XDR portal.

Geoblocking

During an active incident and incident response process it is advisable to restrict access to and monitor the Microsoft Entra ID infrastructure to ensure that only legitimate users should sign in. Even during on-premises domain compromise events, attackers can potentially gain access to the cloud environment through locally stored passwords, password reuse, or password synchronization in hybrid domains.

The recommended approach is to utilize Conditional Access policies, available in the Microsoft Entra ID P1 license, and higher⁹⁹. Through Conditional Access, access can be limited to specific geolocations based on either IP geolocation or GPS geolocation using the Microsoft Authenticator. Additionally, Conditional Access supports the definition of custom locations based on IP ranges, allowing access only from specific physical sites or VPN ranges.

To define custom trusted locations, access the Microsoft Entra ID for workforce admin panel, navigate to „Security“, „Named locations“ and select either „Countries location“ or „IP ranges location“.



Figure 118: Geoblocking

For modifying Conditional Access policies, access the Microsoft Entra ID for workforce admin panel, navigate to „Security“, „Conditional Access“ and click „Create new policy“.

To create a geoblocking policy, follow these specific steps:

1. Name your policy

 A screenshot of the 'Name' field in the Microsoft Entra ID Conditional Access policy creation interface. The field is labeled 'Name *' and contains the text 'Geoblocking example'. A green checkmark is visible at the end of the input field, indicating that the name is valid.

Figure 119: Geoblocking step 1

2. Select which users should the policy apply to. When all users should be included, select “All users” under Include.

Include Exclude

☐ None

☒ All users

☐ Select users and groups

 Don't lock yourself out! This policy will affect all of your users. We recommend applying a policy to a small set of users first to verify it behaves as expected. [Learn more](#)

Figure 120: Geoblocking step 2a

IMPORTANT: Make sure to exclude yourself and/or one emergency access admin account to prevent complete lockout out of the tenant.

Include **Exclude**

Select the users and groups to exempt from the policy

☐ Guest or external users ⓘ

☐ Directory roles ⓘ

☒ Users and groups

Select excluded users and groups

1 user

Figure 121: Geoblocking step 2b

3. Under Target resources, select “All cloud apps”.

Select what this policy applies to

Cloud apps

Include Exclude

☐ None

☒ All cloud apps

☐ Select apps

 Don't lock yourself out! This policy impacts the Azure portal. Before you continue, ensure that you or someone else will be able to get back into the portal. Disregard this warning if you are configuring persistent browser session policy that works correctly only if "All cloud apps" are selected. [Learn more](#)

Figure 122: Geoblocking step 3

- Under Conditions, exclude allowed or trusted locations that you have pre-defined.

Configure ⓘ

Yes No

Include **Exclude**

Select the locations to exempt from the policy

☐ All trusted locations

☒ Selected locations

Select

Slovakia

Slovakia ...

Figure 123: Geoblocking step 4

- Under Grant, click on "Block access".

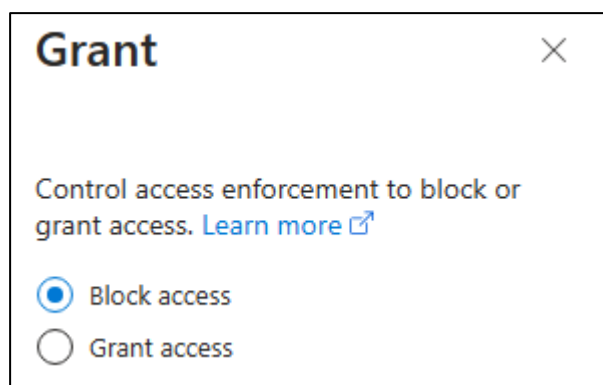


Figure 124: Geoblocking step 5

6. Enable the policy and click on Create.

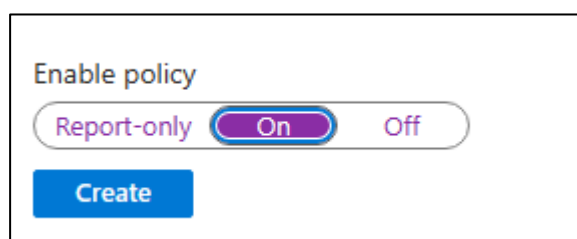


Figure 125: Geoblocking step 6

References

- [1] CICHONSKI P. et al.: Computer security incident handling guide. NIST Special Publication, 2012, 800.61: 1-147 [online] [13.3.2024], available at <https://csrc.nist.gov/pubs/sp/800/61/r2/final>
- [2] Directive (EU) No. 2016/1148 of the European Parliament and of the Council of 6 July 2016, concerning measures for achieving a high common level of security of network and information systems across the European Union
- [3] Directive (EU) No. 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for achieving a high common level of cybersecurity across the European Union, amending Regulation (EU) No. 910/2014 and Directive (EU) No. 2018/1972, and repealing Directive (EU) No. 2016/1148 (NIS 2 Directive)
- [4] Regulation (EU) No. 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and the free movement of such data, and repealing Directive No. 95/46/EC (GDPR)
- [5] FIRST: The Traffic Light Protocol (TLP) – Standards Definitions and Usage Guidance, August 2022 [online] [13.3.2024], available at <https://www.first.org/tlp/>
- [6] ISO/IEC 27001:2022 – Information security, cybersecurity and privacy protection – Information security management systems
- [7] ISO/IEC 27001:2022 – Information security, cybersecurity and privacy protection – Information security management systems
- [8] ENISA: Reference Incident Classification Taxonomy, Task Force Status and Way Forward, January 2018, [online] [13.3.2024], available at https://www.enisa.europa.eu/publications/reference-incident-classification-taxonomy/at_download/fullReport
- [9] STIKVOORT, D.: Incident Classification / Incident Taxonomy according to eCSIRT.net – adapted, [online] [13.3.2024], available at <https://www.trusted-introducer.org/Incident-Classification-Taxonomy.pdf>
- [10] CIRCL.LU: CIRCL Taxonomy – Schemes of Classification in Incident Response and Detection, [online] [13.3.2024], available at <https://www.circl.lu/pub/taxonomy/>
- [11] EUROPOL: Common Taxonomy for Law Enforcement and The National Network of CSIRTs, [online] [13.3.2024], available at https://www.europol.europa.eu/cms/sites/default/files/documents/common_taxonomy_for_law_enforcement_and_csirts_v1.3.pdf
- [12] CICHONSKI P. et al.: Computer security incident handling guide. NIST Special Publication, 2012, 800.61: 1-147 [online] [10.3.2024], available at <https://csrc.nist.gov/pubs/sp/800/61/r2/final>
- [13] <https://www.csirt.gov.sk/nahlasit-incident.html?csrc=2677669608893138056>
- [14] Incident Classification / Incident Taxonomy according to eCSIRT.net – adapted, available at <https://www.trusted-introducer.org/Incident-Classification-Taxonomy.pdf>
- [15] <https://www.talosintelligence.com/>
- [16] <https://ip-api.com/>
- [17] <https://www.abuseipdb.com/>

- [18] <https://www.virustotal.com/>
- [19] <https://www.unodc.org/e4j/zh/cybercrime/module-6/key-issues/handling-of-digital-evidence.html>
- [20] <https://www.kroll.com/en/insights/publications/cyber/kroll-artifact-parser-extractor-kape>
- [21] <https://learn.microsoft.com/en-us/sysinternals/downloads/>
- [22] <https://www.kroll.com/en/services/cyber-risk/incident-response-litigation-support/kroll-artifact-parser-extractor-kape>
- [23] <https://ericzimmerman.github.io/#!index.md>
- [24] <https://github.com/EricZimmerman/MFTECmd>
- [25] <https://f001.backblazeb2.com/file/EricZimmermanTools/net6/MFTExplorer.zip>
- [26] <https://github.com/EricZimmerman/WxTCmd>
- [27] <https://en.wikipedia.org/wiki/ROT13>
- [28] https://www.nirsoft.net/utils/userassist_view.html
- [29] <https://github.com/EricZimmerman/AppCompatCacheParser>
- [30] <https://github.com/EricZimmerman/AmcacheParser>
- [31] <https://github.com/EricZimmerman/Srum>
- [32] <https://github.com/EricZimmerman/JLECmd>
- [33] <https://github.com/EricZimmerman/LECmd>
- [34] <https://github.com/EricZimmerman/RBCmd>
- [35] https://www.nirsoft.net/utils/browsing_history_view.html
- [36] https://www.nirsoft.net/utils/web_browser_downloads_view.html
- [37] <https://github.com/obsidianforensics/hindsight>
- [38] <https://man7.org/linux/man-pages/man8/anacron.8.html>
- [39] <https://man.archlinux.org/man/logwatch.8.en>
- [40] <https://www.systutorials.com/docs/linux/man/1-mboxgrep/>
- [41] <https://man.archlinux.org/man/grub-mkconfig.8.en>
- [42] <https://www.mankier.com/1/grub2-editenv>
- [43] <https://man7.org/linux/man-pages/man1/hostnamed.1.html>
- [44] <https://man7.org/linux/man-pages/man1/domainname.1.html>
- [45] <https://www.freedesktop.org/software/systemd/man/latest/journalctl.html>
- [46] <https://man7.org/linux/man-pages/man1/dmesg.1.html>
- [47] <https://lowfatlinux.com/linux-zmore-zless.html>
- [48] <https://linux.die.net/man/8/auditctl>
- [49] <https://linux.die.net/man/8/auditd>
- [50] <https://man7.org/linux/man-pages/man8/ausearch.8.html>

- [51] <https://linux.die.net/man/8/aureport>
- [52] <https://linux.die.net/man/5/crontab>
- [53] <https://man7.org/linux/man-pages/man1/last.1.html>
- [54] <https://man7.org/linux/man-pages/man1/who.1.html>
- [55] <https://linux.die.net/man/1/lastb>
- [56] <https://man7.org/linux/man-pages/man1/utmpdump.1.html>
- [57] <https://man7.org/linux/man-pages/man8/dumpe2fs.8.html>
- [58] <https://www.sleuthkit.org/sleuthkit/man/fsstat.html>
- [59] <https://www.sleuthkit.org/sleuthkit/man/mmls.html>
- [60] <https://linux.die.net/man/1/disktype>
- [61] <https://man7.org/linux/man-pages/man8/fdisk.8.html>
- [62] <https://linux.die.net/man/8/gdisk>
- [63] <https://linux.die.net/man/8/parted>
- [64] <https://www.commandlinux.com/man-page/man1/logger.1.html>
- [65] <https://www.man7.org/linux/man-pages/man1/systemctl.1.html>
- [66] <https://celebrite.com/en/ufed/>
- [67] <https://www.msab.com/product/xry-extract/>
- [68] <https://github.com/abrignoni/iLEAPP>
- [69] <https://volatilityfoundation.org/>
- [70] Jirsik, T., Cermak, M., Drichel, A., Mandal, A., Palumbo, P. (2020). D3.1 Data selection and preparation. SAPPAN Consortium. https://sappan-project.eu/?page_id=14
- [71] Hofstede, R., Celeda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A., and Pras, A. (2014). Flow monitoring explained: From packet capture to data analysis with NetFlow and IPFIX. IEEE Communications Surveys and Tutorials, 16(4), 2037–2064. <https://doi.org/10.1109/COMST.2014.2321898>
- [72] Weber, J. (2006). The Fundamentals of Passive Monitoring Access. <https://www.nanog.org/meetings/nanog37/presentations/joy-weber.pdf>
- [73] Tomáš Jirsík. Cyber Situation Awareness via IP Flow Monitoring. Online. Doctoral theses, Dissertations. Brno: Masaryk University, Faculty of Informatics. 2019. Available from: <https://is.muni.cz/th/ejynv/>.
- [74] Paxson, V., Mahdavi, J., Adams, A., and Mathis, M. (1998). An Architecture for Large-scale Internet Measurement. IEEE Communications Magazine, 36(8), 48–54. <https://doi.org/10.1109/35.707817>
- [75] European Parliament, Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). 2016. <https://eur-lex.europa.eu/eli/reg/2016/679/oj>
- [76] A. Oliner and J. Stearley. What Supercomputers Say: A Study of Five System Logs. In 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07), pages 575–584, 2007.

[77] <https://netbox.readthedocs.io>

[78] <https://snipeitapp.com>

[79] Tomáš Jirsík, Milan Čermák, Daniel Tovarňák and Pavel Čeleda. Toward Stream-Based IP Flow Analysis. IEEE Communications Magazine. IEEE, 2017, vol. 55, No 7, p. 70-76. ISSN 0163-6804. Available from: <https://dx.doi.org/10.1109/MCOM.2017.1600972>.

[80] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network Anomaly Detection: Methods, Systems and Tools. IEEE Communications Surveys & Tutorials, 16(1), 303–336. <https://doi.org/10.1109/SURV.2013.052213.00046>

[81] Muhammad Fahad Umer, Muhammad Sher, and Yaxin Bi. Flow-based intrusion detection: Techniques and challenges. 2017. Doi: 10.1016 /j.cose.2017.05.009

[82] <https://www.wireshark.org/>

[83] <https://tshark.dev/>

[84] <https://wiki.wireshark.org/DisplayFilters>

[85] <https://arkime.com/>

[86] <https://gchq.github.io/CyberChef/>

[87] <https://arkime.com/faq#how-do-i-import-existing-pcaps>

[88] <https://www.brimdata.io/>

[89] <https://docs.zeek.org/en/master/script-reference/log-files.html>

[90] <https://zed.brimdata.io/docs/>

[91] <https://zed.brimdata.io/docs/language/overview/>

[92] <https://jupyter.org/about>

[93] <https://www.similarweb.com/top-websites/>

[94] <https://tranco-list.eu/>

[95] <https://www.virustotal.com/gui/home/upload>

[96] <https://github.com/Cyb3r-Monk/ACCD>

[97] <https://netboxlabs.com/docs/netbox/en/stable/>

[98] <https://github.com/intelowlproject/IntelOwl>

[99] To use Conditional Access policies, security defaults policy must be disabled.

Table of figures

1. Incident handling and response

Figure 1 - Phases of security incident resolution according to NIST 800-61 r2.	15
Figure 2 - Proposed phases of security incident resolution	16

2. Specific procedures for security incidents

Figure 3 - Incident handling procedure	42
Figure 4 - Windows File properties extracted with PowerShell command.	44
Figure 5 - Windows File Digital signature extracted with PowerShell command.	45
Figure 6 - Windows file properties in GUI	45
Figure 7 - Linux file properties	45
Figure 8 - CyberChef Detect File Type Recipe.	46
Figure 9 - CyberChef Open File as Input.	47
Figure 10 - CyberChef Detect File Type Output.	47
Figure 11 - Identifying the file type using PSPad (.exe)	47
Figure 12 - SHA256 hash of a file obtained via PowerShell cmdlet.	49
Figure 13 - SHA256 hash of a file obtained using the Bash command.	49
Figure 14 - Example use of CAPA on test file.	49
Figure 15 - Capabilities of a test file revealed by CAPA	49
Figure 16 - File selection confirmation in IDAFree window	50
Figure 17 - Preview of a newly opened file in IDAFree.	51
Figure 18 - "Strings" window in IDAFree.	51
Figure 19 - Saving the output from Process Explorer.	53
Figure 20 - Saving the output from the Autoruns tool.	53
Figure 21 - Saving the output from the ListDlls tool.	54
Figure 22 - Network communication monitored with WireShark.	54

Figure 23 - Saving a completed network communication trace using the WireShark tool.	55
Figure 24 - E-mail incident resolution process	62
Figure 25 - End of an e-mail header	64
Figure 26 - Example of header attributes (To, From, Subject, Return-Path)	66
Figure 27 - Location of IP addresses	66

3. Digital forensics

Figure 28 - Initial phases of forensic investigation according to ISO/IEC 27037:2012 [1]	80
Figure 29 - Suspend a virtual machine	88
Figure 30 - Download the images of the virtual machine	88
Figure 31 - Pausing virtual machine	89
Figure 32 - Extracting virtual machine	89
Figure 33 - Exporting virtual machine	90
Figure 34 - Graphical interface of KAPE	93
Figure 35 - Main Target options	92
Figure 36 - Possible transfer options	93
Figure 37 - Using targets and options at once.	93
Figure 38 - Main options for Modules	93

4. Live forensics

5. Digital artifacts analysis

Figure 39 - Sample output created from \$MFT using the MFTECmd tool.	119
Figure 40 - Example of Available bookmarks	122
Figure 41 - Example of a security event log in Event Viewer	125
Figure 42 - Sample of a loaded prefetch file in PECmd	127
Figure 43 - Graphical Interface of the UserAssistView tool	129

Figure 44 - Example output from RegRipper using recentapps plugin.	130
Figure 45 - Example of a shimcache record extracted by AppCompatCacheParser	131
Figure 46 - Sample output from AmcacheParser, specifically a record of program execution.	133
Figure 47 - Sample output from the SrumECmd tool	134
Figure 48 - Example of scheduled tasks in Autoruns	135
Figure 49 - A sample record of a folder recently accessed using the JLECmd tool	137
Figure 50 - Example of the RBCmd tool.	142
Figure 51 - Example of the BrowserDownlaodView tool.	144
Figure 52 - Example of identified browser artifacts.	146
Figure 53 - Example of Chrome extensions in SQLite database (extracted by Hindsight).	147
Figure 54 – imageinfo.	187
Figure 55 - pslist.	187
Figure 56 - psscan.	188
Figure 57 - pstree.	188
Figure 58 - procdump.	189
Figure 59 - handles.	189
Figure 60 - psxview.	189
Figure 61 - cmdline.	190
Figure 62 - malfind.	190
Figure 63 - dlllist.	191
Figure 64 - netscan.	191
Figure 65 - hivescan.	192
Figure 66 - hivelist.	192

Figure 67 - printkey.	193
Figure 68 - filescan.	193
Figure 69 - Levels of abstraction of the network monitoring data	196
Figure 70 - IP Flow data processing workflow	198
Figure 71 - Taxonomy of anomaly and intrusion detection techniques	199
Figure 72 - Wireshark main window	200
Figure 73 - Wireshark Statistics menu	201
Figure 74 - Wireshark interactive filter selection	202
Figure 75 - Wireshark full text search	202
Figure 76 - Wireshark name resolution options	203
Figure 77 - Wireshark Follow TCP Stream functionality	203
Figure 78 - Arkime Sessions page	204
Figure 79 - Arkime connection details	205
Figure 80 - Arkime SPIView page	205
Figure 81 - Arkime Connections page	206
Figure 82 - Zui initial overview	208
Figure 83 - Zui connection detail	208
Figure 84 - Passive and active hostname resolution detail	210
Figure 85 - The network connections of the defined IP address within the time window	211
Figure 86 - Output for connections to query the specified domains in each time window	212
Figure 87 - Evaluation of the benignity of domains interacting with the specified IP address	213
Figure 88 - The example of statistical output	214
Figure 89 - Visualization of data uploads and downloads to service-specific servers	214

Figure 90 - The looks of playbook implementing the detection algorithm	215
Figure 91 - Overview of IP address CTI data collected	217
Figure 92 - Analyzer assessment of the harmfulness of the IoCs in question	218
Figure 93 - Screening network flows for the presence of extracted IoCs	219
Figure 94 - Illustrative use cases for the analysis of M365 logs	220
Figure 95 - User session retrieval detail	220
Figure 96 - The subsequent network flow analysis corresponding to retrieved session	221
Figure 97 - Disabling user account step 1	221
Figure 98 - Disabling user account step 2	222
Figure 99 - Disabling user account step 3	222
Figure 100 - Changing user password step 1	223
Figure 101 - Changing user password step 2	223
Figure 102 - Revoking sessions step 1	224
Figure 103 - Revoking sessions step 2	224
Figure 104 - Extracting sign-in logs step 1	226
Figure 105 - Extracting sign-in logs step 2	226
Figure 106 - Extracting audit step 1	228
Figure 107 - Extracting audit step 2	227
Figure 108 - Extracting activity step 2	228
Figure 109 - Extracting activity step 3	228
Figure 110 - Extracting activity step 4	229
Figure 111 - eDiscovery Content step 2	231
Figure 112 - eDiscovery Content step 3	231
Figure 113 - eDiscovery Content step 4	232

Figure 114 - eDiscovery Content step 5	232
Figure 115 - eDiscovery Content – download search results step 1	232
Figure 116 - eDiscovery Content – download search results step 3	236
Figure 117 - eDiscovery Content – download search results step 4	233
Figure 118 - Geoblocking	234
Figure 119 - Geoblocking step 1	234
Figure 120 - Geoblocking step 2a	236
Figure 121 - Geoblocking step 2b	235
Figure 122 - Geoblocking step 3	236
Figure 123 - Geoblocking step 4	236
Figure 124 - Geoblocking step 5	237
Figure 125 - Geoblocking step 6	237