Harmonizing People, Processes, and Technology for Robust Cybersecurity

D3.2 Risk Scenario Analysis Toolkit V1

Document Summary Information

Grant Agreement No	101158555	**Acronym**	CYberSynchrony
Full Title	Harmonizing People, Processes, and Technology for Robust Cybersecurity		
Start Date	01/06/2024	**Duration**	36 months
Project URL	https://www.CYberSynchrony.eu/		
Deliverable	D3.2 Risk Scenario Analysis Toolkit V1		
Work Package	WP3 Comprehensive Threat Intelligence and Customized Risk Evaluation (CYBERRA) and Cybersecurity Resilience and Incident Response Enhancement (CYRESCUE)		
Contractual due date	31/05/2025	**Actual submission date**	05/06/2025
Type	Other	**Dissemination Level**	Public
Lead Beneficiary	Meridian Group srl (MER)		
Responsible Author	Andrea Purificato (MER)		
Contributions from	Andrea Purificato (MER), Dimitris Koutras, Malamas Vangelis (FP)		

Funded by the European Union

ECCC

Funded by the European Union under Grant Agreement no. 101158555 and supported by the European Cybersecurity Competence Centre (ECCC). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Cybersecurity Competence Centre. Neither the European Union nor the granting authority can be held responsible for them.

Revision history

Version	Issue Date	Changes	Contributor(s)
0.1	17.03.2025	Initial Deliverable structure	Laskarina Mavropoulou (MER)
0.2	02.04.2025	Initial review and compilation	Andrea Purificato (MER)
0.3	29.04.2025	Review	Andrea Purificato (MER)
0.4	02.05.2025	Draft ready for first review	Andrea Purificato (MER), Dimitris Koutras (FP)
0.5	26.05.2025	First draft addressing comments from internal review	Andrea Purificato (MER)
0.6	27.05.2025	Addressing comments, new additions and corrections	Dimitris Koutras, Malamas Vangelis (FP)
0.7	28.05.2025	Corrections	Andrea Purificato (MER)
0.8	04.06.2025	Final quality check	Panayiotis Klitou (EBOS)
1.0	04.06.2025	Final version ready for submission	Andrea Purificato (MER)

Disclaimer

The content of the document herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the CYberSynchrony consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the CYberSynchrony consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the CYberSynchrony consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

Copyright message

© CYberSynchrony Consortium This Deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorised provided the source is acknowledged.

[1] According to CYberSynchrony's Quality Assurance Process for reports:

- 3 months before Deliverable's Due Date: Table of Contents ready to be reviewed by WP leader/ Technical Manager/ Quality Manager
- 2 months before Deliverable's Due Date: 50% should be complete. Review by the Quality Manager (& Technical Manager for software).
- 1 month before Deliverable's Due Date:100% should be complete. Review by the Quality Manager and Peer Reviewers.
- M: Sent to PC for Submission to the EC after addressing all comments by Quality Manager and Peer Reviewers

Table of Contents

Executive Summary 7
1 Introduction 8
1.1 Mapping CYberSynchrony Outputs 8
1.2 Linkage to other project outputs 9
1.3 Deliverable Structure 10
2 Position in CYberSynchrony Architecture 12
3 Kitsune Monitoring 14
3.1 Component Description 14
3.2 Architecture 15
3.3 Technologies Used 15
4 Cybersecurity Posture Assessment (CSPA) 16
4.1 Component Description 16
4.2 Architecture 17
4.3 Technologies Used 18
5 Attack Surface Assessment (ASA) 20
5.1 Component Description 20
5.2 Architecture 20
5.3 Technologies Used 22
6 Purple Team Assessment (PTA) 23
6.1 Component Description 23
6.2 Architecture 23
6.3 Technologies Used 24
7 Conclusions 26
8 References 27
Annex I: *Preliminary PoC and Prototypes (Kitsune Monitoring)* 28
Monitoring Configuration / Onboarding 28
YAML File (Kitsune Monitoring, Search) 28
Mock Prototype / Demo 28
Authentication/Authorization API 28
Monitoring API (Customer Notification) 30
Annex II: *Preliminary PoC and Prototypes (Cybersecurity Posture Assessment)* 33
Annex III: *Preliminary PoC and Prototypes (Attack Surface Assessment)* 35
Annex IV: *Preliminary PoC and Prototypes (Purple Team Assessment)* 40

List of Figures

Figure 1: Risk Scenario Analysis toolkit components and architecture overview 13
Figure 2: Kitsune Monitoring interactions overview 15
Figure 3: Sans Top 20 Critical Security Controls 16
Figure 4: diagram illustrating the architecture of the Cybersecurity Posture Assessment component 18
Figure 5: ASA Architecture 21
Figure 6: diagram illustrating the architecture of the PTA component 24
Figure 6: diagram illustrating the API login process 29

© CYberSynchrony, 2024

Figure 7: Diagram illustrating the API refresh authorization process 30
Figure 8: representation of “Alerts” functionality with response 30
Figure 10: gathering asset inventories through ansible (1) 35
Figure 11: gathering asset inventories through ansible (2) 36
Figure 12: gathering asset inventories through ansible (3) 37
Figure 13: gathering asset inventories through ansible (4) 38
Figure 13: correlating CVEs 39
Figure 15: Wazuh Nmap Related Generated Event (1) 40
Figure 16: Wazuh Nmap Related Generated Event (2) 41
Figure 17: Wazuh Nmap Related Generated Event (3) 41
Figure 18: Wazuh Nmap Related Generated Event (4) 42
Figure 19: Wazuh Brute Forcing Related Event Detailed (1) 42
Figure 20: Wazuh Brute Forcing Related Event Detailed (2) 43
Figure 21: Wazuh Firewall IP Block from Directory bruteforcing Related Event 44
Figure 22: Running an operation with caldera and collecting evidence (1) 44
Figure 23: Running an operation with caldera and collecting evidence (2) 45
Figure 24: Running an operation with caldera and collecting evidence (3) 45
Figure 25: Running an operation with caldera and collecting evidence (4) 46
Figure 26: Wazuh Specific Powerview Script Trace Log 47
Figure 27: Wazuh Specific Mimicatz Script Trace Log 48

List of Tables

Table 1: Adherence to CYberSynchrony’s GA 8
Table 2: Links to other Project Outputs/Work 9
Table 3: Kitsune Monitoring alignment with CYberSynchrony objectives. 14
Table 4: Kitsune Monitoring technology stack. 15
Table 5: CSPA alignment with CYberSynchrony objectives. 17
Table 6: CSPA Technologies 19
Table 7: ASA alignment with CYberSynchrony objectives. 20
Table 8: ASA Technologies 22
Table 9: PTA alignment with CYberSynchrony objectives. 23
Table 10: PTA technology stack. 24
Table 11: Posture Assessment Control Mapping Table 33

© CYberSynchrony, 2024

Acronyms and abbreviations

Abbreviation / Term	Description
API	Application Programming Interface
APT	Advanced Persistent Threat
ATT	Adversarial Tactics, Techniques
CRUD	Create, Read, Update, Delete
CSV	Comma-Separated Values
CTI	Cyber Threat Intelligence
CVE	Common Vulnerabilities and Exposures
DNS	Domain Name System
DPM	Data Preparation Module
GDPR	General Data Protection Regulation
HEUR	Heuristic
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
JSON	JavaScript Object Notation
LAMP	Linux, Apache, MySQL, PHP (web stack)
LDAP	Lightweight Directory Access Protocol
LEMP	Linux, Nginx (Engine-X), MySQL, PHP (web stack)
LLC	Limited Liability Company
MFA	Multi-Factor Authentication
MISP	Malware Information Sharing Platform
MITRE	MITRE Corporation
PHP	Hypertext Preprocessor
RAW	Raw data format
RBAC	Role-Based Access Control
REST	Representational State Transfer
SAML	Security Assertion Markup Language
SDK	Software Development Kit
SHA	Secure Hash Algorithm

© CYberSynchrony, 2024

SOC	Security Operations Center
STIX	Structured Threat Information Expression
TLP	Traffic Light Protocol
UFT	Unalterable Forensic Techniques
WHOIS	Domain Registry Lookup Protocol
XML	eXtensible Markup Language
YAML	YAML Ain't Markup Language

© CYberSynchrony, 2024

Executive Summary

Deliverable D3.2, titled "*Risk Scenario Analysis Toolkit v1*", falls under WP3 of the CYberSynchrony Project.

The primary goal of this deliverable is to describe a software suite that offers a range of tools designed for the assessment and comprehension of specific cyber risk scenarios. Specifically, this deliverable aims to enhance real-time cyber threat evaluation capabilities and to deliver specialized risk scenario analyses, potentially adaptable to the needs of Member States.

The overarching objective is to equip stakeholders with effective instruments to analyse and understand the various dimensions of cyber threats, thereby enabling improved preparedness and incident response. It is essentially a cornerstone of the project, designed to empower stakeholders with advanced tools for comprehensive cyber risk scenario assessment.

This toolkit integrates four key standalone components, *Kitsune Monitoring*, *Cybersecurity Posture Assess*ment (CSPA), *Attack Surface Assessment* (ASA), and *Purple Team Assessment* (PTA) to provide a holistic, evidence-based approach to cyber threat evaluation and incident preparedness.

- *Kitsune Monitoring* offers proactive surveillance for early detection of threats across open and clandestine sources.
- The *CSPA* delivers a control-mapped, real-world assessment of organizational security maturity, grounded in operational data and aligned with industry standards.
- *ASA* automates endpoint discovery and vulnerability analysis, enabling rapid identification and remediation of risks.
- *PTA* bridges offensive and defensive operations, simulating adversary behaviours to validate and enhance detection and response capabilities.

Together, these components form a modular, adaptable suite that not only strengthens risk monitoring and response but also aligns with the broader objectives of CYberSynchrony: harmonizing people, processes, and technology for robust, scalable, and collaborative cybersecurity across Member States.

Building upon the foundational capabilities presented, D3.4, titled "*Risk Scenario Analysis Toolkit v2*" will provide an enhanced and extended version of the D3.2; this updated release will incorporate refinements based on stakeholder feedback, operational testing, and evolving threat landscapes.

© CYberSynchrony, 2024

1 Introduction

This section presents the CYberSynchrony's Grant Agreement (GA) commitments, as extracted from the formal deliverable and task description, with respect to their outputs and work to be performed.

This deliverable (D3.2) is closely interconnected with several foundational and operational deliverables across the CYberSynchrony project:

- *D2.1 (System Requirements and Architecture, WP2)*: D3.2 components are specified and implemented in accordance with the overall system architecture and requirements defined in D2.1.
- *D3.1 (Toolkit Specification, WP3)*: D3.2 builds directly on the specifications, requirements, and use cases established in D3.1, translating high-level designs into operational capability.
- *D4.1 (Integration & Validation, WP4)*: The functionalities, APIs, and operational outputs described in D3.2 are tested and validated per the criteria and procedures set out in D4.1, ensuring seamless integration into the overall platform.
- *D8.1 (Risk and Exploitation Plan, WP8)*: D3.2 delivers actionable threat, vulnerability, and risk data to D8.1, supporting ongoing project risk management and the exploitation strategy for CYberSynchrony outcomes.

1.1 Mapping CYberSynchrony Outputs

Table 1 below provides a mapping of D3.2 sections to related deliverables and describes the specific reference or dependency for each, ensuring project-wide alignment and awareness of cross-WP interactions.

Table 1: Adherence to CYberSynchrony's GA

Component no	Component title	Respective Document Chapter(s)	Justification
DELIVERABLE			
D3.2	Risk Scenario Analysis Toolkit v1	Chapter 1,2,7	Defining a software suite offering tools for evaluating and understanding specific cyber risk scenarios.
		Chapter 3 (Kitsune Monitoring), Annex I	In the context of WP3, Kitsune Monitoring component contributes to the strengthening of risk monitoring services and the early identification of cyber incidents, offering specialized risk scenario analysis tailored for MS.
		Chapter 4 (CSPA), Annex II	The CSPA service contributes directly to project's objective of supporting customized, operational cyber risk assessments for MS.
		Chapter 5 (ASA), Annex III	The ASA component is closely aligned with the overarching objectives of the project, as it provides a proactive, context-aware evaluation of cyber risks tailored to each MS. By simulating real adversary behaviours, customizing attack vectors to reflect national infrastructure

© CYberSynchrony, 2024

			and threat profiles, and identifying exploitable attack chains, ASA offers a realistic and operationally relevant stress test of existing defences.
		Chapter 6 (PTA), Annex IV	PTA service by simulating realistic attack scenarios and assessing how well current security measures detect and respond to them, helps identify vulnerabilities and validate the effectiveness of detection mechanisms. This dual focus on exploiting weaknesses and strengthening response capabilities aligns with WP3's aim to enhance real-time threat evaluation and improve incident preparedness, ultimately empowering organizations and MS with more resilient and adaptive cybersecurity postures.
TASKS			
T3.3	Crafting Personalized Risk Assessment Tools	Chapter 3,4,5,6, Annex I, II, III, IV	This task aims to develop a versatile software framework tailored to the diverse cybersecurity landscapes of MS. The framework (Chapters 3-6) excels in creating customized risk assessment metrics based on each state's specific threats and vulnerabilities.
T3.4	Dynamic Risk Parameter Refinement	Chapter 4,5,6, Annex II, III, IV	By continuously feeding real-time threat data into its risk parameter engine, the framework ensures that its threat models are always in sync with the most current cyber threat landscapes (Chapters 4-6). This relentless adaptation equips MS with risk profiles that remain perennially relevant, thereby enabling them to remain agile in the face of an ever-transforming cyber environment.

1.2 Linkage to other project outputs

The table below explains how this deliverable is related to other project outputs and specific deliverables.

Table 2: Links to other Project Outputs/Work

WP Number	Task Number	Deliverable Number related	Content

© CYberSynchrony, 2024

WP2	T,2.1, T2.3, T2.4	D2.1	Deliverable D2.1 serves as a foundational element for the overall architecture of the CYberSynchrony ecosystem, making it highly relevant to D3.2.
WP3	T.3.4	D3.1	For D3.2 to effectively perform its risk analysis, its specifications, such as data input formats, expected types of threat indicators, and analytical capabilities, must be aligned with and informed by the outputs and capabilities of D3.1.
WP4	T4.3, T4.4	D4.1	D4.1 provides the crucial monitoring capabilities necessary to assess D3.2's performance and functional integration within the broader CYberSynchrony ecosystem.
WP8	T8.1	D8.1	D8.1 serves as the foundational document detailing the integration roadmap for all CYberSynchrony modules, including components from Work Package 3 (WP3) such as D3.2.

1.3 Deliverable Structure

This document is divided into the following chapters:

- **Chapter 1: Introduction**

  This chapter is an overview of Deliverable 3.2. It introduces us to the Document's objectives and its contribution to the overall project while it outlines its alignment with it.

- **Chapter 2: Position in CYberSynchrony Architecture**

  This chapter starts by presenting an overall architecture diagram of the project and sub-diagrams of the components involved in this Deliverable while highlighting the relationships between them. Moreover, it goes deeper by analyzing the position as well as the function of the tools/components of the Deliverable and their interaction with the other elements in the CYberSynchrony Ecosystem.

- **Chapter 3: Kitsune Monitoring**

  This chapter focuses on one of the four tools defining the Toolkit, Kitsune. It highlights the component's purpose and how it aligns with the objective of the Project and WP. In addition to this, it provides highlights on tool's architecture, functionalities, inner workings and its advantages over existing solutions while showcasing its programming languages, frameworks and libraries.

- **Chapter 4: Cybersecurity Posture Assessment (CSPA)**

  This chapter goes deep on analysing Cybersecurity Posture Assessment, the second tool to be used for Risk Scenario Analysis in the Ecosystem of CYberSynchrony. It provides useful information regarding the tool's purpose and its alignment with the CYberSynchrony Project while bringing knowledge on component's architecture and functionalities as well as on its technology.

- **Chapter 5: Attack Surface Assessment (ASA)**

  This chapter presents Attack Surface Assessment as one of the four components of the Risk Scenario Analysis Toolkit. It focuses on tool's purpose and its alignment with the objective of the Project and WP. It goes

© CYberSynchrony, 2024

further than this, describing tool's architecture, functionalities, inner workings and its advantages over existing solutions, offering at the same time highlights on its programming languages, frameworks and libraries.

- **Chapter 6: Purple Team Assessment (PTA)**

This chapter discusses Purple Team Assessment as the last of the four tools presented in this Deliverable. As in the previous chapters, the main focus is on the purpose of the component and its alignment with the objective of the Project and the WP. It includes highlights on tool's architecture, functionalities, inner workings and its advantages over existing solutions. At the same time, it underlines the programming languages, frameworks and libraries used.

- **Chapter 7: Conclusion**

This concluding chapter summarizes the output of Risk Scenario Analysis Toolkit while analyzing its four components, their role in defining the Risk Scenario Analysis Toolkit and their contribution to the whole CYberSynchrony Ecosystem.

© CYberSynchrony, 2024

2 Position in CYberSynchrony Architecture

This section presents the positioning and interaction of the standalone components described in this deliverable within the overall CYberSynchrony architecture.

The Risk Scenario Analysis Toolkit v1 is a suite that directly supports the project's mission to harmonise people, processes, and technology for robust, scalable cybersecurity and risk management.

The Toolkit is composed of four standalone components

- Kitsune Monitoring
- Cybersecurity Posture Assessment (CSPA)
- Attack Surface Assessment (ASA)
- Purple Team Assessment (PTA)

each designed to address a specific layer of cyber risk identification, assessment, and mitigation.

Within the overall architecture, these components function both autonomously and collaboratively, interfacing with the broader CYberSynchrony ecosystem through well-defined APIs and data flows.

- *Kitsune Monitoring* acts as an external intelligence module, continuously gathering and analyzing threat data from diverse sources, including the dark web, to provide early warning and situational awareness.
- The *CSPA* aggregates operational data from ASA and PTA, mapping findings against established security controls to deliver an evidence-based assessment of organizational security maturity.
- *ASA* automates asset discovery and vulnerability assessment across endpoints, feeding critical exposure data into both the CSPA and PTA modules.
- *PTA* simulates adversary behaviours and evaluates detection and response capabilities, generating actionable insights that further inform the CSPA.

This tightly integrated design reflects CYberSynchrony's architectural principles of modularity, interoperability, and adaptability, ensuring that each component can evolve independently while contributing to a unified risk management framework.

The Toolkit's position in the architecture enables seamless information exchange, supports rapid incident response, and facilitates cross-border collaboration among Member States, in line with CYberSynchrony's objective of delivering enhanced security, operational efficiency, and regulatory compliance through a comprehensive, adaptive approach to cybersecurity.

The architecture diagram below, visually illustrates these relationships, highlighting the flow of data and the interaction points between the Toolkit's components and the broader CYberSynchrony platform, thereby underscoring its pivotal role in enabling proactive, evidence-driven cyber risk management across diverse organizational environments.

The figure highlights the relevant components associated with D3.2: **Kitsune Monitor**, **PTA, ASA** and **CSPA**:

© CYberSynchrony, 2024

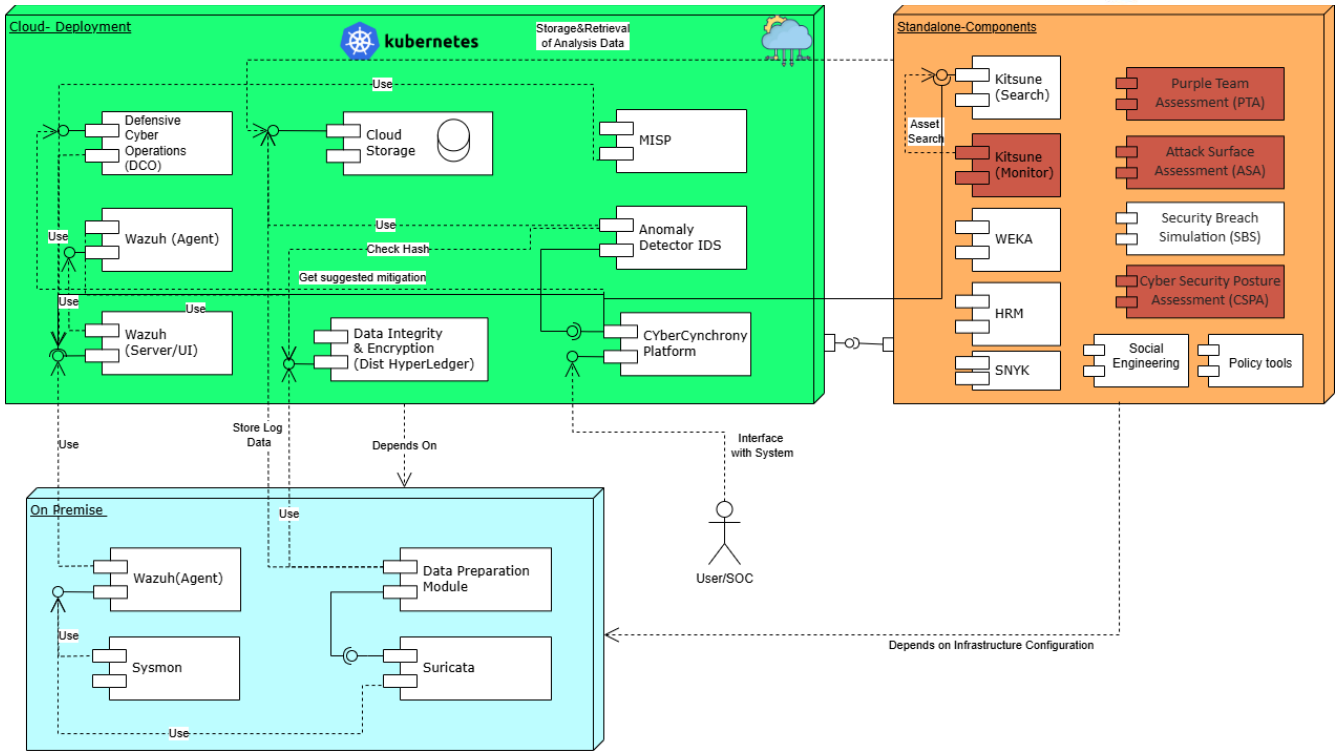

Figure 1: Risk Scenario Analysis toolkit components and architecture overview

© CYberSynchrony, 2024

3 Kitsune Monitoring

3.1 Component Description

The "*Kitsune Monitoring*" component provides a proactive surveillance service for the identification of potential cyber threats and security incidents.

Following an initial configuration involving specific monitoring parameters, the service initiates a process of continuous and automated analysis of various information sources.

When an event matching the defined parameters is detected, the system generates an alert, providing evidence of the suspicious activity. In general terms, the standalone component is designed to detect compromises of user systems, exposures of sensitive data, and relevant mentions in hidden environments.

The data provided by the "Kitsune Monitoring" component can be instrumental in describing the threat landscape of Member States (MS) by offering specific insights into prevailing cyber threats.

More specifically, this type of service is capable of:

- Identifying anomalous activities that could indicate botnet infections affecting monitored users.
- Detecting new unauthorized disclosures of data belonging to specific email domains.
- Tracking the appearance of predefined keywords within the dark web and darknet, in order to intercept potential threats or information leaks.

The innovation lies in the use of the tool for proactive risk detection and a timely response to emerging incidents, minimizing the impact of potential cyber threats.

"Kitsune Monitoring", as part of CYBRITE, contributes to the strengthening of risk monitoring services and the early identification of cyber incidents.

Table 3: Kitsune Monitoring alignment with CYberSynchrony objectives.

Project Objective	Justification
Obj 1: Enhance real-time cyberthreat assessment capabilities.	When an event matching the defined parameters is detected, “Kitsune Monitoring” generates an alert, providing evidence of the suspicious activity. It is designed to detect compromises of user systems, exposures of sensitive data, and relevant mentions in hidden environments.
Obj 4: Provide insights into potential cyber risks faced by entities.	When an event matching the defined parameters is detected, “Kitsune Monitoring” generates an alert, providing evidence of the suspicious activity. It contributes to the strengthening of risk monitoring services and the early identification of cyber incidents.
Obj 2: Offer specialized risk scenario analysis tailored for MS.	By aggregating and analyzing threat intelligence data across a Member State, a more comprehensive and up-to-date picture of the specific cyber threats and vulnerabilities prevalent in that MS can be developed.

© CYberSynchrony, 2024

3.2 Architecture

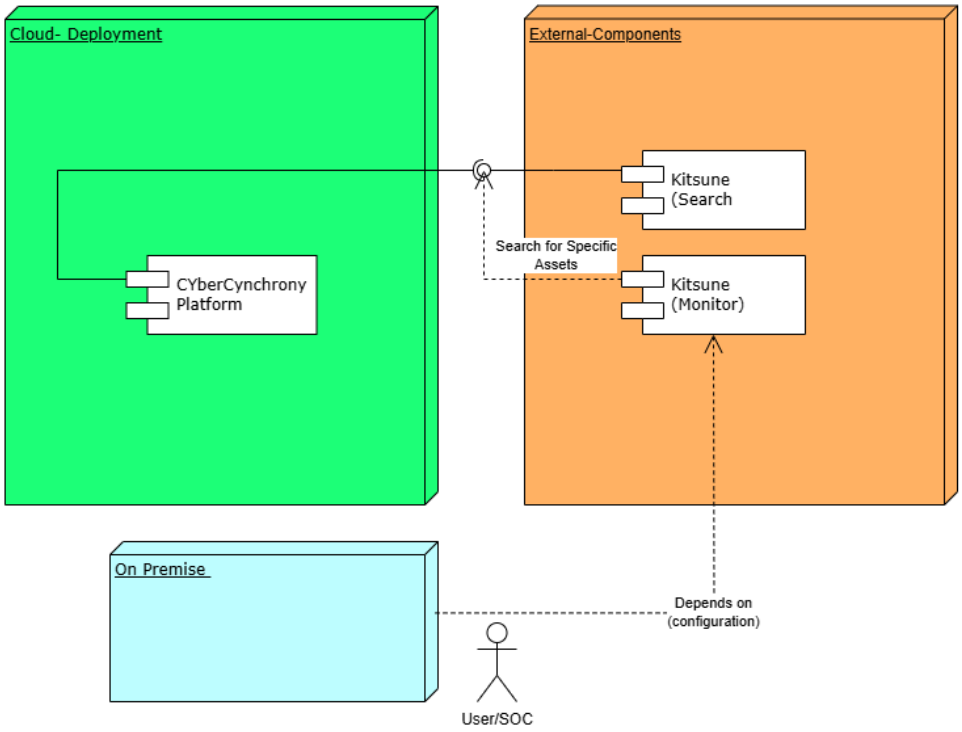

Figure 2: Kitsune Monitoring interactions overview

Kitsune Monitor functions as an autonomous external module, thereby providing the CYberSynchrony platform with advanced monitoring capabilities across a broad spectrum of cyber threat intelligence sources.

This approach facilitates the platform's capacity to investigate issues such as vulnerability exploitation, data leaks, and suspicious activity on the dark web or underground forums involving key internal assets.

3.3 Technologies Used

Kitsune Monitoring is a commercial cybersecurity service under the CYBERRA module. Its primary functionality is to monitor predefined keywords and generate security alerts when specific events occur.

The system detects botnet infections (Infostealer), mentions, data leaks, and references to specific email domains or keywords within the dark web and underground markets; it collects input data in the form of *keyword lists* and produces *security alerts* as output.

The service is classified as a standalone component. Kitsune Monitoring provides a **REST API** that exchanges data in **JSON format,** it does not provide a user interface. The tool is primarily focused on reporting underground mentions, bot exposures, and data leaks, enhancing cybersecurity threat intelligence.

Table 4: Kitsune Monitoring technology stack.

Technology	Purpose
REST	RESTful APIs use standard HTTP methods for CRUD (Create, Read, Update, Delete) operations

© CYberSynchrony, 2024

4 Cybersecurity Posture Assessment (CSPA)

4.1 Component Description

Cybersecurity Posture Assessment (aligned with the SANS Top 20 Critical Security Controls)

FPs **Cybersecurity Posture Assessment** (CSPA) service acts as the reporting and analysis layer for the operational cybersecurity activities. The CSPA service leverages the detailed insights gathered through the **Endpoint Discovery and Vulnerability Assessment (EDVA)** and **Purple Team Assessment (PTA)** services, and then delivers a comprehensive view of an organization's security maturity, mapped against the industry-recognized **SANS Top 20 Critical Security Controls**[1].

The **Endpoint Discovery and Vulnerability Assessment** enables detailed visibility into hardware and software inventories, as well as the identification of known vulnerabilities, directly supporting the mapping of controls such as Inventory and Control of Hardware Assets, Inventory and Control of Software Assets, and Continuous Vulnerability Management. Simultaneously, the **Purple Team Assessment** simulates real-world adversary behaviours and evaluates detection and response capabilities, providing validation against controls including Malware Defences, Application Software Security, Incident Response and Management, and Penetration Testing.

01 Inventory of Hardware
02 Inventory of Software
03 Continuous Vulnerability Management
04 Control of Admin Privileges
05 Secure Configuration
06 Maintenance and Analysis of Logs
07 Email and Browser Protections
08 Malware Defenses
09 Limitations of Ports and protocols
10 Data Recovery
11 Secure Configuration of Network Devices
12 Boundary Defense
13 Data Protection
14 Controlled Access Based on Need to Know
15 Wireless Access Control
16 Account Monitoring and Control
17 Security Awareness Training
18 Application Security
19 Incident Management
20 Penetration Testing

Figure 3: Sans Top 20 Critical Security

By consolidating the operational results of these two services, the CSPA offers a practical, evidence-based mapping of the security environment across the SANS framework. The produced reporting is based on real-world operational data and threat simulation outcomes, ensuring that identified strengths and gaps are based on observed performance rather than assumptions.

[1] https://www.sans.org/blog/cis-controls-v8/

© CYberSynchrony, 2024

At the conclusion of the engagement, the CSPA service delivers a detailed control-by-control mapping, accompanied by evidence (where available), along with prioritized recommendations to strengthen security posture. This integrated approach provides organizations with actionable, measurable, and defensible insights to drive their cybersecurity improvements forward with confidence.

To summarize, the CSPA ingests operational insights from EDVA and PTA components. By correlating threat simulation outcomes with actual endpoint inventories, it builds an evidence-based, dynamic security posture map.

Key functions of the component include:

- **Control Mapping:** Translates findings to specific SANS CSC controls (e.g., Malware Defences, Continuous Vulnerability Management).
- **Evidence-Based Reporting:** Instead of relying on surveys or interviews, CSPA generates posture profiles from **real data** obtained through automated and manual red/blue team activities.
- **Prioritized Recommendations:** Organizations receive a remediation roadmap with actionable security improvements.

This dual-input approach ensures that assessment results reflect **actual performance and capabilities**.

The CSPA enables decision-makers to:

- Identify gaps in compliance with industry frameworks.
- Benchmark security maturity across departments or systems.
- Justify budget allocation for cybersecurity remediation activities.

The CSPA service contributes directly to CYberSynchrony's objective of supporting **customized, operational cyber risk assessments** for MS.

Table 5: CSPA alignment with CYberSynchrony objectives.

Project Objective	Justification
Obj 1: Enhance the threat assessment and risk monitoring services	Aggregates operational insights from real attacks (PTA) and vulnerability scans (EDVA) to provide a real-time picture of posture.
Obj 2: Offer specialized risk scenario analysis tailored for Member States	Adjusts control mappings and assessments per Member State based on actual operational data and national threat landscapes.
Obj 4: Provide insights into potential cyber risks faced by entities	Delivers detailed, control-by-control reporting identifying the most vulnerable systems and highest-risk areas per entity.

4.2 Architecture

The architecture of the Cybersecurity Posture Assessment (CSPA) component is structured around three core modules that work together to provide a comprehensive evaluation of an organization's cybersecurity readiness. The first module, known as the Data Aggregator, is responsible for collecting input data from various sources. It gathers detailed asset inventories and known vulnerabilities at the endpoint level from the Endpoint Discovery and Vulnerability Assessment (EDVA) component. Simultaneously, it retrieves logs from adversary simulation exercises, detection performance metrics, and defensive telemetry generated by the Penetration Testing and Analysis (PTA) component.

© CYberSynchrony, 2024

Once collected, this data is passed to the second module, the Mapping Engine. This engine performs two critical functions. First, it maps the operational findings to the 20 controls outlined in the SANS Critical Security Controls (SANS CSC) framework. This alignment ensures that the assessment is standardized and actionable. Second, it normalizes the raw data into a structured and consistent format, enabling meaningful comparisons across different organizations or environments.

The third module is the Report Generator and Recommender, which synthesizes the processed information into interactive reports. These reports present control-level ratings, such as "Satisfactory", "Partial", or "Fail", based on the mapped findings. Each rating is supported by comprehensive evidence, including system logs, screenshots, and other relevant artifacts. Furthermore, the module provides a set of ranked remediation suggestions tailored to the specific weaknesses identified, ensuring that organizations receive clear guidance on how to improve their cybersecurity posture.

The entire system is powered by input streams from Ansible-based endpoint discovery and CVE (see [1]) scans provided by the EDVA, as well as red team attack simulations and blue team detection logs supplied by the PTA. Together, these feed into the CSPA's assessment engine, which delivers a real-time, dashboard-driven view of an organization's security strengths and areas for improvement.

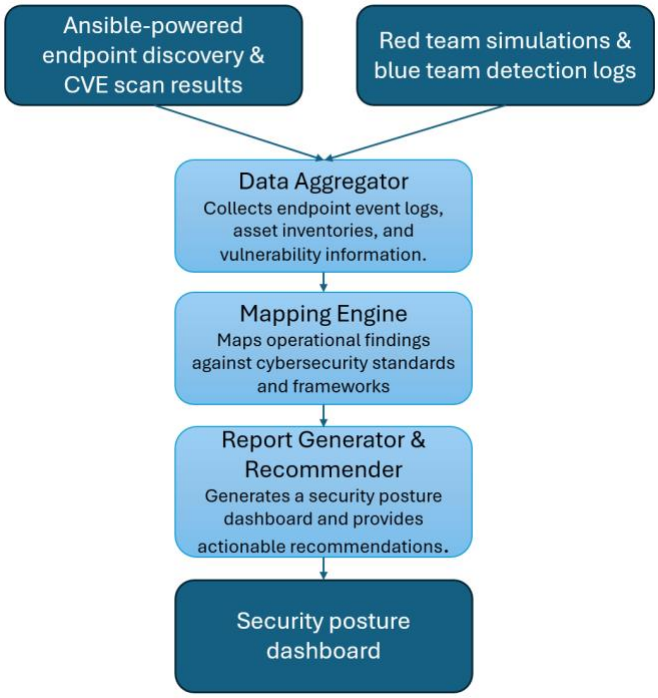

Figure 4: diagram illustrating the architecture of the Cybersecurity Posture Assessment component

4.3 Technologies Used

The Cyber Security Posture Assessment (CSPA) leverages a combination of stable, industry-standard technologies to ensure reliability, security, and interoperability.

These technologies support asset discovery, vulnerability scanning, user behaviour analysis, reporting, and secure data management. Emphasis is placed on proven tools and libraries that are widely adopted in cybersecurity practice and are aligned with EU cybersecurity standards.

© CYberSynchrony, 2024

Table 6: CSPA Technologies

Technology / Stack	Purpose
OpenVAS [2]	Vulnerability scanning across internal and external assets
Nmap	Network discovery and port/service enumeration
Microsoft Defender for Endpoint	Endpoint-level risk monitoring, threat detection, and vulnerability analysis
Microsoft Sentinel	Security data aggregation, incident detection, and analytics
OSQuery	Query-based visibility into endpoint configurations and compliance
Python	Custom logic for posture evaluation and score calculation
Flask / FastAPI	Lightweight web backend for serving dashboards and APIs
Docker	Containerization for safe, isolated deployment of CSPA components
PostgreSQL	Centralized, structured storage of scan data and assessment results

© CYberSynchrony, 2024

5 Attack Surface Assessment (ASA)

5.1 Component Description

This tool automates the discovery and vulnerability assessment of applications, services, and processes running across a fleet of endpoints. By leveraging Ansible, it dynamically deploys platform-specific scripts (PowerShell for Windows and Bash for Linux) to collect system inventory data in standardized CPE (Common Platform Enumeration) URI format.

The collected data is centralized, validated, and cleaned to correct potential inconsistencies, ensuring high-quality input for security analysis. The cleaned CPE list is then queried against the National Vulnerability Database (NVD) API to identify known vulnerabilities (CVEs) associated with the detected software. This process enables organizations to quickly and efficiently assess their endpoint exposure, prioritize remediation efforts, and strengthen their overall security posture.

ASA leverages Ansible to orchestrate platform-specific scripts (e.g., PowerShell for Windows, Bash for Linux), which are deployed to organizational endpoints without requiring dedicated agents. These scripts collect system metadata and generate a normalized Common Platform Enumeration (CPE) list describing applications, processes, and services.

Once collected:

- The system cleans and validates the CPE data to ensure consistency and eliminate duplicate or malformed entries.
- It then queries the National Vulnerability Database (NVD) API to extract CVEs related to the discovered software versions.
- The result is a vulnerability-enriched inventory, giving cybersecurity teams the ability to:
    - Prioritize patching
    - Evaluate risk exposure per system
    - Understand the real-time breadth of their attack surface

This process significantly enhances operational risk awareness and feeds actionable intelligence back into both the Posture Assessment and Purple Team simulation planning.

Table 7: ASA alignment with CYberSynchrony objectives.

Project Objective	Justification
Obj 1: Enhance the threat assessment and risk monitoring services	Provides up-to-date information on vulnerable assets and exposed services within an organization's infrastructure.
Obj 2: Offer specialized risk scenario analysis tailored for Member States	Supports asset-specific CVE tracking and can adapt discovery campaigns to Member State-specific environments.
Obj 4: Provide insights into potential cyber risks faced by entities	Enables continuous visibility over changes in the IT environment and highlights weak points with publicly known exploits.

5.2 Architecture

Architecture image showcasing the inner workings of the tool/component.

© CYberSynchrony, 2024

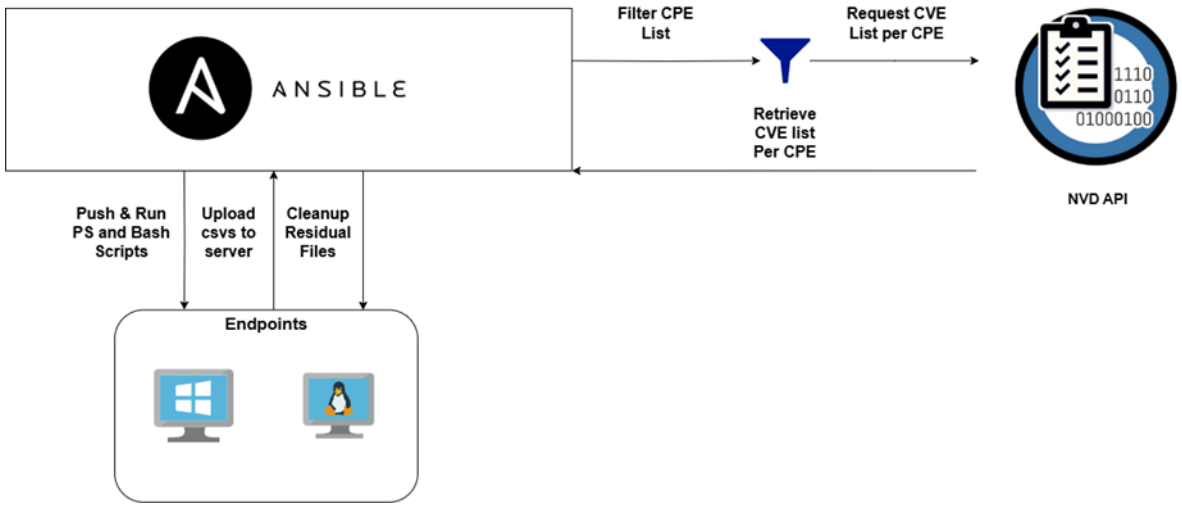

Figure 5: ASA Architecture

The ASA architecture, as illustrated in the diagram, comprises three main components: the Ansible Server, a collection of Endpoints (grouped into Windows and Linux machines), and the external NVD API (National Vulnerability Database), which provides authoritative CVE (Common Vulnerabilities and Exposures) data.

At the heart of the architecture lies the Ansible Server, a centralized orchestration node that operates without agents, making the solution lightweight and highly scalable. This agentless nature, enabled by Ansible, offers a significant operational advantage over traditional asset discovery tools, which often require installation and maintenance of endpoint clients. By leveraging native system utilities on Windows (via PowerShell scripts) and Linux (via Bash scripts), the ASA ensures compatibility across diverse environments without the need for third-party dependencies.

The ASA process begins when the Ansible Server pushes lightweight scripts to designated endpoints. These scripts are designed to inventory running applications, system services, and active processes by capturing their identifying strings in the CPE (Common Platform Enumeration) URI format.

This standardized format is vital for interoperability with vulnerability databases and subsequent automation steps. Once deployed, the endpoints autonomously execute the scripts. During this execution phase, each endpoint collects detailed asset metadata, covering software versions, configurations, and process attributes, and organizes this data into structured CSV files. These CSVs provide an efficient, portable medium for transmitting data back to the Ansible Server, minimizing bandwidth consumption and ensuring traceability.

After the scripts complete execution, a clean-up process is immediately triggered on the endpoints. This step deletes both the scripts and the CSV files from the local machines, ensuring no residual data is left behind that could pose a security risk or consume system resources. This automatic sanitation enhances the operational hygiene of the ASA system. Once the CSVs are successfully received, the Ansible Server validates and cleans the CPE data. This involves checking for typos, inconsistencies, or format deviations that could impede the next step of the workflow. Clean, validated CPE strings are essential for ensuring accurate mapping to known vulnerabilities.

Following this data normalization step, the Ansible Server interfaces with the NVD API, which is a publicly available threat intelligence service maintained by the U.S. National Institute of Standards and Technology (NIST). The server requests a list of CVEs for each unique CPE entry, essentially linking every discovered software component or process with its known security exposures. This query-response cycle allows ASA to build a detailed risk profile for each endpoint, grounded in real-world vulnerability intelligence.

From a cybersecurity operations perspective, this workflow delivers immense value.

© CYberSynchrony, 2024

Firstly, it provides rapid asset discovery and risk exposure mapping, which are foundational elements in any vulnerability management lifecycle. Organizations often struggle to maintain updated software inventories across complex environments, and ASA addresses this gap efficiently. Secondly, the agentless architecture ensures that deployment can be done in highly regulated or restricted environments where agent installation is either prohibited or operationally infeasible.

Furthermore, ASA's integration with real-time vulnerability intelligence via the NVD ensures that results are contextually accurate and continuously relevant. This connection enables security teams to prioritize remediation based on the latest threat landscape rather than relying on static, outdated datasets. By mapping CVEs directly to CPEs collected from live systems, ASA eliminates guesswork and false positives that commonly plague other tools.

Another distinguishing feature of this solution is its extensibility and customization potential. Because it uses open scripting (PowerShell and Bash) and standard CSV formats, security teams can easily tailor the inventory process to include additional data points such as software configurations, patch levels, or system roles. Similarly, the modular architecture allows the integration of supplementary threat intelligence feeds or machine learning-based prioritization engines in future iterations.

In comparison to traditional vulnerability scanners or endpoint management platforms, ASA's state-of-the-art nature lies in its simplicity, interoperability, and automation capabilities. It doesn't depend on proprietary formats or commercial licenses, reducing the cost barrier for adoption. It also scales gracefully in hybrid and multi-cloud environments, as the Ansible engine can manage both on-premises and virtualized endpoints uniformly.

In conclusion, the ASA component provides a modern, efficient, and highly effective method for mapping IT assets to their respective vulnerabilities. By automating the data collection, normalization, and intelligence correlation steps through a robust and agentless architecture, ASA significantly strengthens an organization's ability to monitor and respond to cybersecurity risks in near real-time.

5.3 Technologies Used

List **programming languages, frameworks, and libraries** used in the development.

Table 8: ASA Technologies

Technology	Purpose
Ansible	Agentless orchestration and remote execution of platform-specific scripts
PowerShell (Windows)	Endpoint asset enumeration and process listing on Windows systems
Bash (Linux)	Native inventory collection on Linux systems
Python (pandas, requests, dask)	Data processing, cleaning, and parallel querying of external APIs
CSV	Intermediate file format for transferring data between scripts and aggregator
NVD API	Real-time CVE resolution using Common Platform Enumeration (CPE) references

© CYberSynchrony, 2024

6 Purple Team Assessment (PTA)

6.1 Component Description

Purple Team Assessment service bridges the gap between offensive (Red Team) and defensive (Blue Team) operations to strengthen organization's cybersecurity posture.

In this engagement, the team deploys specialized agents within the environment, including **Caldera** [3] (for red team operations) and **Wazuh** [4] (for blue team monitoring and detection), conducting a combination of **manual attack techniques** and **automated Caldera campaigns** to simulate real-world adversary behaviours against selected targets.

The goal is twofold:

- **Identify effective attack techniques** that successfully bypass existing defences.
- **Validate and enhance detection capabilities**, ensuring that your security tools and teams can accurately detect, respond to, and mitigate threats.

By the end of the assessment, a detailed report will be produced which will highlight:

- Successful attack vectors
- Gaps in detection and response
- Recommendations for improving your defensive measures
- Optional tuning of detection rules based on findings

Purple Team approach not only tests systems but also empowers security team with actionable intelligence to stay ahead of evolving threats.

Table 9: PTA alignment with CYberSynchrony objectives.

Project Objective	Justification
Obj 1: Enhance the threat assessment and risk monitoring services	Simulates real adversary behaviour to stress-test defences in near-operational environments.
Obj 2: Offer specialized risk scenario analysis tailored for Member States	Attack paths, payloads, and detection tests are customized per Member State's infrastructure and threat landscape.
Obj 4: Provide insights into potential cyber risks faced by entities	Identifies attack chains that can bypass current defensive controls and suggests realistic mitigation strategies.

6.2 Architecture

The PTA framework is based on a **modular, agent-assisted architecture**. It combines both automated and manual testing components to simulate end-to-end adversary scenarios while collecting evidence across each phase of the attack lifecycle.

Core Modules:

1. **Caldera Agent** (Red team):
    - Executes predefined ATT&CK techniques and command chains on compromised systems.
    - Generates adversarial telemetry (e.g., privilege escalation, lateral movement).
2. **Wazuh Agent** (Blue team):
    - Monitors system logs, file access, and network anomalies in real-time.

© CYberSynchrony, 2024

- Correlates alerts with detection rules and classifies them by severity.

3. **PTA Orchestrator**:
   - Manages synchronization between attack execution and detection feedback.
   - Records metrics like **attack success rate**, **response latency**, **alert fidelity**.
4. **Scenario Builder**:
   - Allows creation of tailored playbooks per organization or Member State profile.

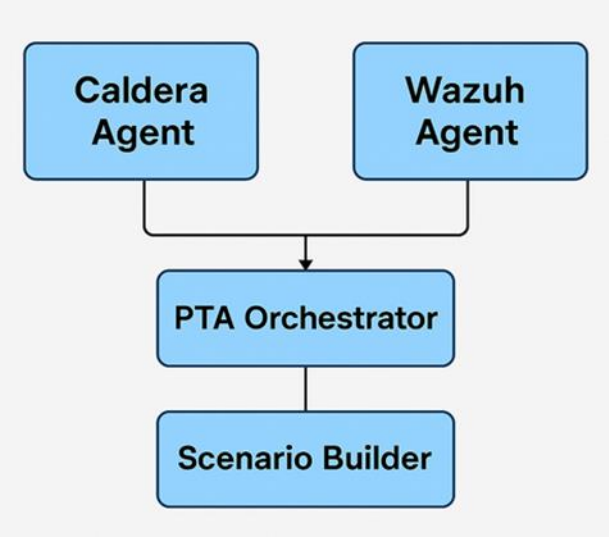

Figure 6: diagram illustrating the architecture of the PTA component

Example Scenario (Executed Campaign):

- **Caldera** launches a credential dumping attack using simulated Mimikatz payload.
- The activity is logged by **Wazuh**, but alert is delayed by 2 minutes.
- Blue Team did not block the attack due to missing detection rule.
- Post-campaign, a new Sigma rule is added to detect lsass.exe access attempts.

Results are forwarded to **CSPA** for posture re-evaluation.

6.3 Technologies Used

PTA emphasizes **realism, automation, and measurability**. It blends advanced attack simulation with fine-grained defensive logging, creating a data-rich environment for evaluating detection quality.

Table 10: PTA technology stack.

Technology	Purpose
Caldera (MITRE)	Emulated Red Team operations and adversary campaigns
Wazuh (SIEM/EDR)	Endpoint monitoring, log aggregation, and alert correlation
PowerShell / Mimikatz	Manual attack testing and privilege escalation (controlled)
Python (custom agents)	Campaign orchestration and evidence collection

© CYberSynchrony, 2024

Kibana / Elasticsearch	Log visualization and exploration of detection telemetry
YAML / JSON	Configuration of Caldera playbooks and Wazuh rules

© CYberSynchrony, 2024

7 Conclusions

The Risk Scenario Analysis Toolkit presented in this deliverable significantly advances the CYberSynchrony ecosystem by providing a unified, operationally driven framework for cyber risk assessment and mitigation.

Each component (Kitsune Monitoring, CSPA, ASA, and PTA) contributes unique value, from real-time threat intelligence and automated asset mapping to dynamic vulnerability assessment and adversary simulation.

This integrated approach ensures that risk analysis is actionable, measurable, and directly tied to organizational objectives and regulatory requirements.

The toolkit not only enhances preparedness and incident response but also fosters a culture of continuous improvement and collaboration among stakeholders, ensuring that the CYberSynchrony platform remains at the forefront of cybersecurity innovation and resilience.

© CYberSynchrony, 2024

8 References

[1] CVE - Common Vulnerabilities and Exposures. [Online]. Available: https://cve.mitre.org/

[2] OpenVAS - Open Vulnerability Assessment System. [Online]. Available: https://openvas.org/

[3] MITRE Caldera. [Online]. Available: https://caldera.mitre.org/

[4] Wazuh - Open Source Security Platform. [Online]. Available: https://wazuh.com/

© CYberSynchrony, 2024

Annex I: *Preliminary PoC and Prototypes (Kitsune Monitoring)*

Kitsune Monitoring is designed to provide proactive surveillance by continuously scanning for cyber threats across a range of sources, including the dark web, open web, and underground forums. The onboarding process is crucial, as it defines the scope and effectiveness of monitoring.

To initiate monitoring, organizations must supply a curated list of keywords, which typically include sensitive domains, email addresses, and custom threat indicators relevant to their operational context. This keyword set forms the basis for all subsequent threat intelligence gathering.

Once the relevant keywords are configured for monitoring, the service will initiate the tracking process and generate an alert upon detecting a specified event. It will monitor for BOTNET infections among users, new data LEAKS affecting specific email domains, and MENTIONS of designated keywords within the dark web and darknet perimeter.

Monitoring Configuration / Onboarding

To configure the Kitsune Monitoring service, it is necessary to provide specific keywords of interest to be monitored, including at least the **main domain** and **email domain**. These keywords enable the system to track relevant events effectively.

As an example of the types of data that can be monitored, you can refer to the following Excel file, which is commonly used during the onboarding process on the platform:

Kitsune CTI Assessment v1.0.xlsx

YAML File (Kitsune Monitoring, Search)

Integration with Kitsune Monitoring is facilitated via a RESTful API, with configuration and interaction details specified in a YAML file. It contains the API structure for utilizing the Kitsune Monitoring (alert) and Kitsune Search services and provides the necessary endpoints, parameters, and request formats required to interact with these services, enabling seamless integration and efficient data monitoring and searching.

kitsune_cybers_swagger.yml

Mock Prototype / Demo

A mock test environment with fictitious data is provided, allowing users to interact and work on API integration.

While this environment is exposed and accessible via the internet, access requires specific IP addresses to be allowed through the external firewall to ensure proper connectivity.

Authentication/Authorization API

Secure access to Kitsune Monitoring is enforced through a **Bearer Token** authentication mechanism.

Process Description:

© CYberSynchrony, 2024

1. **Login Request:**
    - The user submits credentials to the authentication endpoint.
    - Upon successful authentication, the system returns a short-lived access token (valid for 20 minutes).
2. **Token Usage:**
    - All subsequent API calls must include the token in the Authorization header: Authorization: Bearer <token>
3. **Token Refresh:**
    - Before expiry, the token can be refreshed via a dedicated endpoint, ensuring uninterrupted access.

To authenticate, the user must send a request to the authentication endpoint with valid credentials. Upon successful authentication, the system returns an access token, which must be included in the *Authorization header* of subsequent API requests in the format: **Authorization: Bearer <token>**.

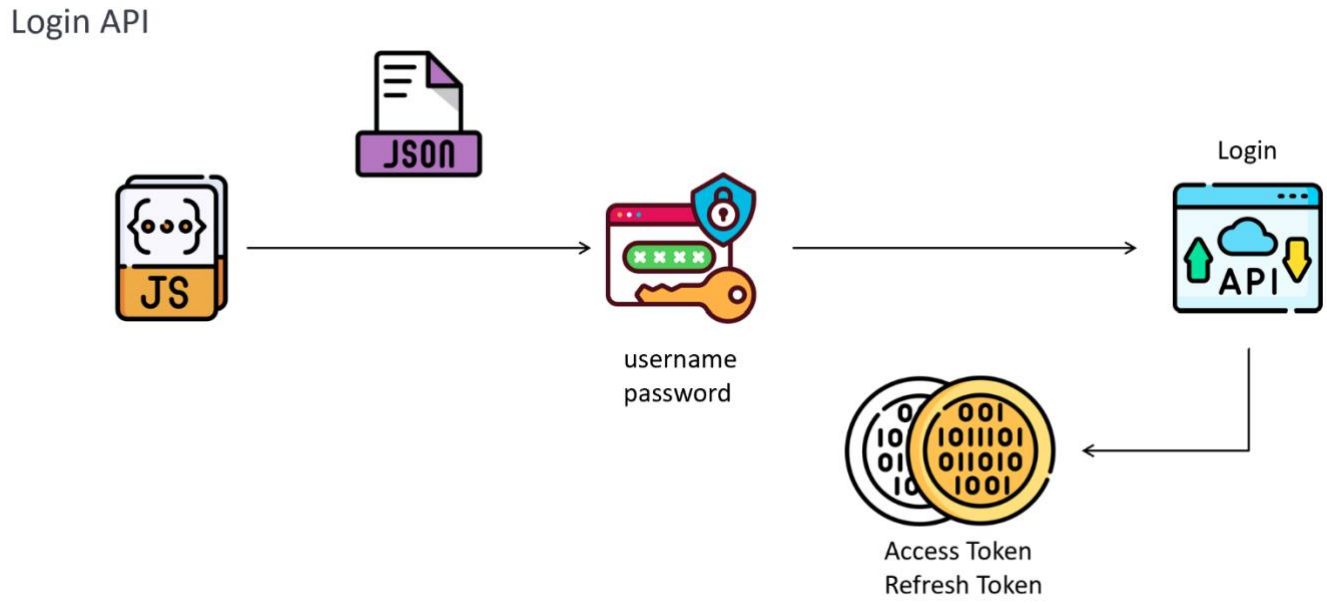

Figure 7: diagram illustrating the API login process

This token grants access to the protected resources for a limited period (**20 minutes**), after which it may need to be refreshed or renewed.

© CYberSynchrony, 2024

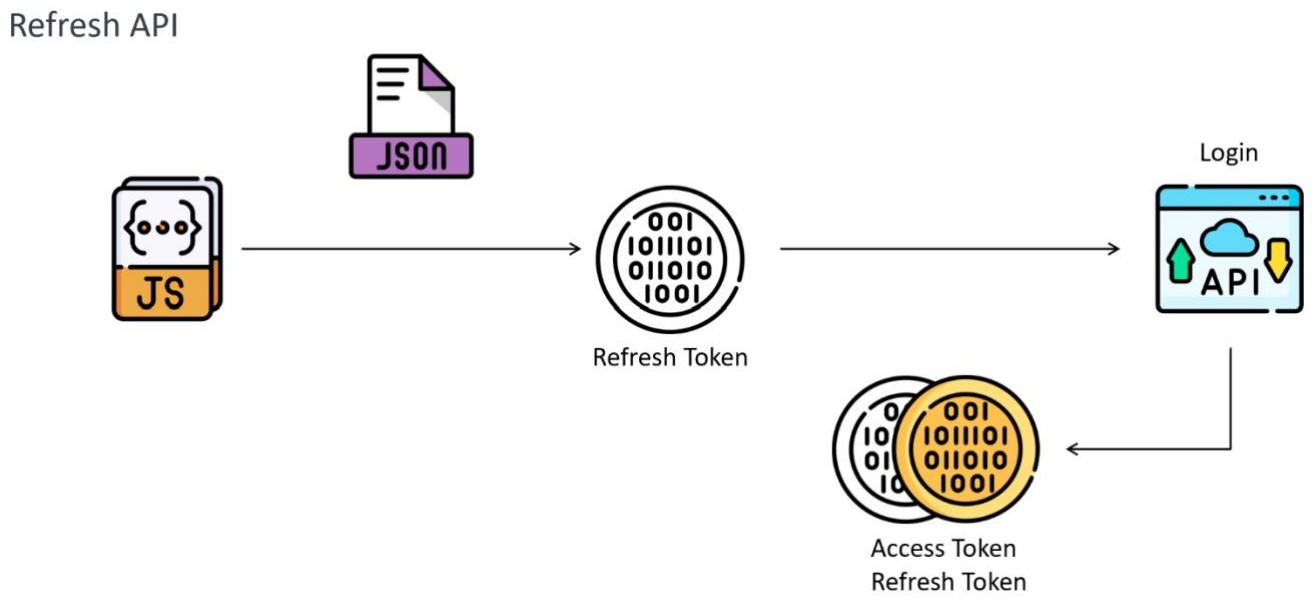

Figure 8: Diagram illustrating the API refresh authorization process

This mechanism ensures that only authorized users can interact with sensitive monitoring resources, reducing the risk of unauthorized access

Monitoring API (Customer Notification)

Once the relevant keywords are configured for monitoring, the service will initiate the tracking process and generate an alert upon detecting a specified event.

The diagram highlights key fields in the alert payload, such as the detected keyword, source of the mention (e.g., specific forum or marketplace), timestamp, and risk classification:

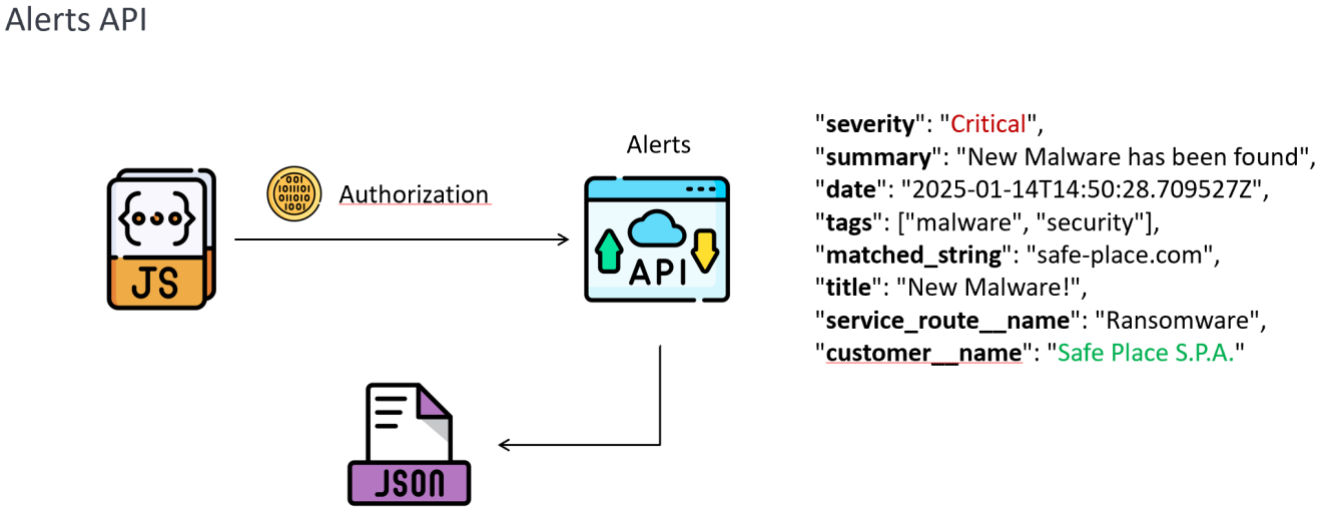

Figure 9: representation of "Alerts" functionality with response

© CYberSynchrony, 2024

This sample demonstrates the structured format of alerts, enabling automated processing and rapid triage by security teams. Each alert includes actionable intelligence, supporting swift investigation and response:

RAW Request/Response sample

RAW Request

```
GET /api/cs/alerts?limit=10&offset=0 HTTP/2
Host: <endpoint-domain>
Accept: application/json, text/plain, */*
Authorization: Bearer b34r3rt0k3n
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

RAW Response

```
HTTP/2 200 OK
<response headers>

{"count":140,"next":"https://<redacted>/api/cs/alerts?limit=10&offset=10"
,"previous":null,"results":[
    {
        "id": 28401,
        "customer_fk": "a0083563-017c-479b-8d3b-379e16213db6",
        "ks_id": "8HSNwn",
        "severity": "HIGH",
        "summary": "Title: news title",
        "date": "2024-08-21",
        "object_id": null,
        "tags": null,
        "additional_informations": null,
        "overview": null,
        "matched_string": null,
        "is_visualized": false,
        "service_route__name": "RSS",
        "service_route__id": 86
    },
    {
```

© CYberSynchrony, 2024

```
            "severity": "HIGH",
            "summary": "description and details",
            "date": "2025-03-30T10:29:25.358633Z",
            "service_route__id": 58,
            "overview": "",
            "customer_fk": " a0083563-017c-479b-8d3b-379e16213db6",
            "ks_id": "8HSNwn ",
            "additional_informations": "",
            "title": "Botnet detected",
            "object_id": "id5a3b02452365e5",
            "tags": [],
            "customer__name": "customer",
            "service_route__name": "Bots",
            "matched_string": "keyword",
            "id": 168,
            "is_visualized": false,
            "status": "PUBLISHED"
        }
]
```


© CYberSynchrony, 2024

Annex II: *Preliminary PoC and Prototypes (Cybersecurity Posture Assessment)*

As part of the PoC for the CSPA component, the following table provides a snapshot of how operational findings from various testing tools are aligned with the **SANS Critical Security Controls (CSC)**. The table helps quantify the organization's cybersecurity posture by presenting observed evidence, associated assessments, and prioritization across selected controls. It demonstrates how evidence-based mapping to SANS CSCs can support a structured approach to identifying coverage gaps, prioritizing remediation efforts, and informing future defensive strategies.

The table outlines seven selected SANS CSCs and maps them against tools used in the PoC, namely **EDVA (Ansible and CVE scans)**, **PTA (via Caldera, Wazuh, and red team activities)**, and **CSPA analysis modules**. For each control, the observed evidence is summarized, a qualitative assessment is made (e.g., "Fail", "Partial Coverage"), and a priority level is assigned

Table 11: Posture Assessment Control Mapping Table

SANS CSC Control	Source Tool	Observed Evidence	Assessment	Priority
CSC 1: Inventory of Hardware Assets	EDVA (Ansible)	Endpoint scan: 312 devices identified; 28 unmanaged or unregistered	Partial Coverage	High
CSC 3: Continuous Vulnerability Mgmt	EDVA (CVE Scan)	97 vulnerabilities detected; 21 critical CVEs unpatched for >60 days	Fail	Critical
CSC 8: Malware Defenses	PTA (Caldera/Wazuh)	Wazuh log showed detection of Meterpreter payload in 3 of 5 simulated attacks	Partial Coverage	Medium
CSC 13: Data Protection	PTA (Simulated attack)	Sensitive documents (classified.pdf) exfiltrated in 2 campaigns without alerting	Fail	High
CSC 18: Penetration Testing	PTA	Manual Red Teaming: achieved domain admin via unpatched Jenkins exposure	Fail	Critical
CSC 19: Incident Response Mgmt	PTA + Wazuh	Log correlation showed detection; no automated response triggered	Partial Coverage	High
CSC 20: Security Skills Assessment	CSPA Analysis	Defensive team missed 2 alerts; Blue team escalation delayed by 45 minutes	Needs Improvement	Medium

© CYberSynchrony, 2024

For example, CSC 1: Inventory of Hardware Assets reveals that although 312 devices were identified, 28 remain unmanaged, resulting in a Partial Coverage rating with High priority.

CSC 3: Continuous Vulnerability Management stands out as Critical, with 97 vulnerabilities detected, including 21 unpatched critical CVEs.

Similarly, CSC 13 (Data Protection) and CSC 18 (Penetration Testing) both receive Fail ratings after red team simulations resulted in successful exfiltration of sensitive data and unauthorized domain access via unpatched Jenkins servers.

Notably, CSC 19: Incident Response logs revealed successful detection of suspicious activity without any automated response, emphasizing a Partial Coverage and underscoring the need for better orchestration.

Finally, CSC 20 (Security Skills Assessment) reflects operational weaknesses in team readiness, as delays in escalation were recorded.

© CYberSynchrony, 2024

Annex III: *Preliminary PoC and Prototypes (Attack Surface Assessment)*

As part of the development of the ASA component, a preliminary PoC has been implemented to validate the core functionality of agentless asset enumeration and vulnerability correlation. The goal of this PoC is to demonstrate the feasibility of using Ansible to automate data collection from Windows and Linux endpoints, structure the output in Common Platform Enumeration (CPE) format, and integrate with public threat intelligence sources, such as the National Vulnerability Database (NVD), for vulnerability mapping.

The following figures provide a detailed walkthrough of this prototype, showcasing how scripts are pushed and executed remotely, how inventories are retrieved and formatted, and how CVE correlation is performed in a reproducible and extensible way.

Figure 10 shows the initial execution of the Ansible playbook (*run_services_enum.yml*) from the control node. The playbook orchestrates a multi-step process targeting three Windows hosts (srv02, dc01, and dc02).

The first tasks involve gathering system facts, ensuring the existence of a temporary folder, and cleaning up any residual output files from previous executions. Ansible then pushes PowerShell scripts to each endpoint, which are designed to enumerate active services, processes, and installed applications in CPE format.

The output indicates successful execution across all nodes, with changes reported for each host as the environment is prepared for fresh data collection. This step confirms that the agentless nature of Ansible works reliably and is capable of handling script-based asset enumeration across heterogeneous endpoints.

Gathering asset inventories through ansible:

```
ansible@netbox:~/ansiki$
ansible@netbox:~/ansiki$ ansible-playbook -i inventory.ini playbooks/run_services_enum.yml

PLAY [Run PowerShell script on Windows hosts] ****************************************************

TASK [Gathering Facts] ***************************************************************************
ok: [srv02]
ok: [dc02]
ok: [dc01]

TASK [Ensure Temp folder exists] *****************************************************************
ok: [srv02]
ok: [dc02]
ok: [dc01]

TASK [Remove previous output file (if any)] ******************************************************
changed: [srv02]
changed: [dc02]
changed: [dc01]

TASK [Copy PowerShell script to remote host] *****************************************************
```

Figure 10: gathering asset inventories through ansible (1)

Figure 11 captures the next phase of the asset enumeration workflow, where the PowerShell scripts execute on each host and export the discovered data into CSV files. The scripts extract detailed metadata on applications and services, convert them into standardized CPE format strings, and log the output.

The Ansible output confirms successful script execution, with a message indicating that the information was exported to CSV format. This ensures that the discovered inventory is structured for easy parsing and subsequent use in vulnerability correlation.

© CYberSynchrony, 2024

The system also sets a timestamp using strftime, which is crucial for traceability and organizing results. Finally, the figure shows that these CSV files are fetched back to the control node. This confirms the successful round-trip collection of endpoint data, a critical prerequisite for vulnerability assessment.

```
TASK [Show PowerShell output] ********************************************************************
ok: [dc01] => {
    "msg": "Applications information exported to CSV\nApplications information CPE-formatted exported to CSV\n"
}
ok: [dc02] => {
    "msg": "Applications information exported to CSV\nApplications information CPE-formatted exported to CSV\n"
}
ok: [srv02] => {
    "msg": "Applications information exported to CSV\nApplications information CPE-formatted exported to CSV\n"
}

TASK [Set a safe timestamp using strftime] *******************************************************
ok: [dc01]
ok: [dc02]
ok: [srv02]

TASK [Fetch output CSV file back to control node with timestamp] *********************************
changed: [srv02]
changed: [dc02]
changed: [dc01]

TASK [Fetch output CSV file back to control node with timestamp] *********************************
changed: [srv02]
changed: [dc02]
changed: [dc01]

PLAY RECAP ***************************************************************************************
dc01                       : ok=10   changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
dc02                       : ok=10   changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
srv02                      : ok=10   changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Figure 11: gathering asset inventories through ansible (2)

In Figure 12, the focus shifts to the enumeration of services, following a similar procedure to the application inventory. The Ansible playbook runs service-specific scripts on the same set of hosts, again generating CPE-formatted outputs and saving them to CSV files.

After execution, each CSV file is retrieved by the control node. The consistency in task success and change count across hosts demonstrates the robustness of the approach. The uniformity in data structure and the clear output logs enhance the credibility of the dataset, making it reliable for automated downstream processing. Importantly, no tasks failed, skipped, or were unreachable, showing high operational reliability for this phase of the PoC.

© CYberSynchrony, 2024

```
TASK [Show PowerShell output] ***************************************************************
ok: [dc01] => {
    "msg": "Services information exported to CSV\n"
}
ok: [dc02] => {
    "msg": "Services information exported to CSV\n"
}
ok: [srv02] => {
    "msg": "Services information exported to CSV\n"
}

TASK [Set a safe timestamp using strftime] **************************************************
ok: [dc01]
ok: [dc02]
ok: [srv02]

TASK [Fetch output CSV file back to control node with timestamp] ****************************
changed: [srv02]
changed: [dc02]
changed: [dc01]

PLAY RECAP **********************************************************************************
dc01                       : ok=8    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
dc02                       : ok=8    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
srv02                      : ok=8    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Figure 12: gathering asset inventories through ansible (3)

Figure 13, demonstrates the third inventory stage, focused on enumerating OS-level hardware and software (referred to as "oshw2" in the output). Again, the process follows the established Ansible routine: script deployment, execution, CSV export, timestamp setting, and data retrieval. The Ansible logs confirm that the information was successfully exported on all nodes.

The final terminal commands (*cd output/oshw2 and ls*) show that the control node now holds CSV files from all three endpoints, named with clear timestamps for reference. This highlights the repeatable and automated nature of the process, ensuring data integrity and version control. It reflects how the ASA framework creates a reliable data pipeline between endpoint inventories and the control node, without persistent agents or manual interaction.

© CYberSynchrony, 2024

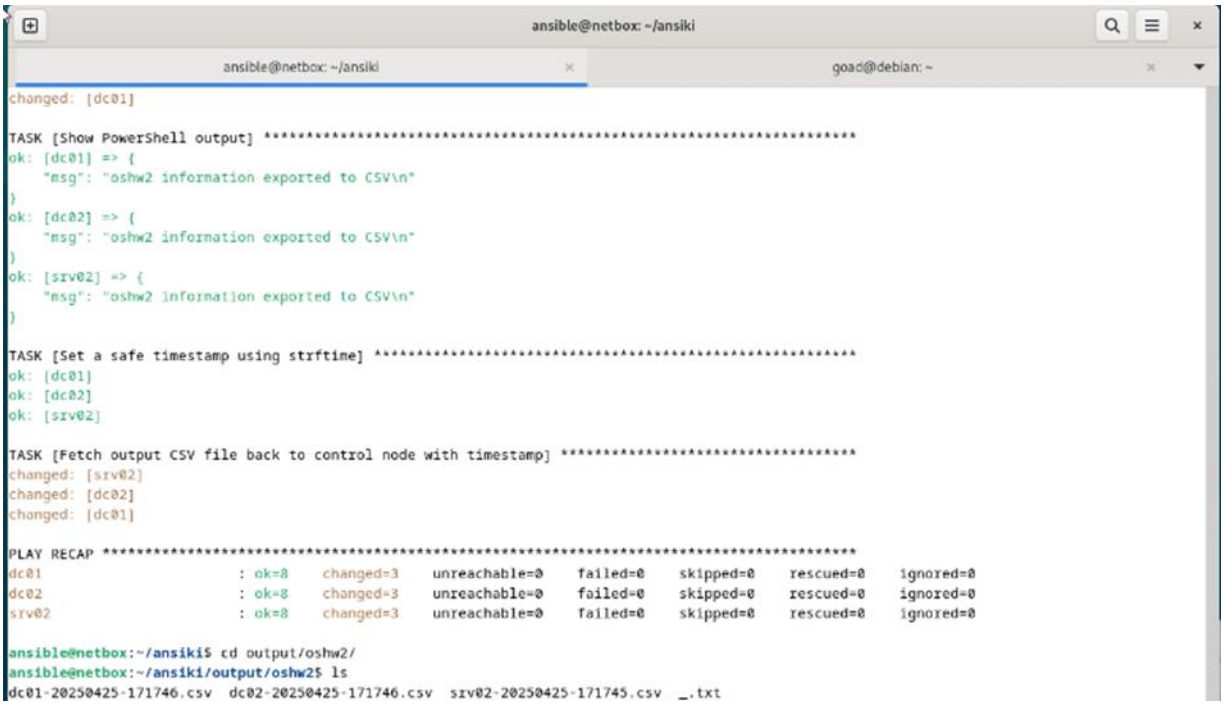

Figure 13: gathering asset inventories through ansible (4)

Finally, the last figure (Fig. 14) presents the CVE mapping phase, where a Python script uses CPE URIs from the CSV files to query the NVD API. Each service or application is checked against the NVD to retrieve its known vulnerabilities. The terminal output shows real-time querying of multiple CPEs such as Apache Tomcat, Google Chrome, Microsoft Office, and Zoom. For each component, the tool prints the associated CVEs. For instance, Apache Tomcat 9 has multiple vulnerabilities listed, including CVE-2023-28709 and CVE-2023-41080. Some entries return no CVEs (e.g., Microsoft IIS), while others result in lookup failures due to malformed or deprecated CPEs. Results are saved in structured CSVs like services_with_cves.csv, making them immediately available for further analysis or visualization. This figure showcases the integration of asset intelligence with threat intelligence, completing the ASA cycle by translating raw inventory into actionable risk insights.

Correlating CVEs

© CYberSynchrony, 2024

```
(base)                \Downloads\official-cpe-dictionary_v2.3.xml\cpetest>python cpecve.py
Checking CVEs for cpe:2.3:a:microsoft:internet_information_services:10.0:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:apache:tomcat:9.0.73:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:apache:http_server:2.4.57:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:google:chrome:124.0.6367.91:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:mozilla:firefox:125.0.3:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:microsoft:edge:124.0.2478.80:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:adobe:acrobat_reader_dc:2024.001.20643:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:microsoft:office:2019:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:microsoft:sql_server:2019:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:oracle:jre:8u391:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:openjdk:openjdk:17.0.9:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:videolan:vlc_media_player:3.0.20:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:7-zip:7-zip:23.01:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:rarlab:winrar:6.24:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:notepad-plus-plus:notepad++:8.6.2:*:*:*:*:*:*:*...
Failed to fetch CVEs for cpe:2.3:a:notepad-plus-plus:notepad++:8.6.2:*:*:*:*:*:*:*: 404
Checking CVEs for cpe:2.3:a:putty:putty:0.80:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:filezilla-project:filezilla:3.66.5:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:microsoft:teams:1.6.00.4472:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:zoom:zoom:5.16.10:*:*:*:*:*:*:*...
Checking CVEs for cpe:2.3:a:slack:slack:4.36.140:*:*:*:*:*:*:*...
Done! Results saved to 'services_with_cves.csv'
Service: Microsoft Internet Information Services (IIS)
CVEs:
  None found

--------------------------------------

Service: Apache Tomcat 9
CVEs:
  - CVE-2023-28709
  - CVE-2023-41080
  - CVE-2023-44487
  - CVE-2023-42794
  - CVE-2023-42795
  - CVE-2023-45648
  - CVE-2023-46589
  - CVE-2024-38286
  - CVE-2025-24813

--------------------------------------

Service: Apache HTTP Server 2.4
CVEs:
  - CVE-2007-4723
  - CVE-2009-0796
  - CVE-2009-2299
  - CVE-2011-1176
  - CVE-2011-2688
  - CVE-2012-3526
  - CVE-2012-4001
```

Figure 14: correlating CVEs

© CYberSynchrony, 2024

Annex IV: *Preliminary PoC and Prototypes (Purple Team Assessment)*

The preliminary PoC implementation of PTA aims to validate its operational design: executing realistic adversarial behavior using tools like Nmap, Hydra, and CALDERA, and monitoring the defensive response using Wazuh as a SIEM and security analytics engine. This testbed allows for measuring detection effectiveness, response latency, and the quality of forensic evidence produced during simulated intrusions. The following figures capture various phases of the PTA lifecycle, from attack generation and detection to correlation with MITRE ATT&CK tactics and techniques.

Example:

This figure below (Fig.15) shows the initiation of a comprehensive Nmap scan from a Kali Linux host against a target machine at IP 192.168.30.20. The scan parameters used (-Pn -p- -sC -sV -oA) indicate that no ping checks were performed, all ports were scanned, default scripts were executed, service versions were detected, and the results were output in multiple formats. This emulates a typical reconnaissance phase of an adversary attempting to fingerprint the target's open services.

The use of full port scanning with scripting enhances realism and aims to trigger meaningful alerts within the target network's detection systems. This setup is foundational for verifying whether intrusion detection systems can recognize noisy scans and appropriately classify them under MITRE ATT&CK reconnaissance techniques.

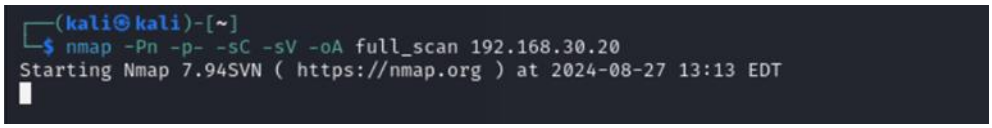

Figure 15: Wazuh Nmap Related Generated Event (1)

In the next phase (Fig.16), the defensive monitoring system (Wazuh) detects and logs a network event generated by the pfSense firewall in response to the Nmap scan. The log shows a block action taken against traffic originating from the scanning host (*192.168.30.141*) to the target (*192.168.30.20*) on port *26103*. The alert includes details like the source and destination IPs, protocol (TCP), and decoded metadata, such as packet flags and TCP options. This demonstrates that the attacker's actions are not only detected but also actively blocked at the firewall level, supporting effective network perimeter defense. Wazuh's visibility into firewall logs enables correlation with offensive behavior and sets the stage for generating higher-level alerts.

© CYberSynchrony, 2024

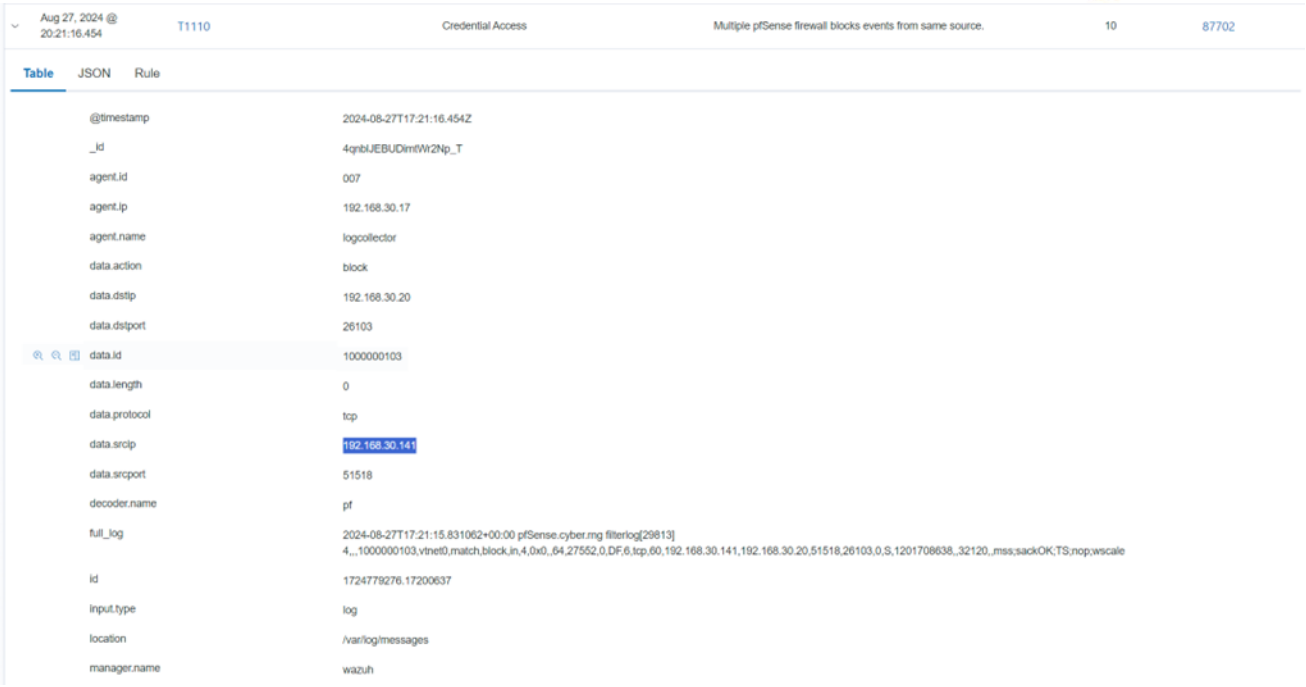

Figure 16: Wazuh Nmap Related Generated Event (2)

Next, Fig.17, illustrates an RDP brute-force attack being executed with Hydra on a remote system (*192.168.4.2*) using a wordlist of passwords. The attacker targets the Administrator account and configures Hydra to limit parallelism to avoid overwhelming the RDP service. The output shows login attempts being processed at a sustained rate, simulating a real brute-force scenario. This attack is representative of the credential access tactic in the MITRE ATT&CK framework.

The intent of this simulation is to test whether the detection system can identify brute-force patterns, classify them accurately, and generate actionable security alerts that indicate potential compromise attempts on critical administrative interfaces.

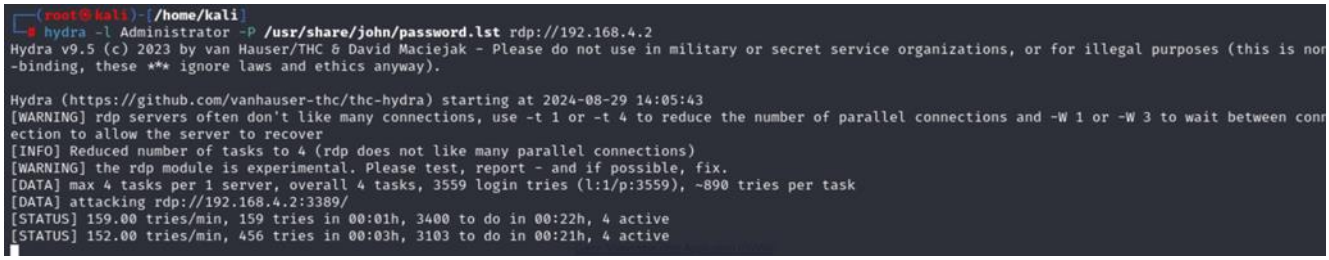

Figure 17: Wazuh Nmap Related Generated Event (3)

In Fig.18 a chronological list of Wazuh-generated alerts triggered during the brute-force attack is depicted. Each alert corresponds to a failed logon attempt, marked by techniques T1078 (Valid Accounts) and T1531 (Account Access Removal), associated with multiple tactics including Initial Access, Persistence, and Impact. These high-fidelity detections validate that Wazuh is capable of recognizing brute-force attack signatures and mapping them to appropriate MITRE ATT&CK entries.

This level of granularity enhances the purple team's ability to measure the fidelity of security telemetry and how well the defensive systems surface meaningful insights to security analysts.

© CYberSynchrony, 2024

Security Alerts

Time ↓	Technique(s)	Tactic(s)	Description
Aug 29, 2024 @ 21:08:50.678	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.
Aug 29, 2024 @ 21:08:49.489	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.
Aug 29, 2024 @ 21:08:49.478	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.
Aug 29, 2024 @ 21:08:49.476	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.
Aug 29, 2024 @ 21:08:48.286	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.
Aug 29, 2024 @ 21:08:48.273	T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	Logon failure - Unknown user or bad password.
Aug 29, 2024 @ 21:08:48.272	T1110	Credential Access	Multiple Windows logon failures.

Figure 18: Wazuh Nmap Related Generated Event (4)

The detailed log of Fig.18, showcases a Wazuh event capturing a failed login attempt to a Windows host. The record includes the logon process name, authentication method (NTLM), status codes, user name (Administrator), computer name, and more. Such detailed forensic evidence is critical for blue teams to conduct root-cause analysis post-incident. It also allows the red team to validate which steps were visible to the defenders, helping refine their techniques. The integration with MITRE ATT&CK (technique T1110) reinforces the contextual interpretation of raw log data and provides a valuable training dataset for detection engineering improvements.

Security Alerts

Time ↓	Technique(s)	Tactic(s)	Description	Level	Rule ID
Aug 29, 2024 @ 21:05:44.038	T1110	Credential Access	Multiple Windows logon failures.	10	60204

Table JSON Rule

@timestamp	2024-08-29T18:05:44.038Z
_id	5alQn5EBUDlmtW2m-0N
agent.id	001
agent.ip	192.168.4.2
agent.name	winadmin
data.win.eventdata.authenticationPackageName	NTLM
data.win.eventdata.failureReason	%%2313
data.win.eventdata.ipAddress	192.168.4.11
data.win.eventdata.ipPort	0
data.win.eventdata.keyLength	0
data.win.eventdata.logonProcessName	NtLmSsp
data.win.eventdata.logonType	3
data.win.eventdata.processId	0x0
data.win.eventdata.status	0xc000006d
data.win.eventdata.subStatus	0xc000006a
data.win.eventdata.subjectLogonId	0x0
data.win.eventdata.subjectUserSid	S-1-0-0
data.win.eventdata.targetUserName	Administrator
data.win.eventdata.targetUserSid	S-1-0-0
data.win.eventdata.workstationName	kali
data.win.system.channel	Security
data.win.system.computer	DESKTOP-5SL2BL9.cyber.rng
data.win.system.eventID	4625
data.win.system.eventRecordID	43932
data.win.system.keywords	0x8010000000000000
data.win.system.level	0
data.win.system.message	"An account failed to log on.

Figure 19: Wazuh Brute Forcing Related Event Detailed (1)

As a next step, Fig.20, captures the use of OWASP DirBuster on the attacking host, targeting the web interface at *192.168.4.2:80*. DirBuster performs recursive directory and file enumeration using common path patterns and extensions.

© CYberSynchrony, 2024

The goal is to identify misconfigured or exposed web resources that could serve as initial access points. This attack falls under the reconnaissance and resource development tactics. The output highlights successful discoveries of paths like */cgi-bin/, /icons/*, and various PHP scripts.

This is a practical simulation of pre-exploitation mapping, and the test helps assess whether such stealthy reconnaissance can be flagged by Wazuh or correlated with suspicious access behavior.

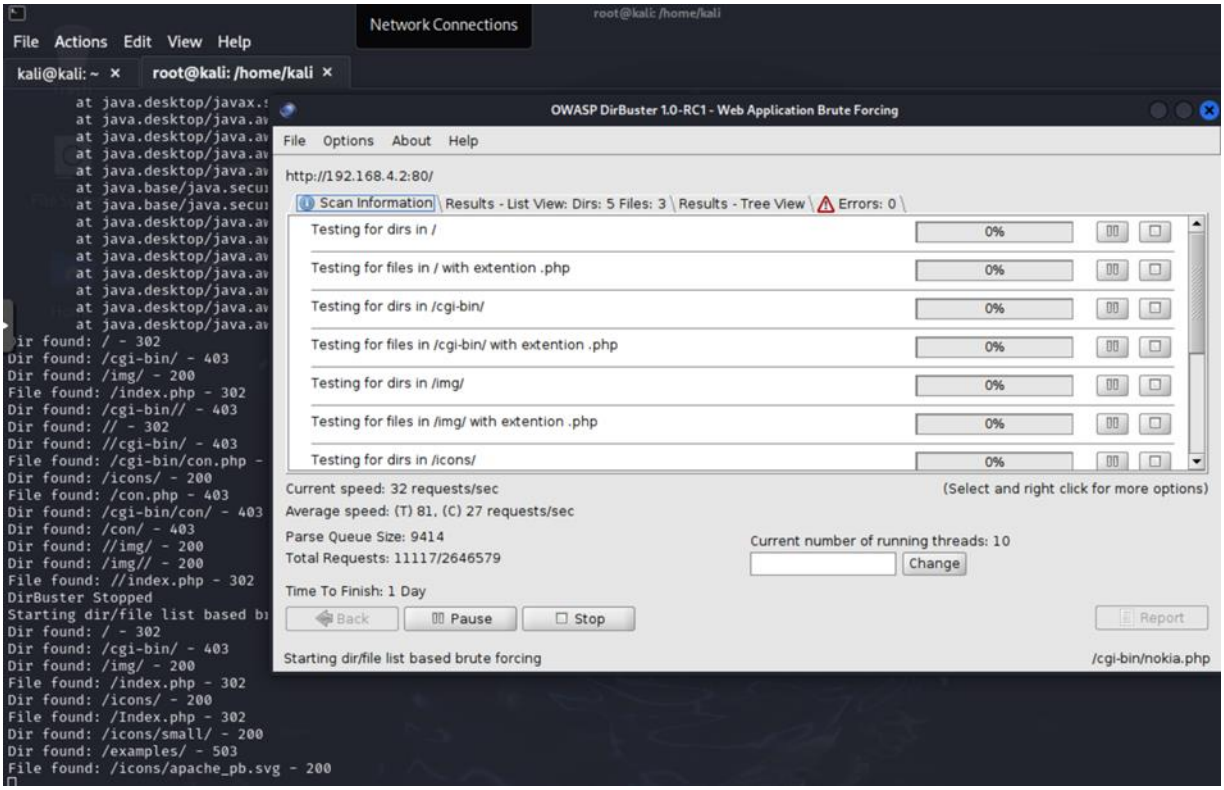

Figure 20: Wazuh Brute Forcing Related Event Detailed (2)

In Fig. 21, Wazuh captures a sequence of pfSense firewall blocks that occurred as a result of the directory brute-force attempts launched via DirBuster. The event logs include TCP metadata and clearly associate the traffic with port 80 on the targeted web server. Wazuh, acting as a central SIEM, consolidates these logs into an actionable alert with high severity, again mapped to technique T1110. This proves that firewall logs, when properly collected and parsed, provide critical insight into reconnaissance behaviors. It also highlights how repeated connection patterns from the same source can trigger automated IP blocking, a defensive control that was successfully validated in this PoC.

© CYberSynchrony, 2024

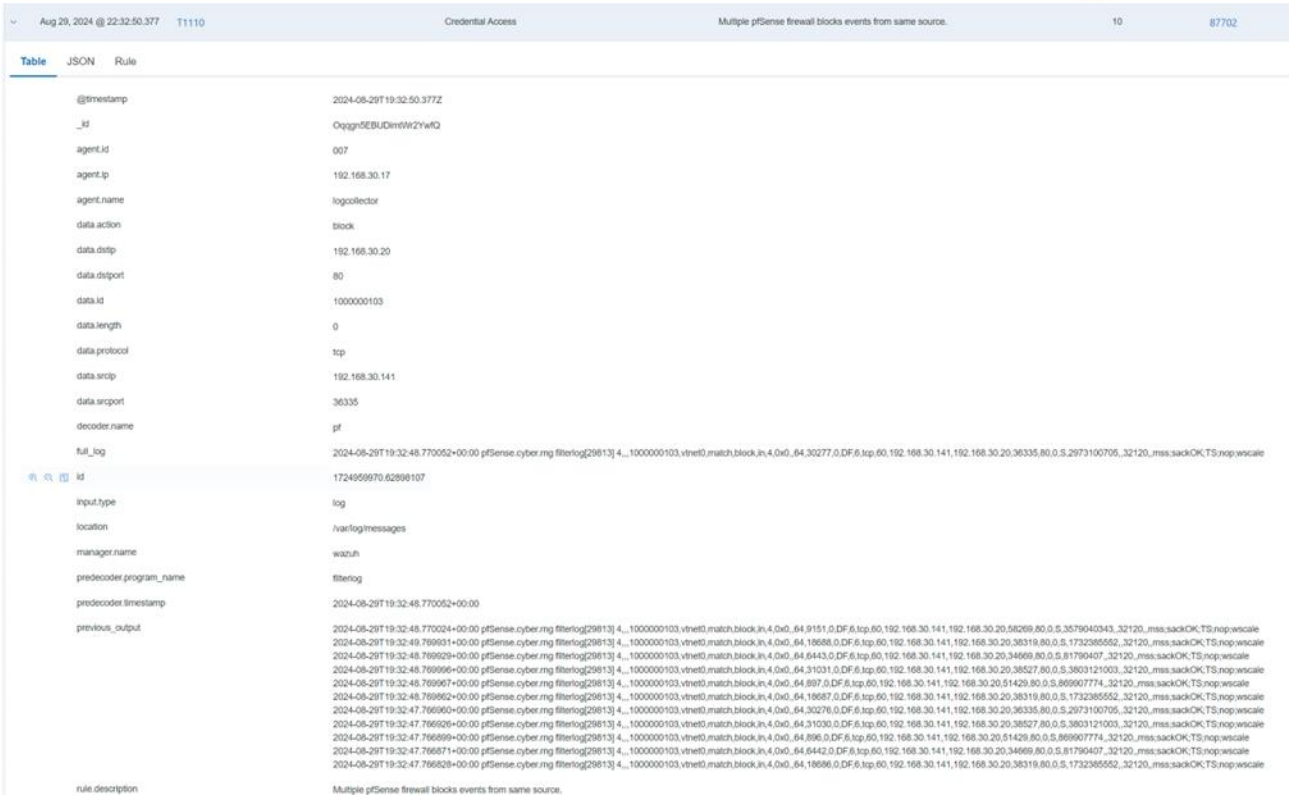

Figure 21: Wazuh Firewall IP Block from Directory bruteforcing Related Event

Running an operation with caldera and collecting evidence

The following figure (Fig.22) shows the start of a CALDERA operation using the adversary profile “Alice 2.0” against a live endpoint.

CALDERA is an automated adversary emulation platform that uses the MITRE ATT&CK framework to plan and execute multi-step attack chains. The operation, named "PenTest1," is launched with a basic fact source and demonstrates the orchestration capability of the PTA component to simulate real-world cyber threat behavior in a controlled and repeatable fashion. This tool enables red teams to validate whether defensive systems like Wazuh can detect behavior-based threats rather than just signature-based alerts.

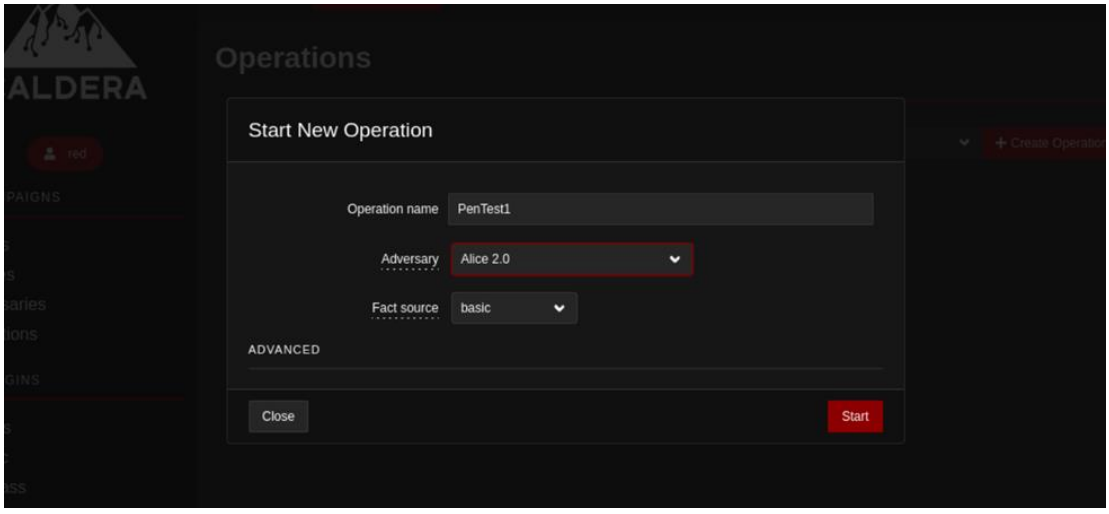

Figure 22: Running an operation with caldera and collecting evidence (1)

© CYberSynchrony, 2024

Next, Fig.23 displays the results of the CALDERA operation, showing a sequence of completed attack steps such as discovering hosts, executing Powerkatz, and retrieving domain information.

Each step has a timestamp and status, and the interface allows viewing the commands used and their output. This granular visibility is essential for evaluating which steps were executed, what evidence was generated, and whether these behaviors were observed by Wazuh. It provides a feedback loop between offensive actions and defensive detections, central to the purple teaming philosophy.

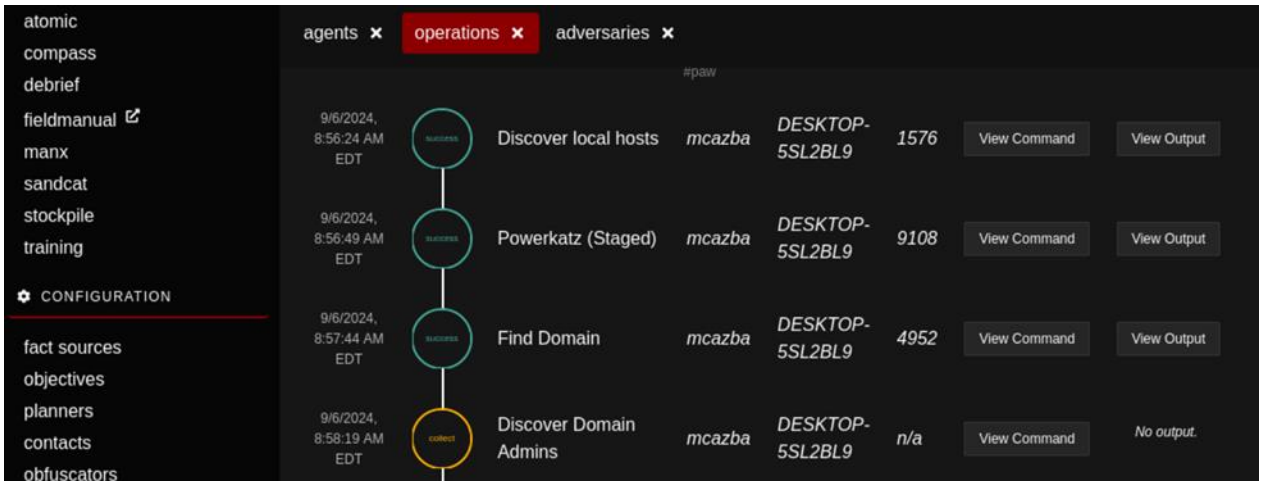

Figure 23: Running an operation with caldera and collecting evidence (2)

The final figure (Fig.24) captures the successful collection of domain credentials by CALDERA using techniques like Mimikatz. The output includes the domain username, NTLM hash, and SHA1 value.

This confirms that the simulated adversary was able to escalate privileges and access sensitive information. Such outcomes help evaluate the criticality of missed detections and provide insight into what defensive controls need to be hardened. When correlated with Wazuh alerts, this evidence becomes a powerful benchmark for improving detection rules, endpoint monitoring strategies, and response protocols.

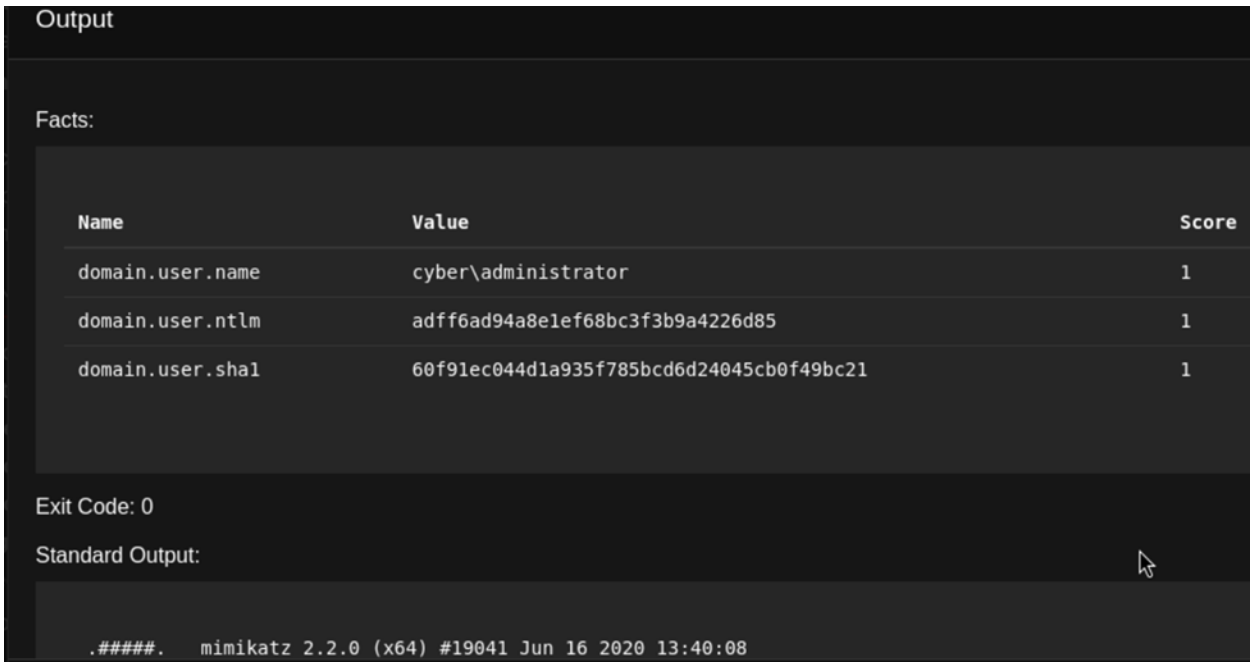

Figure 24: Running an operation with caldera and collecting evidence (3)

© CYberSynchrony, 2024

The following figures (Fig.25, Fig.26, Fig.27) present a timeline of security events captured by Wazuh in response to a CALDERA operation involving execution of post-exploitation tools and scripts.

The logs indicate a sequence of suspicious activities, including the loading of taskschd.dll, which is commonly associated with malware or delayed execution, multiple successful Windows logons, and the dropping of executables into folders that are typically abused by malware (e.g., Users\Public, Temp).

The entries also include process injections via PowerShell and LDAP queries used for remote system discovery. These behavioral traces align with techniques T1059 (Command and Scripting Interpreter), T1218 (Signed Binary Proxy Execution), and T1033 (System Owner/User Discovery).

The breadth of visibility into each granular event proves the strength of Wazuh as a detection engine, reinforcing the Purple Teaming methodology. The logs enable defenders to reconstruct adversary behavior and validate whether detection rules adequately cover lateral movement, privilege escalation, and persistence techniques.

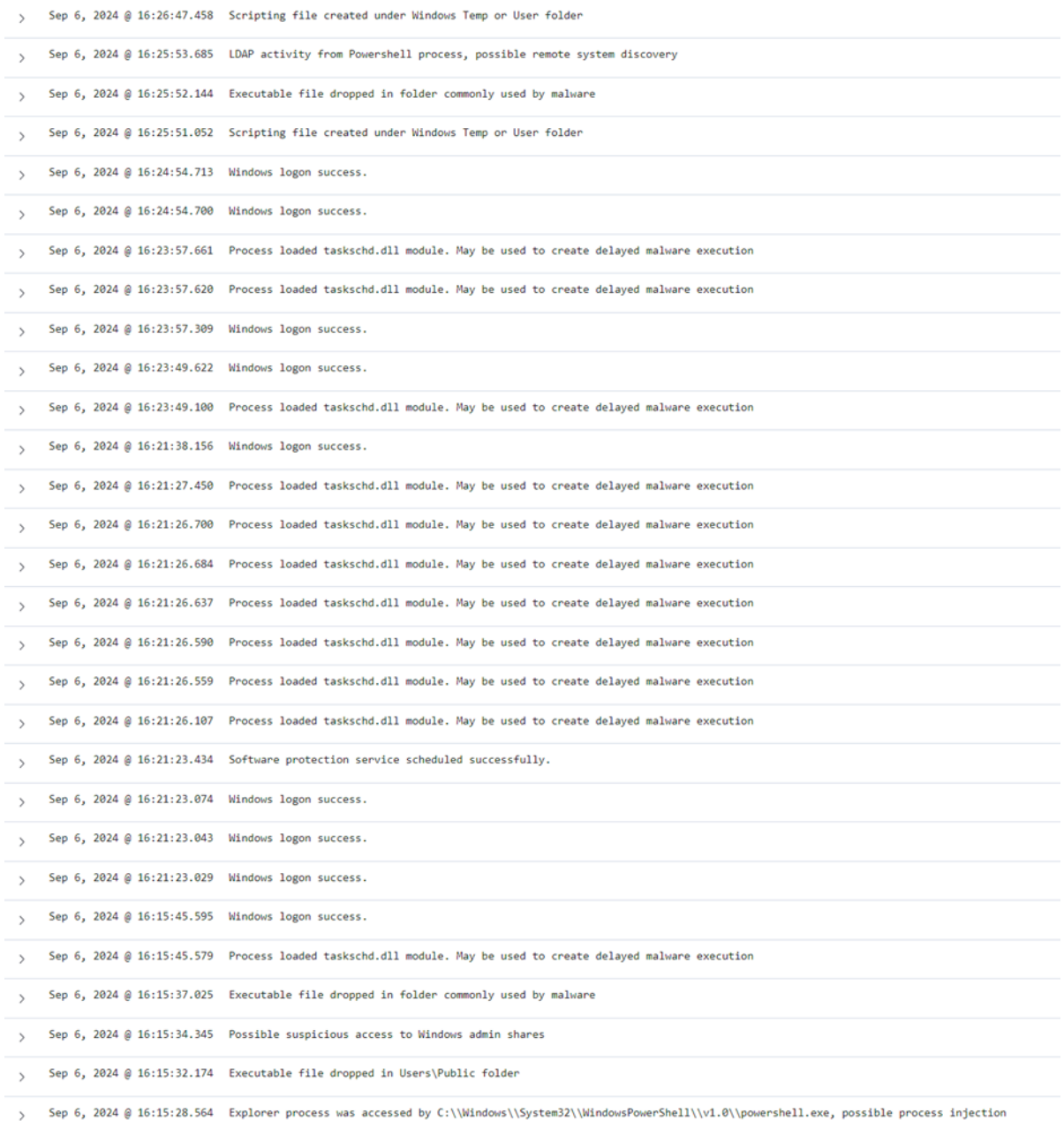

Figure 25: Running an operation with caldera and collecting evidence (4)

© CYberSynchrony, 2024

Fig.26, zooms into a specific Wazuh alert related to the execution of PowerView.ps1, a well-known PowerShell tool used in adversary simulations for Active Directory reconnaissance.

The process that triggered the alert is *splunkd.exe*, a legitimate binary misused here to execute a script located in the Administrator profile directory. The event includes detailed telemetry such as the GUID of the process, process ID, user context (*CYBER\\Administrator*), rule classification (*T1059.001 - PowerShell*), and originating host. This level of detail is crucial for forensic investigation, as it not only reveals the tactic and technique used but also provides insight into how a legitimate tool (*splunkd.exe*) was repurposed for malicious intent, a hallmark of living-off-the-land (LotL) techniques.

The ability of Wazuh to detect such masquerading behavior elevates its value in Purple Team evaluations.

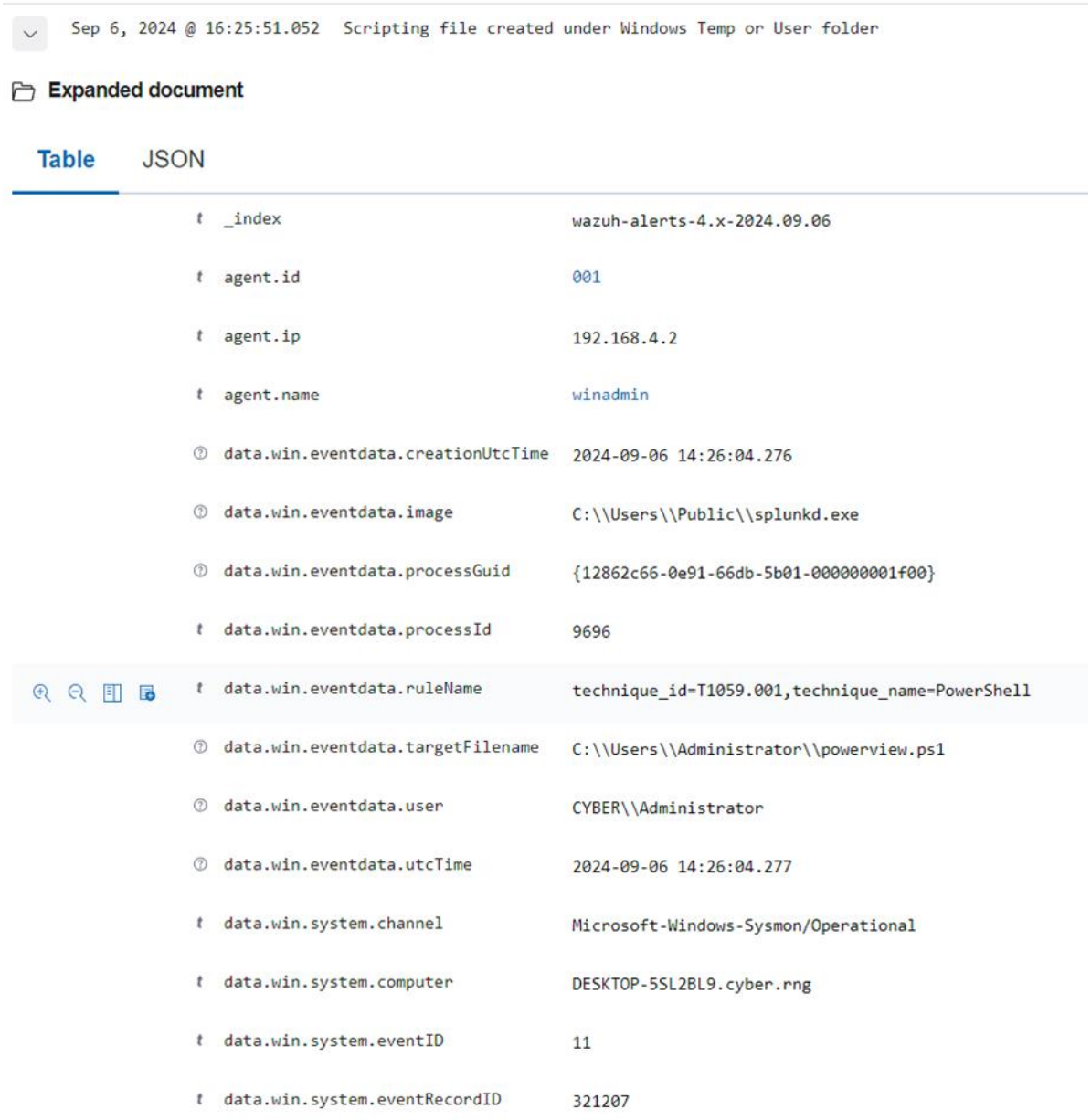

Figure 26: Wazuh Specific Powerview Script Trace Log

© CYberSynchrony, 2024

In the final figure (Fig.27), Wazuh logs another scripting activity executed under the same splunkd.exe process, this time involving the execution of invoke-mimi.ps1, which is a PowerShell-based Mimikatz variant.

The detection metadata matches closely with the previous log, confirming the same process GUID and user context. This behavioral continuity across multiple high-risk script executions demonstrates the adversary's ability to chain post-exploitation actions within a single process space, a technique designed to evade traditional endpoint detection mechanisms.

The alert is also mapped to technique T1059.001, consolidating the classification under PowerShell-based execution. Wazuh's real-time detection of both reconnaissance (PowerView) and credential dumping (Mimikatz) scripts highlights its effectiveness in catching lateral movement and privilege escalation activities commonly used in red team and APT scenarios.

For the Purple Team Assessment component, this evidence further supports the operational effectiveness of the detection stack when faced with multi-stage, fileless adversary behaviors.

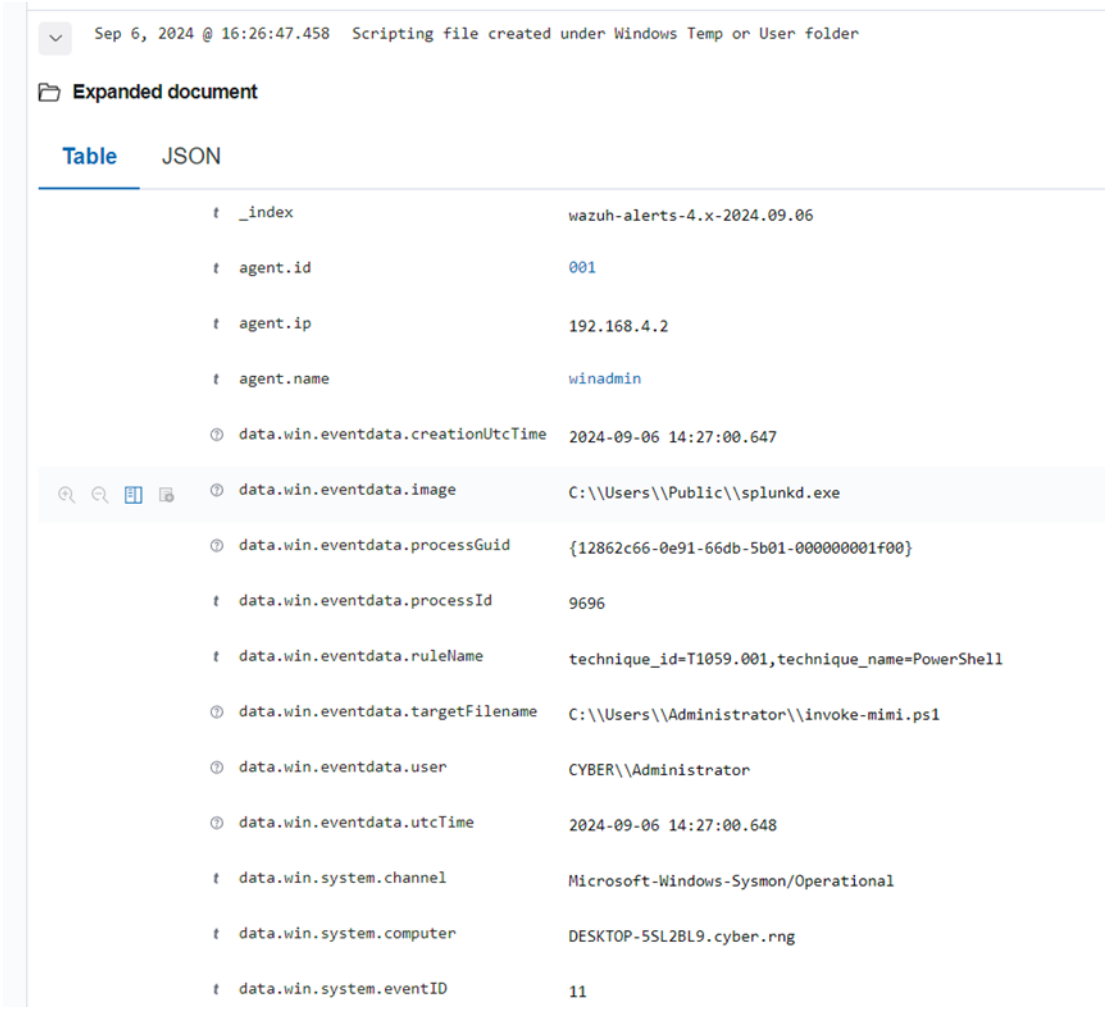

Figure 27: Wazuh Specific Mimicatz Script Trace Log

© CYberSynchrony, 2024