Harmonizing People, Processes, and Technology for Robust Cybersecurity

D4.2 Automated Response 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	D4.2 Automated Response Toolkit v1		
Work Package	WP4 Advanced Proactive Cyber Risk Monitoring and Rapid Response (CYBRITE)		
Contractual due date	31 May 2025	**Actual submission date**	31 May 2025
Type	OTHER	**Dissemination Level**	Public
Lead Beneficiary	Focal Point (FP)		
Responsible Author	Koutras Dimitris (FP), Malamas Vangelis (FP)		
Contributions from	Andrea Purificato(MER), Spyros Papagiannopoulos (UPRC), Michos Evangelos (IANUS), Panayiotis Klitou (EBOS)		

Funded by the European Union

Funded by the European Union under Grant Agreement no. 101158555. 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	Koutras Dimitris (FP)
0.2	02.04.2025	Deliverable Structure Finalization	Malamas Evangelos (FP)
0.3	09.04.2025	First contribution	Koutras Dimitris, Malamas Evangelos (FP)
0.4	20.04.2025	Contribution	Malamas Evangelos, Koutras DImitris (FP)
0.5	05.05.2025	Final draft ready for internal review	Koutras Dimitris, Malamas Evangelos (FP)
0.6	15.05.2025	Peer review	Spyros Papagiannopoulos (UPRC)
0.6	09.05.2025	Peer review	Andrea Purificato (MER)
0.7	22.05.2025	Final draft addressing comments from internal review	Koutras Dimitris, Malamas Evangelos (FP) Michos Evangelos (IANUS)
0.8	29.05.2025	Final quality check	Panayiotis Klitou (EBOS)
1.0	30.05.2025	Final version ready for submission	Koutras Dimitris (FP)

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

© CYberSynchorny, 2024

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 Architecture of the Defensive Cyber Operations (DCO) 11
2.1 Overview of Defensive Cyber Operations (DCO) SIEM & SOAR Integration 11
2.2 Diagram Representation 12
3 Design of Incident Response Playbooks 15
3.1 Playbook Structure & Methodology 15
3.2 Playbook Execution & SOAR Integration 16
3.3 Case Studies – Example Playbook Scenarios 17
3.3.1 Phishing Email Response Playbook 17
3.3.2 Simulated cyber scenarios and game-based Playbook 20
4 Deployment Strategy for Automated Playbooks 23
4.1 Deployment Environment 23
4.2 Testing & Validation of Playbooks 23
5 Conclusions 25
Annex I: Example of DCO implementation 26

List of Figures

Figure 1: System architecture 12
Figure 2: Phishing Playbook 18
Figure 3: Game based scenario steps 20
Figure 4: Game Example 21
Figure 5: Sentinel case_1st step 26
Figure 6: Trigger action_2nd step 26
Figure 7: Disable action_3rd step 27
Figure 8: Extract UPN_4th step 27
Figure 9: Connector in Azure Logic Apps_5th step 28
Figure 10: Permission to the logic app Identity_6th step 29
Figure 11: Run logic app_7th step 30
Figure 12: Final check 30

List of Tables

Table 1: Adherence to CYberSynchrony's GA 8
Table 2: Links to other Project Outputs/Work 9
Table 3: DCO Components 13

© CYberSynchorny, 2024

Table 4: Playbook Components 19
Table 5: Summary of the Phishing Response Playbook 22

© CYberSynchorny, 2024

Acronyms and abbreviations

Abbreviation / Term	Description
AD	Active Directory
API	Application Programming Interface
ATT&CK	Adversarial Tactics, Techniques & Common Knowledge
CI/CD	Continuous Integration / Continuous Deployment
C&C	Command and Control
DCO	Defensive Cyber Operations
DLP	Data Loss Prevention
EDR	Endpoint Detection and Response
FP	Focal Point
IaC	Infrastructure as Code
IDS	Intrusion Detection System
IOC	Indicator of Compromise
JSON	JavaScript Object Notation
KQL	Kusto Query Language
MITRE	MITRE Corporation
MFA	Multi-Factor Authentication
POI	Point of Interest
REST	Representational State Transfer
SIEM	Security Information and Event Management
SOAR	Security Orchestration, Automation and Response
SOC	Security Operations Center
Sysmon	System Monitor
TTPs	Tactics, Techniques, and Procedures
UEBA	User and Entity Behaviour Analytics
UPN	User Principal Name
VM	Virtual Machine
YAML	YAML Isn't Markup Language

© CYberSynchorny, 2024

Executive Summary

D4.2 – Automated Response Toolkit v1 is part of Work Package 4 (WP4), which focuses on *Advanced Proactive Cyber Risk Monitoring and Rapid Response* under the CYberSynchrony project. This deliverable contributes to that goal by designing and developing an automated toolkit for incident response, integrating Security Orchestration, Automation, and Response (SOAR) mechanisms to ensure rapid and reliable mitigation of cyber threats. It represents a foundational step in implementing real-time, structured playbooks for proactive cyber defence within operational and simulated environments.

The work carried out under this deliverable includes the design and specification of automated playbooks, built upon real-world threat scenarios and industry-standard frameworks such as MITRE ATT&CK. These playbooks outline step-by-step response actions for a variety of cyber incidents, including phishing, ransomware, malware intrusions, and anomalous network behaviours. To enable real-time execution, the toolkit is developed with a modular architecture that integrates with SIEM platforms (Wazuh, Microsoft Sentinel), endpoint security solutions (Microsoft Defender for Endpoint), and IDS/IPS systems (Suricata, anomaly-based IDS models). The underlying infrastructure supports adaptive decision-making, automated containment actions, and real-time threat intelligence correlation to minimize response time and limit incident impact.

The main findings from the development process highlight the efficacy of automated playbooks in reducing manual intervention, improving detection-to-response speed, and enhancing security posture through continuous monitoring and automated mitigation. By leveraging AI-enhanced anomaly detection and centralized security event correlation, the system minimizes false positives while maintaining high accuracy in threat response. Testing and validation within controlled cyber range environments demonstrate the robustness of the designed workflows, ensuring their adaptability across multiple operational settings.

The purpose of this deliverable is to establish a foundation for real-time, automated cyber incident response within the broader CYberSynchrony framework. This first version of the Automated Response Toolkit sets the stage for further refinements, including the integration of machine learning-driven playbook optimizations, dynamic risk-based response adaptation, and cross-border incident coordination mechanisms. The insights gained from this phase will be instrumental in shaping D4.4 – Advanced Automated Response Toolkit, further enhancing the ability of organizations to detect, contain, and neutralize cyber threats efficiently and effectively.

© CYberSynchorny, 2024

1 Introduction

1.1 Mapping CYberSynchrony Outputs

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

Table 1: Adherence to CYberSynchrony's GA

Component no	Component title	Respective Document Chapter(s)	Justification
DELIVERABLE			
D4.2	Automated Response Toolkit v1	Chapters: 1 Introduction, 3 Design of Incident Response Playbooks, 4 Deployment Strategy for Automated Playbooks	The deliverable defines the design and initial implementation of automated response playbooks, integrating SIEM, SOAR, IDS, and endpoint security solutions to enhance cyber resilience. It ensures real-time automated incident handling, reducing manual intervention and improving response speed.
TASKS			
T4.1	Unveiling SIEM and Traffic Analysis Deployment	Chapter 2 - Architecture of the Defensive Cyber Operations (DCO)	This task focuses on bolstering cybersecurity through two primary methods. First, it deploys advanced SIEM systems to aggregate and analyse security-related data in real-time.
T4.3	Application of Incident Response Automation Playbooks	Chapter 3 - Design of Incident Response Playbooks, Chapter 4 - Deployment Strategy for Automated Playbooks	This task involves the development of structured response playbooks for cyber incidents, ensuring automated mitigation and response actions. The document outlines the playbook methodology, technical architecture, and deployment strategy.
T4.3.1	Design of Playbooks for Phishing, Ransomware, and Malware Response	Chapter 4 – Deployment Strategy for Automated Playbooks	Defines step-by-step response workflows for various cyber threats using SIEM and SOAR integration.
T4.3.2	Integration with SIEM (Wazuh, Sentinel) and SOAR for Automated Response	Chapter 3 – Design of Incident Response Playbooks	Describes how playbooks interact with detection systems and execute real-time response actions.

© CYberSynchorny, 2024

1.2 Linkage to other project outputs

Table 2 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
WP3	T3.3	D3.3 - Personalized Risk Assessment Tools	The risk assessment tools developed in WP3 contribute to defining the risk levels that inform automated response playbooks in WP4. Insights from vulnerability scans and security posture assessments feed into the SIEM and SOAR platforms to prioritize incident responses.
WP3	T3.4	D3.4 - Dynamic Risk Parameter Refinement	The refined threat intelligence and dynamic risk assessment parameters from WP3 improve the adaptability of the automated response toolkit by providing real-time updates to playbooks, allowing for more accurate response actions.
WP4	T4.1	D4.1 - SIEM & Traffic Analysis Deployment	SIEM deployment (Wazuh, Suricata) from WP4 serves as the detection backbone for the automated response toolkit. Logs and alerts generated in **D4.1** trigger playbooks in **D4.2**, ensuring that incident responses are executed immediately upon threat detection.
WP4	T4.2	D4.3 - Behavioural Analytics & Anomaly Detection	Entity behaviour analytics (UEBA) tools provide insights into anomalies that can be used to trigger automated response actions. The data generated in **D4.3** enhances **D4.2** by improving the decision-making process in playbooks.
WP5	T5.2	D5.2 - Incident Response Collaboration Framework	The automated response toolkit integrates with WP5's incident response collaboration framework to ensure that response actions are aligned across multiple entities. The toolkit supports coordinated responses to cyber threats at a cross-organizational level.
WP6	T6.1	D6.1 - Cybersecurity Awareness & Training	The automated playbooks developed in **D4.2** are incorporated into WP6's training platform for cybersecurity professionals. Playbooks serve as structured learning materials for **blue team exercises and response drills**.

© CYberSynchorny, 2024

1.3 Deliverable Structure

This deliverable is structured into five chapters, each covering a distinct layer of the design and implementation of the Automated Response Toolkit developed in CYberSynchrony:

Chapter 1 – Introduction provides a concise overview of the deliverable, summarizing the purpose, components, technologies used, and key findings related to the development of the Automated Response Toolkit v1.

Chapter 2 – Architecture of the Defensive Cyber Operations (DCO) details the technical architecture, including SIEM and SOAR integration, log flow, adversary emulation through CALDERA, and entity-based orchestration. It also describes how the architecture supports real-time and simulated execution environments.

Chapter 3 – Design of Incident Response Playbooks presents the methodology used to develop automated playbooks. It covers threat modelling, detection rules, entity-driven automation, and the embedding of playbooks into SOAR tools. The chapter includes practical examples (e.g., phishing, malware) and interactive validation through training simulations.

Chapter 4 – Deployment Strategy for Automated Playbooks focuses on the practical rollout of playbooks across production and test environments. It includes validation scenarios, performance metrics, and operational considerations for scaling, integration, and security control.

Chapter 5 – Conclusions summarises the key outcomes of the toolkit development, highlights its added value within the broader CYberSynchrony objectives, and outlines the next steps towards the advanced version of the toolkit in D4.4.

© CYberSynchorny, 2024

2 Architecture of the Defensive Cyber Operations (DCO)

The design of a resilient and responsive cybersecurity architecture requires the seamless integration of detection, correlation, and mitigation components. The Automated Response Toolkit developed in CYberSynchrony is built upon a distributed and modular architecture that combines log collection from diverse infrastructure layers, advanced threat detection through Security Information and Event Management (SIEM) platforms, and automated response orchestration via SOAR (Security Orchestration, Automation, and Response) systems.

The core objective of this architecture is to enable real-time, intelligent, and automated response to cybersecurity incidents with minimal manual intervention. This is achieved by tightly coupling the data collection mechanisms from network, endpoint, and application layers with centralized intelligence engines that perform log correlation, anomaly detection, and trigger executable playbooks upon identifying threats.

A critical element of the architecture is its interoperability with simulation tools, which are used to test the effectiveness of detection logic and validate the correct execution of response actions. The use of adversary emulation frameworks like CALDERA enables continuous refinement of the detection-response cycle, ensuring that the system evolves in response to emerging attack vectors.

This architecture is not monolithic but rather highly extensible, allowing for deployments in cloud-native, hybrid, or on-premise environments. It supports dynamic scaling, integration with external APIs, and adaptation to various security domains such as identity management, endpoint protection, and network perimeter control.

2.1 Overview of Defensive Cyber Operations (DCO) SIEM & SOAR Integration

The architecture of the Automated Response Toolkit combines the capabilities of Security Information and Event Management (SIEM) systems with Security Orchestration, Automation, and Response (SOAR) platforms to deliver a robust and adaptive incident response infrastructure. The toolkit integrates log collection, correlation, and automated response execution through playbook-driven logic.

Core Components:

- **Log Sources**: Network firewalls, endpoint detection agents (e.g., Defender for Endpoint), web servers, Active Directory logs, and containerized infrastructure (Docker Engine).
- **SIEM Platforms**: Wazuh and Microsoft Sentinel are used to collect, normalize, and correlate events. Sentinel provides cloud-native scalability and integration with Azure Logic Apps, while Wazuh supports on-premise log aggregation and analysis.
- **SOAR Component**: Azure Logic Apps enables the orchestration of automated actions defined in incident response playbooks. These workflows are triggered upon detection of specific entities and events in the SIEM.
- **Simulation Engine**: CALDERA, an open-source adversary emulation platform, is deployed in controlled environments to simulate real-world attacks aligned with the MITRE ATT&CK framework. It is used to validate detection rules and train the system to respond appropriately.

Data Flow:

1. **Log Ingestion**: Logs from distributed sources are ingested into the SIEM in real-time.
2. **Alert Generation**: SIEM platforms apply correlation rules and detection logic (including UEBA and threat intelligence feeds) to identify anomalies.

© CYberSynchorny, 2024

3. **Triggering Playbooks**: When an alert is generated, a pre-defined entity (e.g., compromised account) activates a specific playbook via the SOAR interface.
4. **Automated Response**: Actions include disabling compromised accounts, isolating endpoints, blocking IP addresses, and notifying analysts.
5. **Analyst Review & Feedback**: The system sends logs and actions to a central dashboard, enabling analysts to review results and adjust detection thresholds or playbook logic.

2.2 Diagram Representation

The architecture on Figure 1 is focused on simulating real-world adversary techniques using a custom-deployed CALDERA instance in a controlled Active Directory environment. The goal is to validate detection capabilities and improve automated playbook responses through log analysis, detection rule tuning, and AI-based correlation.

Figure 1 presents the technical architecture of the Automated Response Toolkit. It depicts the flow of security telemetry from various infrastructure sources (e.g., endpoints, network devices, containers) into SIEM platforms (Microsoft Sentinel and Wazuh), where detection logic is applied. The figure also shows the integration with the CALDERA adversary emulation framework for simulation-based validation, and the SOAR orchestration layer (e.g., Azure Logic Apps) responsible for executing automated response actions such as account disabling, endpoint isolation, and alert escalation. The architecture supports feedback loops, enabling continuous tuning of detection rules and playbooks.

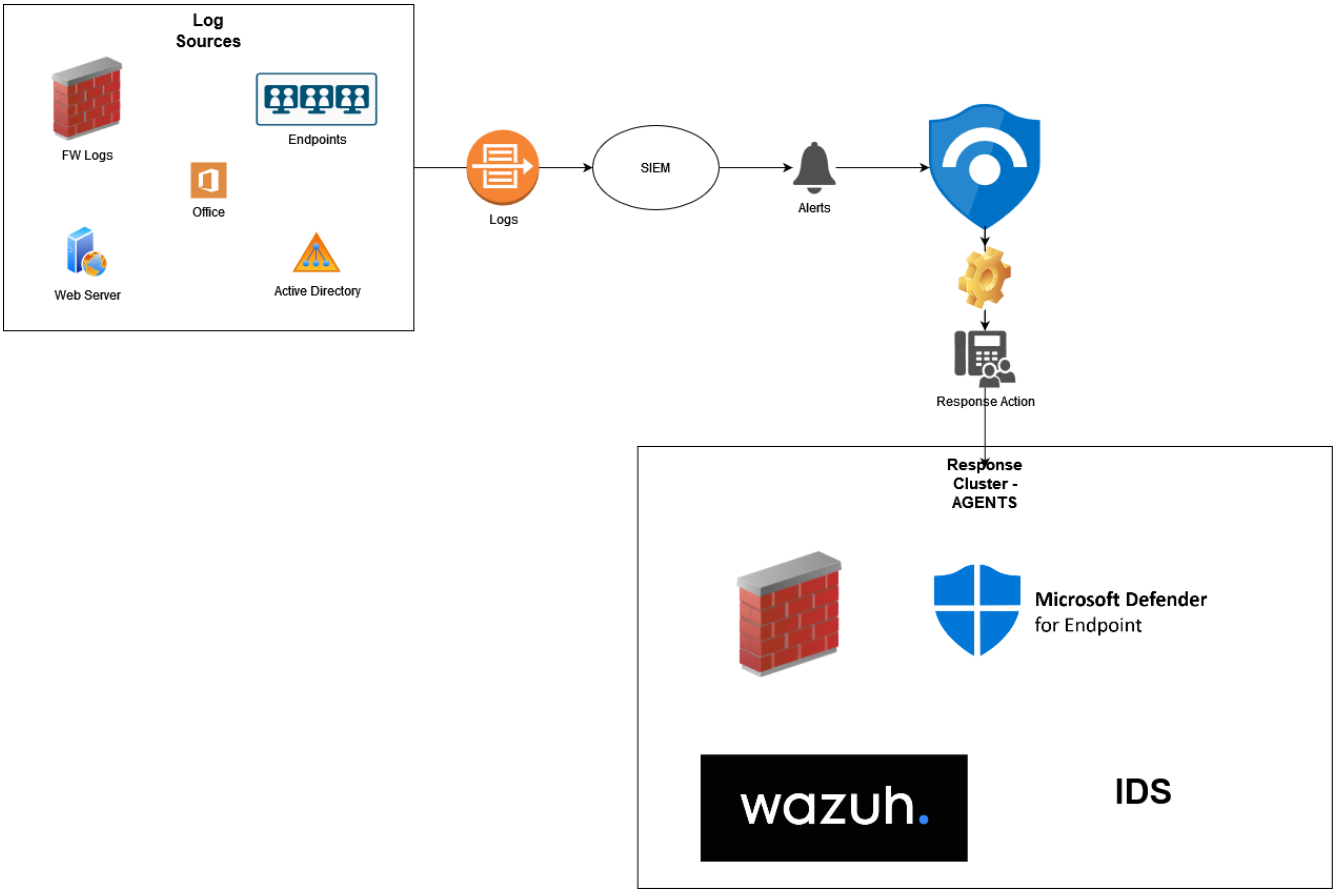

Figure 1: System architecture

A core element of the Automated Response Toolkit architecture lies in the integration of adversary emulation with detection and response workflows. To this end, a customized simulation environment has been developed

© CYberSynchorny, 2024

using the open-source platform CALDERA. This simulated environment is purpose-built to test and enhance the incident detection and response mechanisms supported by the project's SIEM and SOAR components.

The simulation architecture allows for controlled execution of adversary techniques aligned with the MITRE ATT&CK framework. CALDERA acts as the offensive simulation engine, capable of launching sophisticated attack chains that emulate real-world tactics, techniques, and procedures (TTPs). These attack flows include credential dumping, lateral movement, persistence, command and control communication, and privilege escalation. The operational environment consists of both Windows and Linux nodes, supported by extended logging mechanisms to ensure high visibility across all stages of the simulation.

On Windows systems, telemetry is gathered primarily through Sysmon and native Windows Security Event Logs, capturing granular process activities, network connections, file accesses, and login events. On Linux targets, monitoring is achieved via Auditd and Sysmon for Linux, enabling a comparable level of observability across Unix-based infrastructures. Additional system components, such as Docker containers and web servers, are also monitored to detect abnormal behaviour during simulation scenarios.

These logs are collected and correlated within the project's dual SIEM setup. Microsoft Sentinel serves as the cloud-native correlation and visualization platform, offering powerful Kusto Query Language (KQL) support and seamless integration with Azure Logic Apps for playbook execution. In parallel, Wazuh operates as a lightweight on-premise SIEM solution, particularly suited for edge or isolated environments. Both platforms ingest raw telemetry from the cyber range and process it against defined detection rules.

As attack simulations are executed within the CALDERA framework, telemetry from the host environment is streamed into the SIEM. Detected anomalies trigger predefined incident response playbooks through the SOAR layer, either in Sentinel via Logic Apps or through API-driven orchestrations from Wazuh. Automated actions such as disabling compromised user accounts, isolating endpoints, blocking command and control IPs, and notifying security analysts are executed in real-time.

This architecture creates a continuous feedback loop: each simulation not only validates existing detection and response logic but also contributes to the refinement of those mechanisms. Analysts can observe the impact of playbook execution, assess the effectiveness of detection logic, and leverage the insights to adjust or retrain both detection rules and the playbook workflow. Over time, this iterative approach leads to a more resilient and responsive cybersecurity posture.

In essence, this deployment paradigm embodies a Defensive Cyber Operations (DCO) methodology, bridging offensive testing with defensive readiness. It allows for measurable, repeatable improvement in incident response capability, providing the foundation upon which advanced automated playbooks can be tested, tuned, and operationalized within the broader CYberSynchrony framework.

Table 3: DCO Components

Component	Function	Technologies/Tools
Adversary Simulation	Executes real-world attack techniques	CALDERA (MITRE ATT&CK-based operations)
Log Collection	Captures detailed telemetry from endpoints and infrastructure	Sysmon (Windows/Linux), Auditd, Docker Logs, Web Server Logs

© CYberSynchorny, 2024

SIEM Platforms	Ingests, correlates, and analyzes logs to generate alerts	Microsoft Sentinel (cloud-native), Wazuh (on-premise)
SOAR Integration	Automates response actions based on SIEM alerts	Azure Logic Apps, Sentinel Playbooks, Wazuh API integrations
Response Actions	Automatically executes containment, isolation, and notification steps upon threat detection	Disable user accounts, isolate endpoints, alert analysts
Feedback Loop	Facilitates continuous improvement of detection and playbook logic through simulation and tuning	Analyst validation, detection and response rules

The Automated Response Toolkit serves as the technical foundation of the CYberSynchrony vision for Defensive Cyber Operations (DCO). At the core of this toolkit lie the incident response playbooks—modular, structured workflows that translate detection signals into concrete, automated mitigation actions. These playbooks are not standalone components but rather integral mechanisms embedded within the SIEM–SOAR architecture described in Chapter 3. They act as the dynamic execution layer of the overall DCO approach, enabling real-time responses to cyber threats with minimal human intervention.

Through tight integration with telemetry collection (via SIEM), decision-making logic (via detection rules and threat intelligence), and execution engines (via SOAR), playbooks operationalize the architecture's capability to detect, isolate, and remediate security incidents. Moreover, they serve as the primary method of orchestrating multi-stage actions that align with established frameworks like MITRE ATT&CK. As such, the playbooks constitute both the functional and strategic link between the system's technical architecture and its cybersecurity mission, forming the central mechanism by which DCO objectives are achieved.

The following chapter explores the design, methodology, and implementation of these playbooks in detail, showcasing how they are structured to handle specific threat types and how they are validated through both live execution and simulation environments.

© CYberSynchorny, 2024

3 Design of Incident Response Playbooks

The increasing complexity and velocity of cyber threats demand a shift from manual, reactive incident response to automated, structured, and repeatable processes. Within CYberSynchrony, incident response playbooks serve as the cornerstone of this paradigm shift. They encapsulate the operational knowledge of cybersecurity teams into executable, machine-readable workflows that can be triggered in real-time by security alerts and contextual signals.

An incident response playbook is defined as a predefined, modular set of actions that guide and automate the handling of specific types of cybersecurity incidents. These actions are designed to be executed either automatically or semi-automatically via SOAR (Security Orchestration, Automation, and Response) platforms, ensuring that the response is immediate, consistent, and aligned with best practices.

In CYberSynchrony, playbooks are not generic templates but rather custom-built, threat-specific response mechanisms tailored to real-world attack scenarios such as phishing, ransomware, and advanced persistent threats (APT). They are developed using a combination of simulation, detection logic, and orchestration technologies that ensure seamless integration with the organization's monitoring and infrastructure layers.

The primary technologies and tools employed in the design and execution of these playbooks include:

- **SIEM Platforms (Wazuh, Microsoft Sentinel)**: Responsible for collecting, analysing, and correlating logs from various sources (e.g., endpoints, firewalls, Active Directory). These platforms detect anomalies and raise alerts that serve as triggers for playbook execution.
- **SOAR Infrastructure (Azure Logic Apps)**: Acts as the automation engine that carries out the sequence of predefined actions, such as disabling user accounts, isolating endpoints, or sending notifications. Logic Apps are used extensively for their native integration with Microsoft Defender, Sentinel, and Azure AD.
- **Adversary Simulation Tools (CALDERA)**: Enable the validation of playbook effectiveness through realistic attack emulation. These simulations help fine-tune detection rules and response logic by exposing gaps in the current defence posture.
- **Entity-Driven Automation Models**: Playbooks are designed around entities (e.g., user, host, IP address) identified in the alert context, allowing for precise targeting of response actions.
- **AI-Assisted Detection Rule Refinement**: Leveraging machine learning models and AI agents, the system continuously aligns attacker activity patterns with log telemetry and automatically suggests improvements to detection rules and playbook conditions.
- **Game-Based Training Environments**: In parallel with operational deployment, the playbooks are also embedded in interactive simulations for training purposes. This dual-use model ensures both real-time protection and long-term capacity building for security teams.

By combining these technologies and methodologies, the project delivers not only a robust technical framework for incident response, but also a scalable knowledge transfer mechanism. Each playbook becomes a living artifact: continuously updated, tested, and refined across operational and training contexts.

The following sections describe in detail the structure of the playbooks, the methodology used in their design, how they are executed within the SOAR ecosystem, and how they are validated both technically and through gamified, user-facing environments.

3.1 Playbook Structure & Methodology

In the context of CYberSynchrony, incident response playbooks are conceived as structured, step-by-step operational guides that define the automated response to specific cyber threats. They provide a formalized approach to incident handling, allowing for consistency, repeatability, and auditability in organizational

© CYberSynchorny, 2024

cybersecurity operations. A playbook encapsulates both technical logic and decision-making processes, enabling incident containment, mitigation, and recovery in a timely and efficient manner.

The methodology adopted for the design of playbooks is driven by three fundamental principles:

1. **Threat-Centric Mapping**: Each playbook is tailored to a distinct type of cyber threat, such as phishing, ransomware, malware infection, or suspicious login activity. Threat intelligence sources (e.g., MITRE ATT&CK, OSCTI) are used to define adversary behaviours that serve as triggers.
2. **Entity-Based Execution**: Playbooks are initiated based on specific entities (user account, IP address, host, process ID) identified by SIEM alerts. This allows for precise and context-aware automated actions.
3. **SOAR-Ready Modularity**: Playbooks are built using modular components to enable reusability across different SOAR platforms. Each action is atomic and can be composed into more complex response chains.

The development of each playbook follows a multi-phase design cycle:

- **Phase 1 – Threat Identification**: Define the threat scenario using real-world cases and emulate it using adversary simulation tools (e.g., CALDERA). This includes the selection of TTPs aligned with the MITRE ATT&CK matrix.
- **Phase 2 – Detection Logic**: Develop and test correlation rules in the SIEM (Wazuh or Microsoft Sentinel). Rules are expressed in KQL or custom logic and are associated with specific entities.
- **Phase 3 – Response Workflow Definition**: Translate the identified threat and detection logic into a logical workflow of automated responses, including conditions, triggers, and fallback actions.
- **Phase 4 – Implementation in SOAR Platform**: Deploy the workflow in a SOAR platform (e.g., Azure Logic Apps for Sentinel). Configure required permissions and connectors for execution (e.g., Microsoft Graph API for user account management).
- **Phase 5 – Validation and Tuning**: Simulate attacks to validate detection accuracy and effectiveness of the playbook response. Incorporate analyst feedback and AI agent recommendations to refine logic.

This methodology ensures the playbooks are not only technically executable but also semantically aligned with actual security operations and enterprise-specific environments.

3.2 Playbook Execution & SOAR Integration

Playbooks are implemented on top of SOAR-capable infrastructures that allow them to be automatically executed upon alert generation from a SIEM platform. In CYberSynchrony, two distinct but interoperable approaches have been utilized:

1. **Microsoft Sentinel + Azure Logic Apps**
2. **Wazuh + External SOAR Execution Layer (e.g., Python automation engine or API orchestrator)**

Each playbook begins with a **trigger condition**, typically based on a correlation alert from the SIEM. For example, an alert indicating anomalous login behaviour (e.g., multiple failed logins followed by successful access from an unusual IP) will trigger the phishing detection playbook.

A typical flow for playbook execution in Azure Logic Apps includes:

- **Trigger**: Entity-based event from Sentinel (e.g., user account flagged as compromised).
- **Variable Initialization**: Extract entity properties such as UPN (User Principal Name) or IP address from the incident payload using functions like:

```
concat(triggerBody()?['Entity']?['properties']?['Name'],'@',triggerBody()?['Entity']?['properties']?['UPNSuffix'])
```

© CYberSynchorny, 2024

- **Logic Block Execution**: Use pre-built or custom connectors to apply automated actions, such as:
    - Set "accountEnabled" property to false (disable account)
    - Isolate device in Microsoft Defender for Endpoint
    - Add source IP to blocklist
    - Notify SOC via Teams or email
- **Permissions Management**: The Logic App identity is assigned appropriate roles (e.g., Microsoft Sentinel Responder, User.ReadWrite.All) to carry out actions.
- **Completion and Logging**: All actions are logged for traceability, with success/failure recorded and optionally stored in the Incident Comments section in Sentinel.

When Wazuh is used as the SIEM, playbook execution is typically handled by an external orchestration engine that listens to Wazuh alerts via its REST API, matches them against playbook triggers, and runs corresponding Python scripts to execute response logic.

Integration between the SIEM and SOAR layer ensures a near real-time response to evolving threats and minimizes manual effort for incident responders.

3.3 Case Studies – Example Playbook Scenarios

To demonstrate the effectiveness and practical implementation of playbooks within the CYberSynchrony framework, we present two illustrative scenarios:

3.3.1 Phishing Email Response Playbook

One of the most critical and frequent threat scenarios addressed by the Automated Response Toolkit is email-based phishing. The following playbook illustrates a comprehensive and modular approach to managing phishing incidents, as triggered by alerts in the SIEM system. The logic embedded in this playbook aims to not only detect and isolate the initial threat but also to identify impacted users and initiate downstream remediation and recovery actions.

© CYberSynchorny, 2024

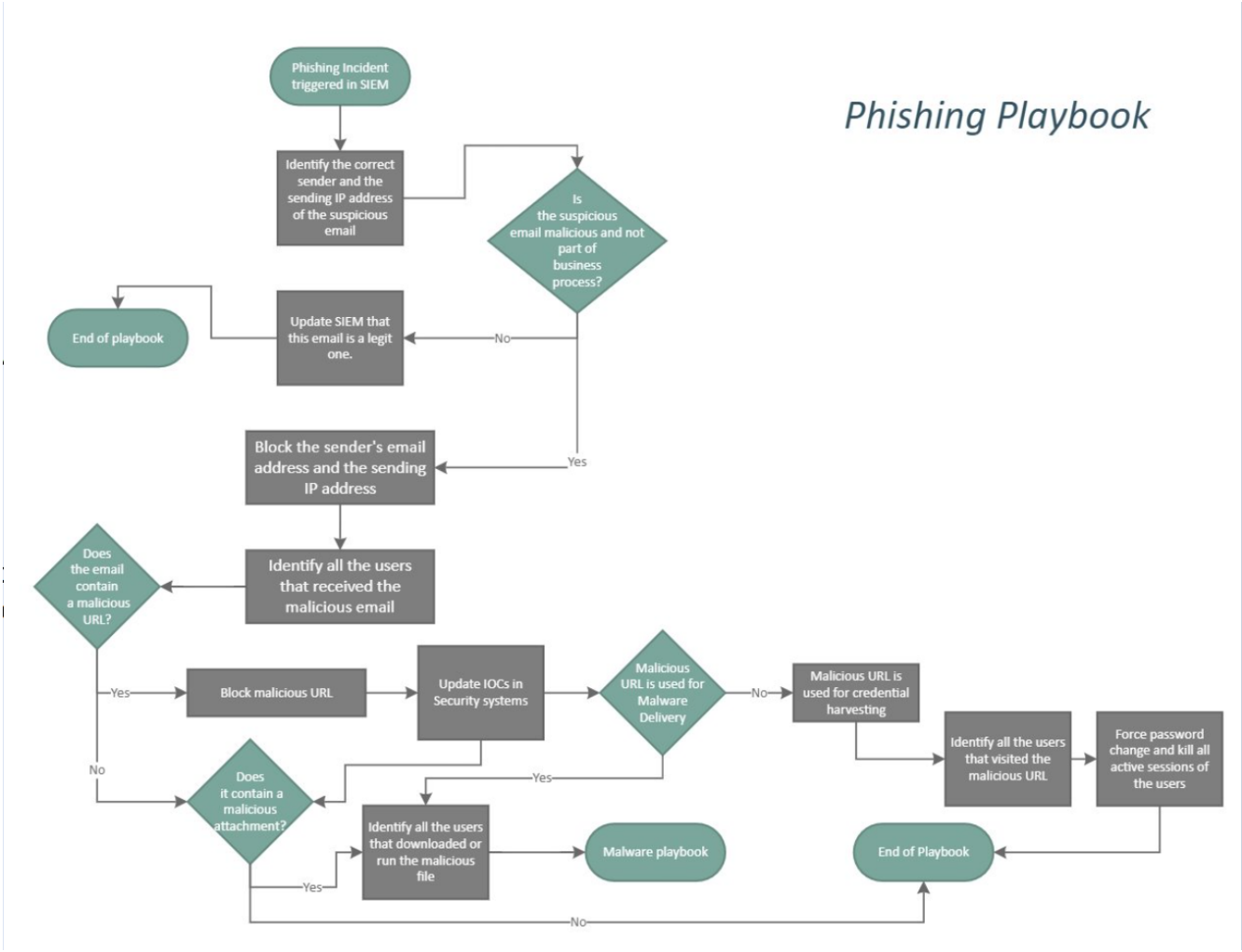

Figure 2: Phishing Playbook

The process begins (as you can see in the figure 2) when a phishing incident is detected by the SIEM (e.g., Microsoft Sentinel or Wazuh), typically based on correlation rules or anomaly detection applied to email metadata, behavioural indicators, or external threat intelligence feeds. Once triggered, the first stage of the playbook involves identifying the sender's email address and the sending IP of the suspicious message. A critical decision follows: determining whether the suspicious email is genuinely malicious or if it originates from a legitimate business process. This stage includes validation against whitelists, past communication records, or integrated threat intelligence databases.

If the email is deemed to be legitimate, the system marks the incident as a false positive and terminates the playbook. However, if the email is confirmed as malicious, the playbook proceeds to the containment phase. The sender's email address and IP are blocked at the relevant security control points (e.g., email gateway, firewall, Defender for Office 365), ensuring that future messages from the same source are neutralized.

The next phase involves the identification of all users who received the malicious email. This step is performed via log correlation in the SIEM or through direct integration with messaging systems such as Microsoft Exchange Online or Defender for Endpoint. Once recipients are identified, the playbook performs a deep inspection of the email content to determine if it contains a malicious URL. If such a URL is present, it is immediately added to blocklists and threat intelligence feeds, and the indicators of compromise (IOCs) are propagated across connected detection systems.

© CYberSynchorny, 2024

At this point, the playbook evaluates the purpose of the malicious URL. If it is used for malware delivery, the execution is redirected to a dedicated malware response playbook, which includes sandbox execution, endpoint isolation, and forensic collection. Alternatively, if the URL is used for credential harvesting—often via spoofed login pages—the system identifies users who interacted with the phishing link. For these users, a password reset is enforced, and all active sessions are terminated to eliminate the risk of ongoing compromise.

If the email does not contain a malicious URL, the playbook examines whether a malicious file attachment is present. In that case, users who downloaded or executed the file are identified using endpoint telemetry (e.g., Sysmon, Defender logs), and the execution again transitions into the malware response playbook.

Each branch of this decision tree is built upon conditional logic and entity-based automation, allowing precise targeting of response actions without unnecessary disruption to unaffected users or systems. *Entity-based automation* refers to the ability of the system to tailor actions based on the attributes and status of specific entities - such as users, devices, IP addresses, or applications - rather than applying blanket rules. This enables more contextual and granular responses. Furthermore, all decisions and actions are logged for auditing and forensic traceability. Furthermore, all decisions and actions are logged for auditing and forensic traceability.

This playbook reflects a robust incident response methodology, transforming initial detection into a full-spectrum automated workflow that mitigates threat propagation, neutralizes adversary infrastructure, and secures user identities. It also serves as a foundation for Blue Team training and game-based validation scenarios, where security analysts can practice executing the playbook logic under simulated conditions, as described in the interactive components of this deliverable.

Table 4: Playbook Components

Component	Function	Technologies/Tools
Adversary Simulation	Executes real-world attack techniques	CALDERA (MITRE ATT&CK-based operations)
Log Collection	Captures detailed telemetry from endpoints and infrastructure	Sysmon (Windows/Linux), Auditd, Docker Logs, Web Server Logs
SIEM Platforms	Ingests, correlates, and analyses logs to generate alerts	Microsoft Sentinel (cloud-native), Wazuh (on-premise)
SOAR Integration	Automates response actions based on SIEM alerts	Azure Logic Apps, Sentinel Playbooks, Wazuh API integrations
Response Actions	Automatically executes containment, isolation, and notification steps upon threat detection	Disable user accounts, isolate endpoints, alert analysts
Feedback Loop	Facilitates continuous improvement of detection and playbook logic through simulation and tuning	Analyst validation, detection and response rules

© CYberSynchorny, 2024

3.3.2 Simulated cyber scenarios and game-based Playbook

In addition to automated response workflows, the CYberSynchrony project incorporates an interactive training and evaluation layer through the use of simulated cyber scenarios and game-based playbook execution. These training scenarios replicate real-world threat conditions and allow security teams to actively engage with the playbooks in a gamified, controlled environment. This approach supports both Blue Team (defensive) and Red Team (offensive) perspectives.

A key feature of this component is the "Learning Content Layer", implemented in the PurpleTeamAssessment and DCO framework, where users engage with attacks via an interactive dashboard, selecting Points of Interest (POIs), executing playbook steps, and receiving feedback based on response accuracy and timing.

Figure 3 presents the game-based simulation flow, structured around a typical adversary attack lifecycle. The diagram includes sequential phases such as Initial Recon, Initial Compromise, Establish Foothold, Escalate Privileges, Internal Recon, Maintain Presence, and Complete Mission — all aligned with the MITRE ATT&CK framework. Between these phases, players interact with Points of Interest (POIs) and execute predefined playbook steps. Each action triggers feedback loops that inform the next step: correct actions advance the scenario, while incorrect ones provide additional hints or escalate the simulated incident. This cyclical structure enables iterative learning and reflects real-world decision-making under pressure.

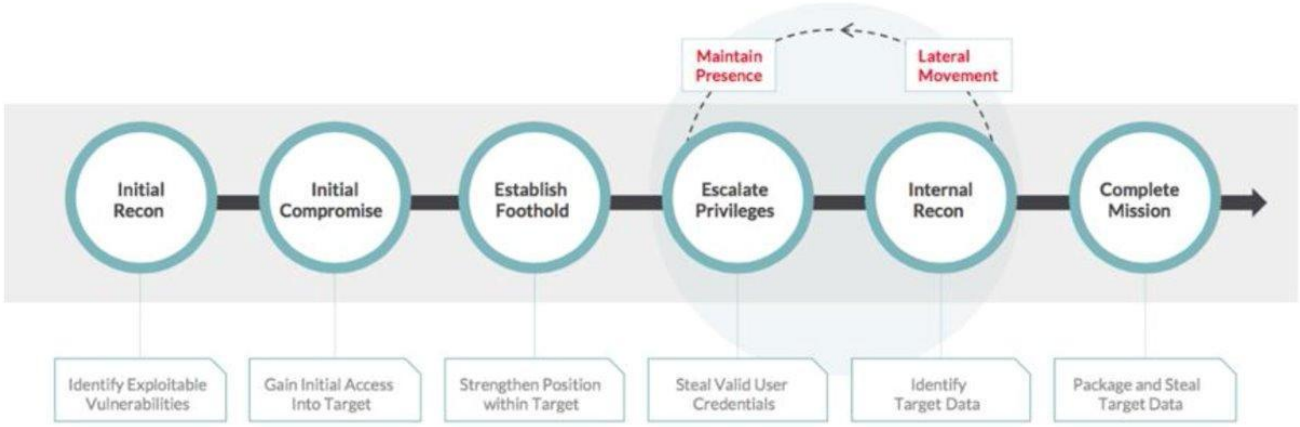

Figure 3: Game based scenario steps

Interactive Playbook Execution – Training Scenario Flow

Each scenario, as you can see in figure 3, begins with an **"Attack Introduction"**, explaining the simulated adversarial campaign (e.g., phishing, lateral movement, privilege escalation). At the top-right corner of the interface, the user is presented with:

- A summary of the attack phase.
- An "Attack Log" showing events and indicators of compromise.
- A "Glossary" to support non-technical participants with key definitions.

The user navigates through sequential stages, where they must:

1. **Select Points of Interest (POIs)**: These are clickable hotspots on a simulated network or endpoint (e.g., user mailbox, workstation, Active Directory controller) where evidence or threat actions are located.
2. **Choose Unlocked Playbook Actions**: These reflect the available automated or manual steps, such as:
    - Quarantine user
    - Isolate endpoint

© CYberSynchorny, 2024

- Block domain or IP
- Launch forensic analysis

Each action is mapped to a real playbook step as implemented in the Logic App or SOAR orchestration layer. Successful actions unlock subsequent phases or provide mitigation bonuses, while incorrect or delayed responses lead to escalation of the simulated incident.

Educational Mini-Games and Testing

To enhance engagement and cognitive reinforcement, the platform includes mini-games that simulate:

- Hash matching and malware family identification.
- Command-line forensic analysis.
- SIEM alert triage and event correlation.

These games (figure 4) serve dual purposes: reinforcing technical knowledge and evaluating the user's familiarity with the playbook flow. Performance metrics such as response time, accuracy, and escalation avoidance are recorded and linked to the maturity of the playbook deployment.

Figure 4: Game Example

Applied Case Example – Multi-Phase Ransomware Simulation

In this advanced scenario, the game environment emulates a ransomware infection progressing across four phases:

- **Initial Infection**: A phishing email with a malicious Excel macro triggers a payload.
- **Persistence**: A scheduled task is created to maintain access.
- **Encryption**: Files across mapped drives are encrypted with random AES keys.
- **Command & Control**: A beacon is sent to an attacker-controlled domain.

The trainee is required to respond using the available playbook:

- Step 1: Quarantine the affected host via Defender API.

© CYberSynchorny, 2024

- Step 2: Disable user via Microsoft Graph call.
- Step 3: Block outbound connection via Sentinel-managed firewall API.
- Step 4: Launch endpoint forensic workflow and isolate backups.

This simulation matches exactly the backend implementation of the ransomware containment playbook described previously and serves as both a training tool and a validation mechanism for playbook completeness and usability.

Table 5: Summary of the Phishing Response Playbook

Phase	Automated Action	Objective
Trigger & Validation	Detect phishing alert in SIEM and extract sender information	Initiate incident response and define scope
Trigger & Validation	Validate if email is part of a legitimate business process	Prevent false positives and avoid unnecessary disruption
Containment	Block sender's email address and originating IP	Prevent further delivery of malicious content
Impact Assessment	Identify all users who received the email	Determine exposure and enable user-focused response
URL Inspection	Check for presence of malicious URLs	Detect embedded threats for further blocking
URL Inspection	Block URL and update threat intelligence (IOCs)	Ensure threat visibility across all security systems
Credential Harvesting Path	Identify users who clicked the link and force password reset	Prevent account compromise and stop lateral movement
Credential Harvesting Path	Terminate all active sessions of impacted users	Eliminate persistence mechanisms used by attackers
Malware Delivery Path	Redirect to malware playbook	Enable advanced containment and forensic analysis
Attachment Inspection	Identify and track users who opened or ran a malicious file	Contain endpoint threats and transition to malware handling
Conclusion	Log actions and mark incident as resolved	Ensure traceability and prepare for audit/reporting

© CYberSynchorny, 2024

4 Deployment Strategy for Automated Playbooks

The successful operationalization of incident response playbooks within the Automated Response Toolkit depends on a well-defined deployment strategy that ensures technical interoperability, organizational readiness, and continuous validation. The deployment approach adopted in CYberSynchrony emphasizes modularity, scalability, and alignment with both production environments and controlled cyber ranges.

4.1 Deployment Environment

The deployment of automated playbooks is carried out across two complementary environments: (a) operational security infrastructure and (b) simulation and training environments. This dual deployment enables both real-time protection and iterative refinement of the playbook logic.

In the production context, playbooks are deployed within the existing SIEM–SOAR pipeline. Microsoft Sentinel, integrated with Azure Logic Apps, serves as the cloud-native environment for most organizations leveraging Microsoft 365 services. Playbooks are implemented as Logic App workflows triggered by entity-based alerts generated in Sentinel, with permissions and API connectors configured to enable actions across Microsoft Defender, Azure Active Directory, and Microsoft Purview. In parallel, Wazuh-based deployments allow for on-premise orchestration using RESTful APIs and Python-based execution engines for environments without cloud-native infrastructure.

To enable continuous validation and advanced experimentation, CYberSynchrony also deploys playbooks within a cyber range environment hosted at the Focal Point (FP) facility. This testbed replicates enterprise infrastructure including Active Directory, email servers, web services, containerized applications, and hybrid cloud endpoints. Playbooks can be triggered by simulated alerts generated from adversary emulation tools such as CALDERA, which executes real-world attack scenarios in line with the MITRE ATT&CK framework. This approach ensures that playbooks are validated not only on the basis of static input/output correctness but also in dynamic, evolving threat conditions.

Furthermore, deployments are supported by automated pipelines for version control, testing, and telemetry collection. Playbooks are packaged as YAML or Logic App JSON templates and deployed using Infrastructure-as-Code (IaC) tools (e.g., Terraform or Azure Resource Manager), allowing for reproducibility and rollback capability.

4.2 Testing & Validation of Playbooks

Robust testing and validation are critical to ensuring that playbooks respond effectively to threats without introducing operational risks or false positives. In CYberSynchrony, playbooks are evaluated against multiple technical and performance dimensions:

a. Functional Accuracy

Each playbook is tested to confirm that it executes the intended actions when triggered by relevant SIEM alerts. This includes verifying:

- Correct entity extraction from incident payloads.
- API call success and correct execution of mitigation steps (e.g., user disablement, session termination).
- Logging of actions in centralized dashboards for auditability.

b. Performance Metrics

Quantitative metrics are used to measure the operational performance of the playbooks:

© CYberSynchorny, 2024

- **Detection-to-response latency**: Time elapsed between SIEM alert generation and SOAR response execution.
- **Automation success rate**: Percentage of actions completed without human intervention or failure.
- **False positive reduction**: Number of confirmed false positives avoided due to contextual validation logic.

c. Scenario-Based Validation

Playbooks are subjected to adversarial simulations using CALDERA, which generates alerts corresponding to realistic attack chains. These scenarios validate the effectiveness of the detection and the appropriateness of the playbook response in near-real time. Validation also includes edge cases, such as malformed alerts, missing entity fields, or simultaneous multi-vector attacks.

d. Integration Testing

Playbooks are tested for integration with external systems and APIs, including:

- Microsoft Defender for Endpoint (device isolation, scan initiation)
- Microsoft Graph API (user management, password reset)
- DNS and firewall automation platforms (block IPs and domains)
- ServiceNow or ticketing systems (automated incident creation)

In parallel, playbooks are validated in a training context, using game-based scenarios to test their usability and comprehensibility from the analyst's perspective. These exercises ensure that newly onboarded personnel can follow and execute the playbook logic with minimal instruction.

f. Security and Access Control

All playbooks are deployed with strict access controls based on the principle of least privilege. The SOAR platform’s managed identity is granted only the permissions needed for each action, such as:

- User.ReadWrite.All
- Directory.ReadWrite.All
- Microsoft Sentinel Responder

These permissions are reviewed periodically to ensure compliance with organizational security policies and EU data protection regulations.

This structured deployment strategy ensures that the Automated Response Toolkit can be scaled, maintained, and continuously improved across a variety of operational contexts. It also lays the groundwork for future phases of the CYberSynchrony project, including the integration of adaptive, risk-based playbook tuning and AI-driven incident orchestration.

© CYberSynchorny, 2024

5 Conclusions

The first version of the Automated Response Toolkit presented in this deliverable marks a significant milestone in the realization of the CYberSynchrony vision for proactive and automated cyber defense. Through the seamless integration of SIEM platforms, SOAR orchestration layers, adversary simulation tools, and structured incident response playbooks, the toolkit enables a new operational paradigm for DCO.

The design and implementation of incident-specific playbooks—tailored for phishing, malware, ransomware, and insider threats—demonstrate the practical feasibility of real-time, automated response workflows. These playbooks not only accelerate containment and recovery but also reduce manual workload for analysts and contribute to an overall improvement in the organization's security posture. Their execution is validated through both live deployment in hybrid infrastructures and simulation exercises using adversary emulation platforms, ensuring both effectiveness and adaptability.

Deliverable D4.2 – Automated Response Toolkit v1 is developed under Work Package 4 (WP4) – Advanced Proactive Cyber Risk Monitoring and Rapid Response. It directly contributes to Task 4.3 by designing and implementing threat-specific incident response playbooks and automating their execution through SOAR platforms. Furthermore, it builds upon the outcomes of Task 4.1 (SIEM deployment) and provides the technical basis for subsequent advancements foreseen in D4.4. Through this work, D4.2 operationalizes the objectives of WP4 by enabling real-time detection-to-response capabilities and validating them through adversary emulation and gamified training environments.

The work also introduced game-based learning and scenario testing as a novel way to combine response automation with cybersecurity training. This dual-use model reinforces knowledge transfer while allowing for continuous feedback and improvement of the automated workflows.

Moving forward, this toolkit will evolve into a more advanced and adaptive system, as foreseen in D4.4 – Advanced Automated Response Toolkit. Future enhancements will include machine learning-driven rule refinement, risk-based response prioritization, integration with cross-border incident coordination frameworks, and tighter coupling with behavioural analytics. Additionally, the growing body of training data and real-world feedback from playbook execution will further strengthen the system's predictive and prescriptive capabilities.

© CYberSynchorny, 2024

Annex I: Example of DCO implementation

For this Playbook we are going to use Azure Logic Apps and Microsoft Sentinel.

1. The Flow starts with an Incident that must be triggered in Microsoft Sentinel that there is a suspicious activity by the compromised account. Microsoft Sentinel Incidents contain key items of the incident which are called Entities. One of these entities is the compromised account. Our logic app will run on the compromised account entity as it appears below.

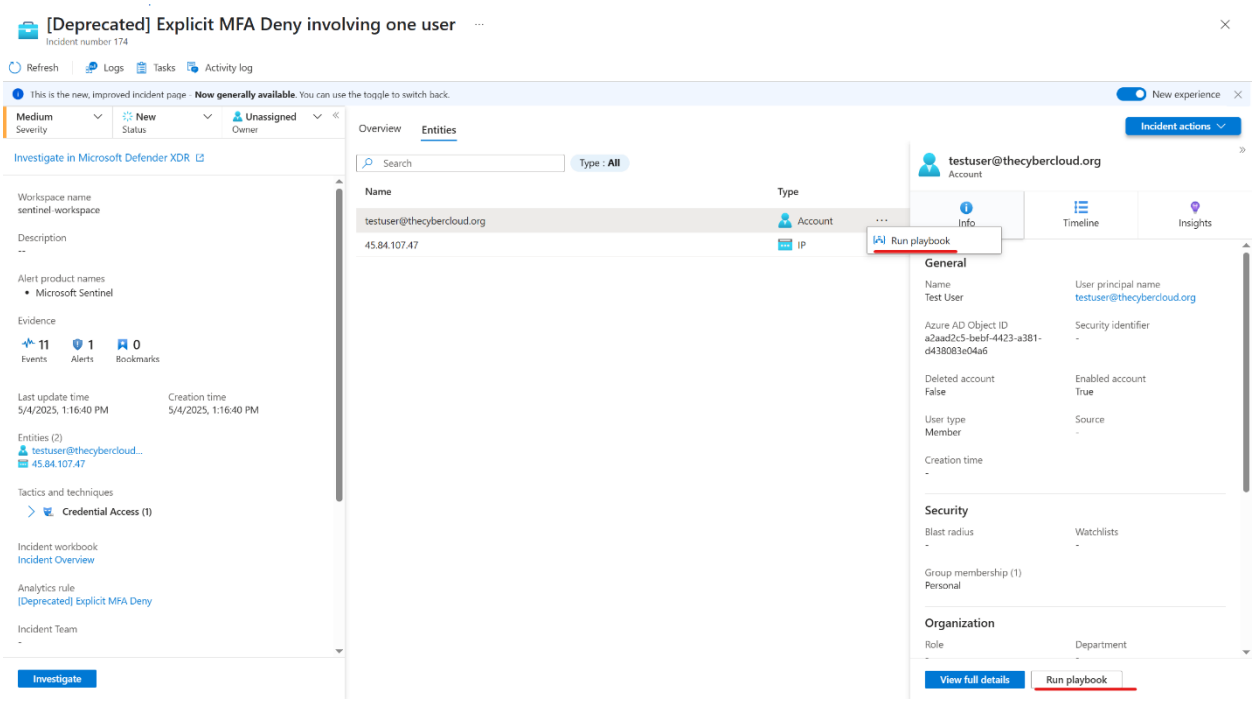

Figure 5: Sentinel case_1st step

2. Now that we know how to run our playbook, let's create it.
    a. The first step of the playbook is a trigger action based on Entities. Because the action is to disable an account the Trigger action should be based on Account type of Entity.

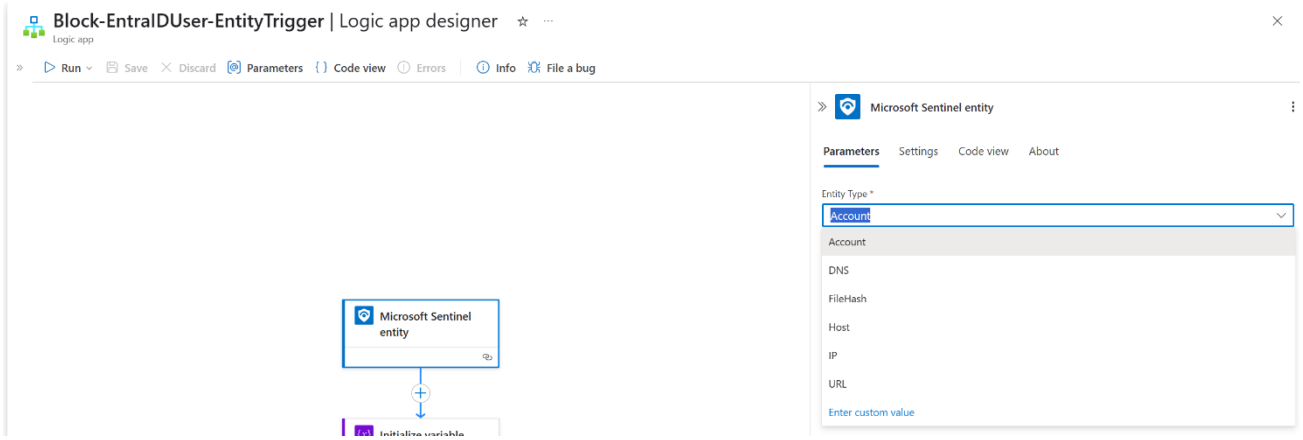

Figure 6: Trigger action_2nd step

© CYberSynchorny, 2024

b. To perform the disable action on the account we need to extract The Account UPN (User principal Name) from the trigger action described previously and put it in a variable. So, we first need to initialize a variable.

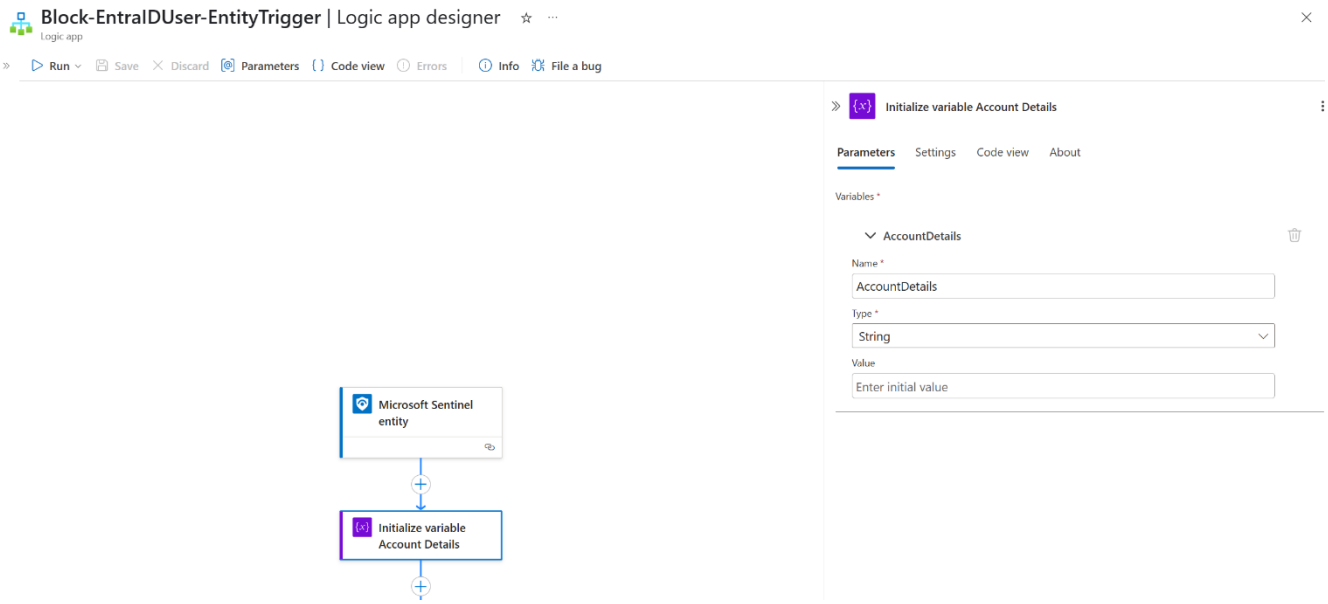

Figure 7: Disable action_3rd step

c. After we extract the Account UPN (User principal Name) from the json body of the trigger action and put it in this variable. To extract the UPN from the json body we need to use the following Login App function: *concat(triggerBody()?['Entity']?['properties']?['Name'],'@',triggerBody()?['Entity']?['properties']?['UPNSuffix'])*

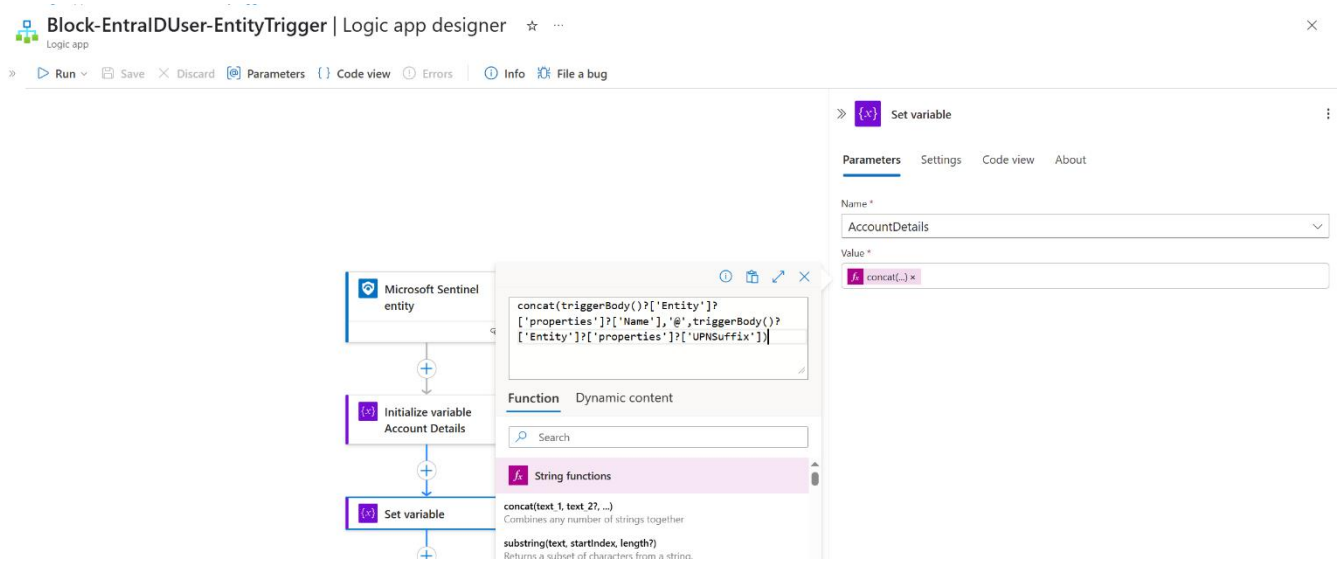

Figure 8: Extract UPN_4th step

d. Now that we have the Account UPN let's create the action that disables the account. There is a ready to use connector in Azure Logic Apps where you can change properties of accounts. One of the properties is if the account is enabled or disabled. So, we will use this connector to change the *Account Enabled* property to No.

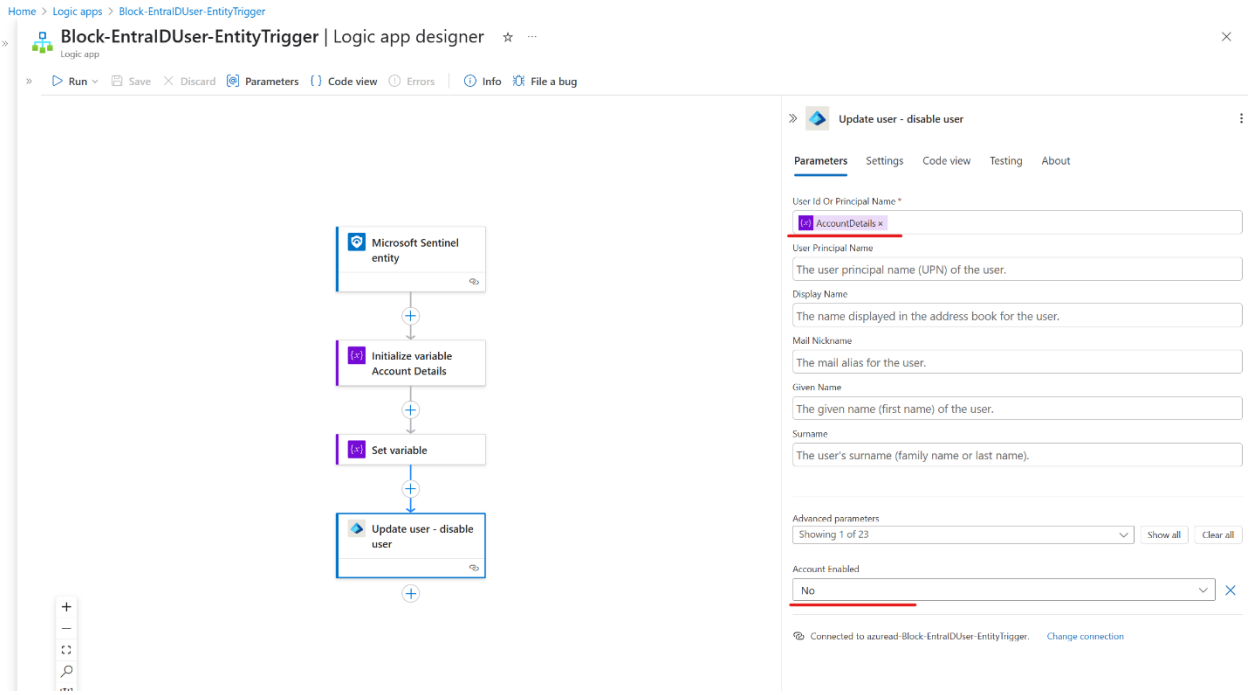

Figure 9: Connector in Azure Logic Apps_5th step

e. For the Logic App to run we need to add specific permission to the logic app Identity.
   i. Assign Microsoft Sentinel Responder role to the Playbook's managed identity.
   ii. Grant User.Read.All, User.ReadWrite.All, Directory.Read.All, Directory.ReadWrite.All permissions to the managed identity.

© CYberSynchorny, 2024

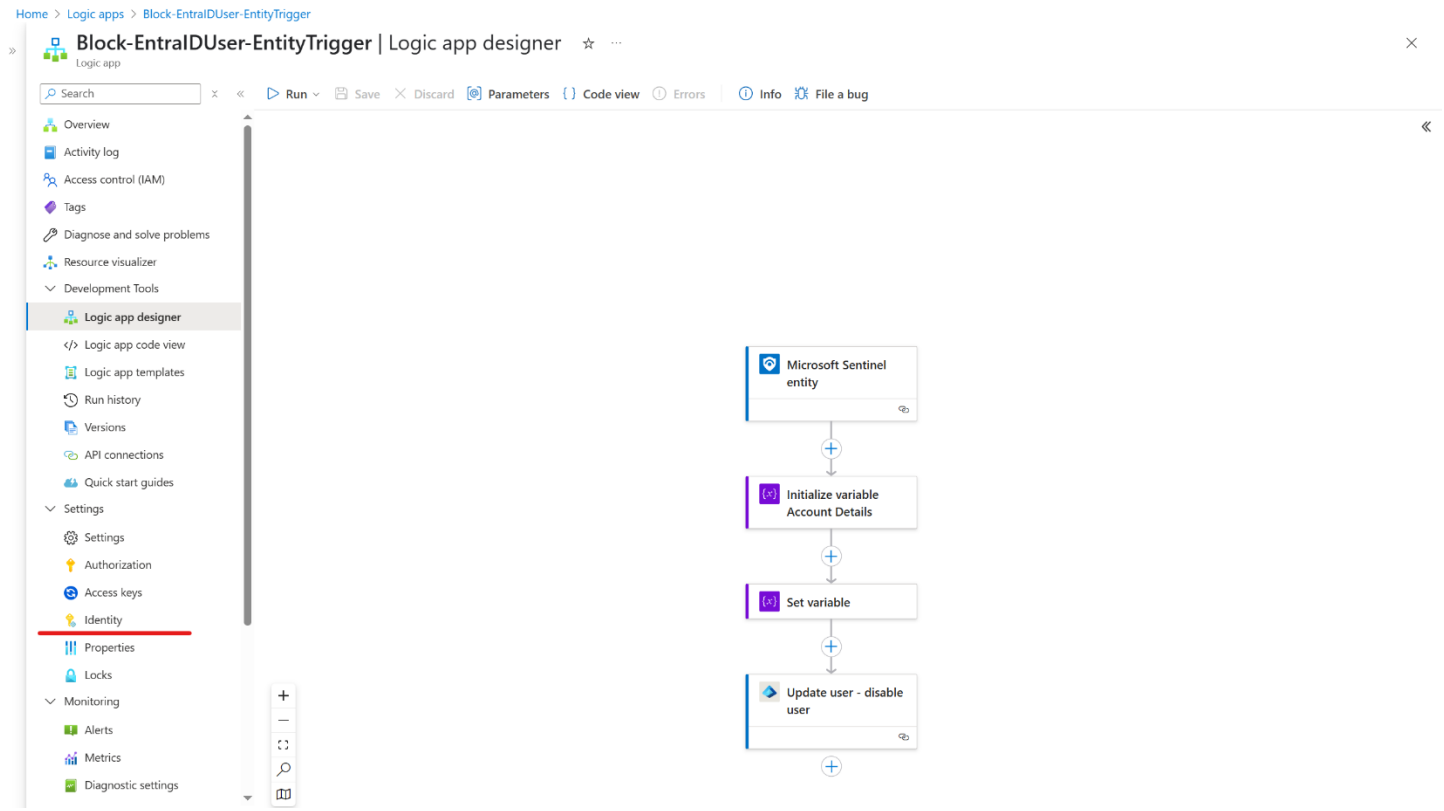

Figure 10: Permission to the logic app Identity_6th step

f. Let's now run the logic app on the entity of the Incident triggered as it appears in the first screenshot.

g. Let's review what happened. The logic app was executed successfully and the account got disabled.

© CYberSynchorny, 2024

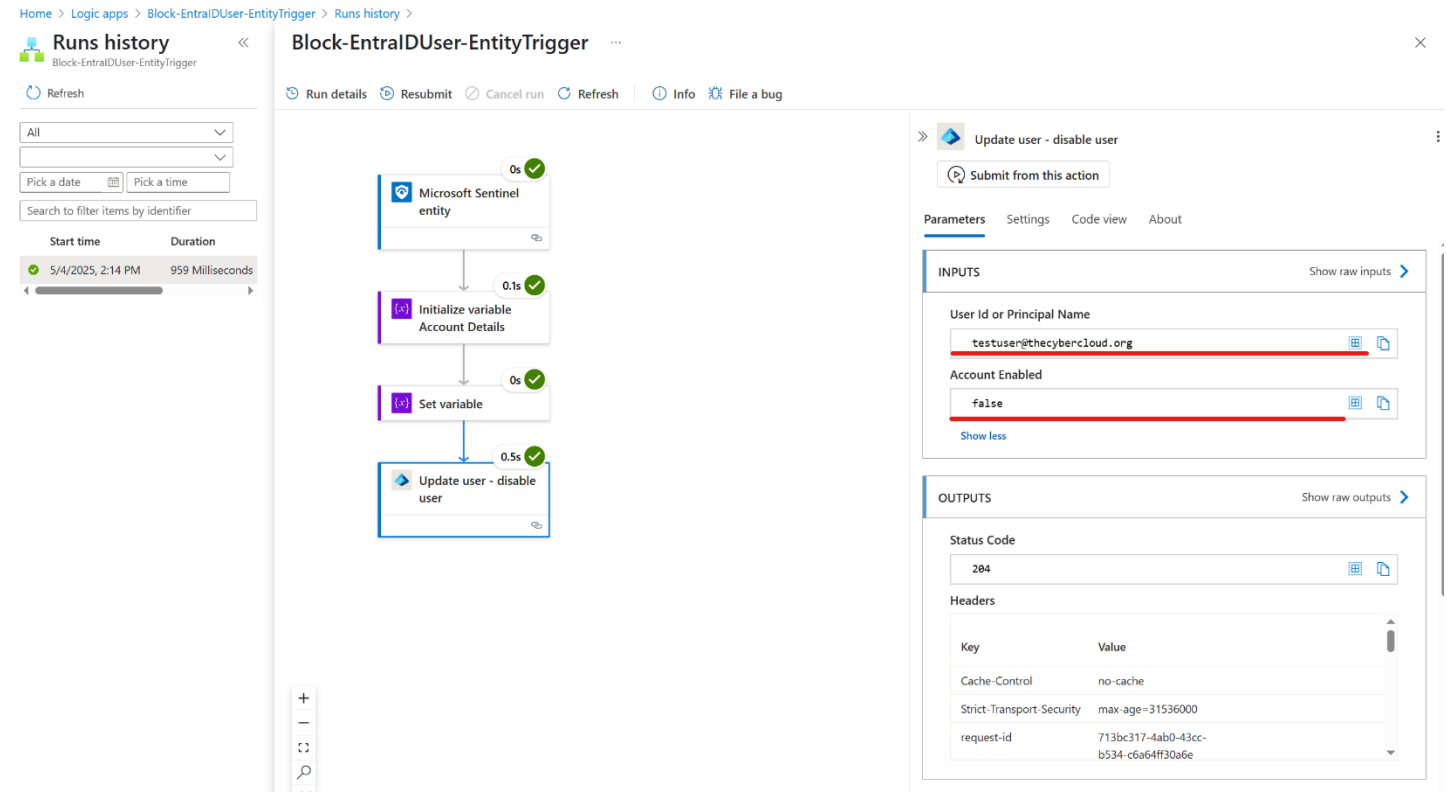

Figure 11: Run logic app_7th step

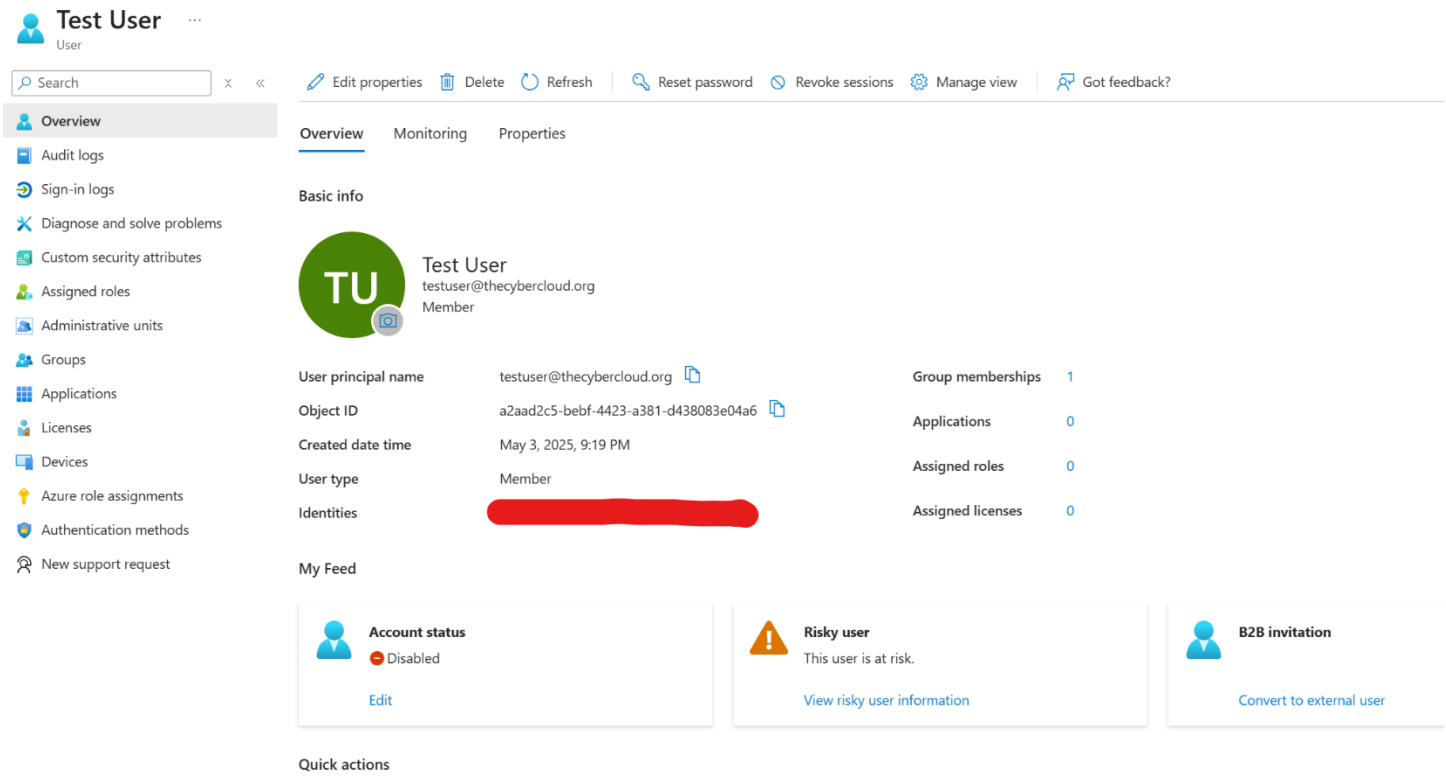

Figure 12: Final check