

When Trust Becomes the Attack Vector: Analysis of the EmEditor Supply-Chain Compromise

 techcommunity.microsoft.com/blog/microsoftsecurityexperts/when-trust-becomes-the-attack-vector-analysis-of-the-embeditor-supply-chain-compromise/4499552

Parth_Jamodkar

March 10, 2026

Blog Post

When Trust Becomes the Attack Vector: Analysis of the EmEditor Supply-Chain Compromise



Microsoft Security Experts Blog

6 MIN READ

There are countless security vulnerabilities and trust in software distribution is one of them. Users install applications from official vendor websites, enterprise controls allowlist signed software, and automated update mechanisms routinely pull code from trusted infrastructure. This post takes you through a supply-chain compromise targeting the EmEditor software distribution channel, where attackers weaponized a trusted WordPress-based download infrastructure to selectively deliver a trojanized MSI installer. It demonstrates how conditional server-side logic, installer abuse, and living-off-the-land techniques can bypass traditional defenses and enable credential theft at scale. It includes how the malicious installer behaved, and how defenders can detect and mitigate similar threats.

Attackers compromised the upstream distribution mechanism for **EmEditor**, a widely used Windows text editor. Instead of delivering malware through phishing or malicious domains, the attackers manipulated server-side logic on the official download site to selectively serve a trojanized installer to public users while preserving legitimate content for administrators.

This campaign highlights two recurring challenges in defending modern environments:

- **Upstream trust abuse:** Malicious payloads delivered from legitimate, trusted domains.
- **Selective evasion:** Conditional logic designed to evade validation, monitoring, and routine testing.

Why this matters more now

Attackers increasingly favor techniques that “live off trust” rather than exploit obvious weaknesses. As organizations harden email gateways, enforce attachment scanning, and restrict macro execution, supply-chain compromises provide an attractive alternative path to initial access.

In this case, the attack required no user interaction beyond installing trusted software and relied entirely on legitimate operating system components for execution. This combination significantly reduced detection opportunities and increased the likelihood of successful compromise.

1. Scope and unique insight

This is not a traditional malware delivery campaign. The distinguishing characteristics include:

- Server-side conditional manipulation rather than client-side redirection
- Weaponization of a legitimate MSI installer
- Use of Windows Installer custom actions to execute in-memory payloads
- Credential theft via named pipe injection without dropping additional executables.

The investigation demonstrates how endpoint, network, and installer telemetry must be correlated to uncover attacks that intentionally blur the line between legitimate and malicious activity.

Server-side conditional tampering enabling selective MSI delivery.

Attackers compromised the software distribution pipeline to selectively serve a trojanized MSI installer to public users while preserving legitimate behavior for administrators. The malicious installer abused Windows Installer execution, in-memory PowerShell staging, and command-and-control infrastructure to enable credential access.

2. Technical analysis

Discovery and investigation overview

The activity was identified through proactive threat hunting across Microsoft Defender telemetry, focusing on anomalous installer behavior and unexpected PowerShell execution chains originating from trusted software installs.

Multiple signals converged during investigation:

- PowerShell execution spawned from `msiexec.exe`
- Network connections from installer-initiated processes to suspicious domains.
- Browser process injection without corresponding file creation events

Together, these indicators pointed to a compromised installer rather than a post-installation infection vector.

2.1 Upstream breach: server-side tampering

The initial compromise occurred on a public-facing WordPress environment associated with the EmEditor download infrastructure. Attackers likely gained access via a vulnerable plugin or exposed administrative interface and deployed a web shell to maintain persistence.

Rather than modifying core WordPress files or defacing the site, the attackers injected **conditional PHP logic** into a theme-level file (footer.php). This logic dynamically altered download behaviour based on visitor context:

- **Authenticated administrators** were served the legitimate EmEditor MSI.
- **Unauthenticated public visitors** were redirected to a trojanized MSI hosted under /wp-content/uploads/.

This split-view evasion technique allowed attackers to weaponize the official domain while avoiding detection by internal validation workflows, routine administrative testing, and automated integrity checks.

2.2 Trojanized MSI installer behavior

The malicious installer closely resembled the legitimate EmEditor MSI in name and functionality but embedded a **custom action** that executed during installation.

Key characteristics included:

- Execution via msiexec.exe -Embedding
- Silent spawning of powershell.exe
- In-memory execution using Invoke-RestMethod piped to Invoke-Expression

The MSI was digitally signed, but not by the legitimate Emurasoft certificate. Instead, it used a certificate issued to a non-trusted publisher that nonetheless reduced user suspicion.

During installation, Windows cached the MSI in C:\Windows\Installer\, enabling silent re-execution and complicating forensic reconstruction.

2.3 Command-and-control infrastructure

The PowerShell stager connected to attacker-controlled infrastructure using multiple fallback paths to ensure reliability:

- **Primary endpoint:** emeditorjp[.]com
- **Mirror endpoint:** emeditorde[.]com, emeditorgb[.]com, emeditorsb[.]com
- **Second-stage delivery:** cachingdrive[.]com

Connections were observed over HTTP, HTTPS, and TCP, indicating deliberate redundancy. This infrastructure delivered a second-stage payload designed to operate entirely in memory.

2.4 Credential access and browser injection

The second-stage payload targeted browser processes, including chrome.exe and msedge.exe, using **named pipe injection** techniques.

By injecting directly into existing browser processes, the malware avoided creating new processes or dropping additional files. This enabled access to:

- Browser-stored credentials
- Authentication cookies
- Active session tokens for web and enterprise services

The absence of obvious malware artifacts strongly suggests that credential theft and session hijacking were the primary objectives.

Impact and targeting

- **Potential targets:** Enterprises and individual users installing EmEditor during the affected window.
- **Industries:** Broad, including technology, professional services, and regulated sectors
- **Impact:** Credential compromise, session hijacking, potential lateral movement
- **Scope:** Limited in time but high impact due to trusted distribution channel

3. Mitigation and protection guidance

3.1 What to do now if you're affected.

Organizations that suspect exposure should take immediate steps to contain potential compromise:

- Isolate affected endpoints from the network
- Block known malicious domains and IP addresses at DNS and firewall layers.
- Force credential resets for users on impacted systems.
- Review active browser sessions and revoke tokens where possible.
- Conduct full endpoint scans using Microsoft Defender XDR

3.2 Defending against similar attacks.

To reduce exposure to supply-chain attacks of this nature, organizations should consider the following measures:

General security practices

- Enforce multi-factor authentication across cloud and enterprise services.
- Limit browser-stored credentials and encourage password managers with strong protections.
- Monitor software installation activity for anomalous child process behavior.

Endpoint and installer protections

- Enforce stricter code-signing validation policies.
- Monitor msiexec.exe spawning scripting engines such as PowerShell.
- Apply attack surface reduction rules to limit abuse of living-off-the-land binaries.

Microsoft Defender XDR coverage

Microsoft Defender XDR provides coordinated detection and investigation across endpoints, identities, email, and cloud applications. Relevant protections include:

- Detection of suspicious PowerShell execution chains
- Network-based indicators tied to known malicious infrastructure.
- Behavioral monitoring of browser process injection
- Cross-domain correlation to identify installer abuse patterns.

Customers are encouraged to review applicable detections and hunting guidance within Microsoft Defender XDR to proactively identify similar activity.

Microsoft Security Copilot

Security Copilot customers can use the standalone experience to [create their own prompts](#) or run the following [prebuilt promptbooks](#) to automate incident response or investigation tasks related to this threat:

- Incident investigation
- Microsoft User analysis
- Threat actor profile
- Threat Intelligence 360 report based on MDTI article
- Vulnerability impact assessment

Advance Hunting queries - Microsoft Defender XDR

Microsoft Defender XDR customers can run the following query to find related activity in their networks:

1) Detects malicious MSI downloads originating from WordPress upload paths or matching known hashes.

```
DeviceFileEvents
| where FileName endswith ".msi"
| where FileOriginUrl has_any ("/wp-content/uploads/", "/uploads/MSI/", "emeditor-core")
  or SHA256 in
("4bea333d3d2f2a32018cd6afe742c3b25bfcc6bfe8963179dad3940305b13c98", "3d1763b037e66bbde222125a21b23fc
24abd76ebab40589748ac69e2f37c27fc")
| project Timestamp, DeviceName, FileName, FolderPath, FileOriginUrl, InitiatingProcessFileName,
InitiatingProcessCommandLine, SHA256 | order by Timestamp desc
```

2) Correlate PowerShell stager with C2 infrastructure

```
DeviceProcessEvents
| where FileName =~ "powershell.exe"
| where ProcessCommandLine has_all ("irm", "iex")
| join kind=inner ( DeviceNetworkEvents | where RemoteUrl has_any ( "cachingdrive.com",
"emeditorde.com", "emeditorgb.com", "emeditorjp.com", "emeditorsb.com" ) ) on DeviceId
| project Timestamp, DeviceName, ProcessCommandLine, InitiatingProcessFileName, RemoteUrl,
RemoteIP, Protocol
| order by Timestamp desc
```

Indicator Of Compromise:

Indicator	Type	Description
4bea333d3d2f2a32018cd6afe742c3b25bfcc6bfe8963179dad3940305b13c98	File hash (SHA-256)	Trojanized EmEditor MSI installer delivered via compromised WordPress infrastructure

3d1763b037e66bbde222125a21b23fc24abd76ebab40589748ac69e2f37c27fc	File hash (SHA-256)	Secondary malicious MSI variant associated with the same campaign
hxxps://cachingdrive[.]com	Domain	Second-stage payload delivery infrastructure
hxxps://emeditorde[.]com	Domain	Stager and command-and-control infrastructure (mirror)
hxxps://emeditorgb[.]com	Domain	Stager and command-and-control infrastructure (regional variant)
hxxps://emeditorjp[.]com	Domain	Primary stager and command-and-control endpoint
hxxps://emeditorsb[.]com	Domain	Stager and command-and-control infrastructure (regional variant)
147.45.50[.]54	IP address	Hosting IP associated with cachingdrive[.]com
46.28.70[.]245	IP address	Hosting IP associated with emeditorde[.]com
5.101.82[.]159	IP address	Hosting IP associated with emeditorgb[.]com
5.101.82[.]118	IP address	Hosting IP associated with emeditorjp[.]com

Microsoft Sentinel

Microsoft Sentinel customers can use the TI Mapping analytics (a series of analytics all prefixed with 'TI maps') to automatically match the malicious domain indicators mentioned in this blog post with data in their workspace. If the TI Map analytics are not currently deployed, customers can install the Threat Intelligence solution from the [Microsoft Sentinel Content Hub](#) to have the analytics rule deployed in their Sentinel workspace.

References

[\[Important\] Follow-up: Security Incident Notice Regarding the EmEditor Installer Download Link – EmEditor \(Text Editor\)](#)

Learn more

For the latest security research from the Microsoft Threat Intelligence community, check out the [Microsoft Threat Intelligence Blog](#).

To hear stories and insights from the Microsoft Threat Intelligence community about the ever-evolving threat landscape, listen to the [Microsoft Threat Intelligence podcast](#).

Updated Mar 06, 2026

Version 1.0

[defender experts for xdr](#)



[View Profile](#)

[Microsoft Security Experts Blog](#)

Follow this blog board to get notified when there's new activity

Enjoying the article? Sign in to share your thoughts.

- © Microsoft 2026

Attackers compromised the upstream distribution mechanism for **EmEditor**, a widely used Windows text editor. Instead of delivering malware through phishing or malicious domains, the attackers manipulated server-side logic on the official download site to selectively serve a trojanized installer to public users while preserving legitimate content for administrators.

This campaign highlights two recurring challenges in defending modern environments:

- \n
- **Upstream trust abuse:** Malicious payloads delivered from legitimate, trusted domains.
- \n
- **Selective evasion:** Conditional logic designed to evade validation, monitoring, and routine testing.
- \n

Attackers increasingly favor techniques that “live off trust” rather than exploit obvious weaknesses. As organizations harden email gateways, enforce attachment scanning, and restrict macro execution, supply-chain compromises provide an attractive alternative path to initial access.

In this case, the attack required no user interaction beyond installing trusted software and relied entirely on legitimate operating system components for execution. This combination significantly reduced detection opportunities and increased the likelihood of successful compromise.

This is not a traditional malware delivery campaign. The distinguishing characteristics include:

\n

- Server-side conditional manipulation rather than client-side redirection

\n

- Weaponization of a legitimate MSI installer

\n

- Use of Windows Installer custom actions to execute in-memory payloads

\n

- Credential theft via named pipe injection without dropping additional executables.

\n

The investigation demonstrates how endpoint, network, and installer telemetry must be correlated to uncover attacks that intentionally blur the line between legitimate and malicious activity.

Server-side conditional tampering enabling selective MSI delivery.

Attackers compromised the software distribution pipeline to selectively serve a trojanized MSI installer to public users while preserving legitimate behavior for administrators. The malicious installer abused Windows Installer execution, in-memory PowerShell staging, and command-and-control infrastructure to enable credential access.

The activity was identified through proactive threat hunting across Microsoft Defender telemetry, focusing on anomalous installer behavior and unexpected PowerShell execution chains originating from trusted software installs.

\n

- PowerShell execution spawned from msixexec.exe

\n

- Network connections from installer-initiated processes to suspicious domains.

\n

- Browser process injection without corresponding file creation events

\n

Together, these indicators pointed to a compromised installer rather than a post-installation infection vector.

The initial compromise occurred on a public-facing WordPress environment associated with the EmEditor download infrastructure. Attackers likely gained access via a vulnerable plugin or exposed administrative interface and deployed a web shell to maintain persistence.

Rather than modifying core WordPress files or defacing the site, the attackers injected **conditional PHP logic** into a theme-level file (footer.php). This logic dynamically altered download behaviour based on visitor context:

\n

- **Authenticated administrators** were served the legitimate EmEditor MSI.

\n

- **Unauthenticated public visitors** were redirected to a trojanized MSI hosted under /wp-content/uploads/.

\n

This split-view evasion technique allowed attackers to weaponize the official domain while avoiding detection by internal validation workflows, routine administrative testing, and automated integrity checks.

The malicious installer closely resembled the legitimate EmEditor MSI in name and functionality but embedded a **custom action** that executed during installation.

\n

- Execution via msixexec.exe -Embedding

\n

- Silent spawning of powershell.exe

\n

- In-memory execution using Invoke-RestMethod piped to Invoke-Expression

\n

The MSI was digitally signed, but not by the legitimate Emurasoft certificate. Instead, it used a certificate issued to a non-trusted publisher that nonetheless reduced user suspicion.

During installation, Windows cached the MSI in C:\Windows\Installer\, enabling silent re-execution and complicating forensic reconstruction.

The PowerShell stager connected to attacker-controlled infrastructure using multiple fallback paths to ensure reliability:

\n

- **Primary endpoint:** emeditorjp[.]com

\n

- **Mirror endpoint:** emeditorde[.]com, emeditorgb[.]com, emeditorsb[.]com

\n

- **Second-stage delivery:** cachingdrive[.]com

\n

Connections were observed over HTTP, HTTPS, and TCP, indicating deliberate redundancy. This infrastructure delivered a second-stage payload designed to operate entirely in memory.

The second-stage payload targeted browser processes, including chrome.exe and msedge.exe, using **named pipe injection** techniques.

By injecting directly into existing browser processes, the malware avoided creating new processes or dropping additional files. This enabled access to:

\n

- Browser-stored credentials

\n

- Authentication cookies

\n

- Active session tokens for web and enterprise services

\n

The absence of obvious malware artifacts strongly suggests that credential theft and session hijacking were the primary objectives.

\n

- **Potential targets:** Enterprises and individual users installing EmEditor during the affected window.

\n

- **Industries:** Broad, including technology, professional services, and regulated sectors

\n

- **Impact:** Credential compromise, session hijacking, potential lateral movement

\n

- **Scope:** Limited in time but high impact due to trusted distribution channel

\n

Organizations that suspect exposure should take immediate steps to contain potential compromise:

\n

- Isolate affected endpoints from the network

\n

- Block known malicious domains and IP addresses at DNS and firewall layers.

\n

- Force credential resets for users on impacted systems.

\n

- Review active browser sessions and revoke tokens where possible.

\n

- Conduct full endpoint scans using Microsoft Defender XDR

\n

To reduce exposure to supply-chain attacks of this nature, organizations should consider the following measures:

\n

- Enforce multi-factor authentication across cloud and enterprise services.

\n

- Limit browser-stored credentials and encourage password managers with strong protections.

\n

- Monitor software installation activity for anomalous child process behavior.

\n

\n

- Enforce stricter code-signing validation policies.

\n

- Monitor msixexec.exe spawning scripting engines such as PowerShell.

\n

- Apply attack surface reduction rules to limit abuse of living-off-the-land binaries.

\n

Microsoft Defender XDR provides coordinated detection and investigation across endpoints, identities, email, and cloud applications. Relevant protections include:

- Detection of suspicious PowerShell execution chains
- Network-based indicators tied to known malicious infrastructure.
- Behavioral monitoring of browser process injection
- Cross-domain correlation to identify installer abuse patterns.

Customers are encouraged to review applicable detections and hunting guidance within Microsoft Defender XDR to proactively identify similar activity.

Security Copilot customers can use the standalone experience to [create their own prompts](#) or run the following [prebuilt promptbooks](#) to automate incident response or investigation tasks related to this threat:

Microsoft Defender XDR customers can run the following query to find related activity in their networks:

1) Detects malicious MSI downloads originating from WordPress upload paths or matching known hashes.

Indicator	Type	Description
4bea333d3d2f2a32018cd6afe742c3b25bfcc6bfe8963179dad3940305b13c98	File hash (SHA-256)	Trojanized EmEditor MSI installer delivered via compromised WordPress infrastructure
3d1763b037e66bbde222125a21b23fc24abd76ebab40589748ac69e2f37c27fc	File hash (SHA-256)	Secondary malicious MSI variant associated with the same campaign
hxxps://cachingdrive[.]com	Domain	Second-stage payload delivery infrastructure

\n hxxps://emeditorde[.]com \n	\n Domain \n	\n Stager and command-and-control infrastructure (mirror) \n
\n hxxps://emeditorgb[.]com \n	\n Domain \n	\n Stager and command-and-control infrastructure (regional variant) \n
\n hxxps://emeditorjp[.]com \n	\n Domain \n	\n Primary stager and command-and-control endpoint \n
\n hxxps://emeditorsb[.]com \n	\n Domain \n	\n Stager and command-and-control infrastructure (regional variant) \n
\n 147.45.50[.]54 \n	\n IP address \n	\n Hosting IP associated with cachingdrive[.]com \n
\n 46.28.70[.]245 \n	\n IP address \n	\n Hosting IP associated with emeditorde[.]com \n
\n 5.101.82[.]159 \n	\n IP address \n	\n Hosting IP associated with emeditorgb[.]com \n

\n 5.101.82[.]118 \n	\n IP address \n	\n Hosting IP associated with emeditorjp[.]com \n
----------------------------	---------------------------	---

Microsoft Sentinel customers can use the TI Mapping analytics (a series of analytics all prefixed with 'TI maps') to automatically match the malicious domain indicators mentioned in this blog post with data in their workspace. If the TI Map analytics are not currently deployed, customers can install the Threat Intelligence solution from the [Microsoft Sentinel Content Hub](#) to have the analytics rule deployed in their Sentinel workspace.

To hear stories and insights from the Microsoft Threat Intelligence community about the ever-evolving threat landscape, listen to the [Microsoft Threat Intelligence podcast](#).

Attackers compromised the upstream distribution mechanism for **EmEditor**, a widely used Windows text editor. Instead of delivering malware through phishing or malicious domains, the attackers manipulated server-side logic on the official download site to selectively serve a trojanized installer to public users while preserving legitimate content for administrators.

This campaign highlights two recurring challenges in defending modern environments:

- **Upstream trust abuse:** Malicious payloads delivered from legitimate, trusted domains.
- **Selective evasion:** Conditional logic designed to evade validation, monitoring, and routine testing.

Attackers increasingly favor techniques that "live off trust" rather than exploit obvious weaknesses. As organizations harden email gateways, enforce attachment scanning, and restrict macro execution, supply-chain compromises provide an attractive alternative path to initial access.

In this case, the attack required no user interaction beyond installing trusted software and relied entirely on legitimate operating system components for execution. This combination significantly reduced detection opportunities and increased the likelihood of successful compromise.

This is not a traditional malware delivery campaign. The distinguishing characteristics include:

- Server-side conditional manipulation rather than client-side redirection
- Weaponization of a legitimate MSI installer
- Use of Windows Installer custom actions to execute in-memory payloads
- Credential theft via named pipe injection without dropping additional executables.

The investigation demonstrates how endpoint, network, and installer telemetry must be correlated to uncover attacks that intentionally blur the line between legitimate and malicious activity.

Server-side conditional tampering enabling selective MSI delivery.

Attackers compromised the software distribution pipeline to selectively serve a trojanized MSI installer to public users while preserving legitimate behavior for administrators. The malicious installer abused Windows Installer execution, in-memory PowerShell staging, and command-and-control infrastructure to enable credential access.

The activity was identified through proactive threat hunting across Microsoft Defender telemetry, focusing on anomalous installer behavior and unexpected PowerShell execution chains originating from trusted software installs.

\n

- PowerShell execution spawned from msixexec.exe

\n

- Network connections from installer-initiated processes to suspicious domains.

\n

- Browser process injection without corresponding file creation events

\n

Together, these indicators pointed to a compromised installer rather than a post-installation infection vector.

The initial compromise occurred on a public-facing WordPress environment associated with the EmEditor download infrastructure. Attackers likely gained access via a vulnerable plugin or exposed administrative interface and deployed a web shell to maintain persistence.

Rather than modifying core WordPress files or defacing the site, the attackers injected **conditional PHP logic** into a theme-level file (footer.php). This logic dynamically altered download behaviour based on visitor context:

\n

- **Authenticated administrators** were served the legitimate EmEditor MSI.

\n

- **Unauthenticated public visitors** were redirected to a trojanized MSI hosted under /wp-content/uploads/.

\n

This split-view evasion technique allowed attackers to weaponize the official domain while avoiding detection by internal validation workflows, routine administrative testing, and automated integrity checks.

The malicious installer closely resembled the legitimate EmEditor MSI in name and functionality but embedded a **custom action** that executed during installation.

\n

- Execution via msixexec.exe -Embedding

\n

- Silent spawning of powershell.exe

\n

- In-memory execution using Invoke-RestMethod piped to Invoke-Expression

\n

The MSI was digitally signed, but not by the legitimate Emurasoft certificate. Instead, it used a certificate issued to a non-trusted publisher that nonetheless reduced user suspicion.

During installation, Windows cached the MSI in C:\Windows\Installer\, enabling silent re-execution and complicating forensic reconstruction.

The PowerShell stager connected to attacker-controlled infrastructure using multiple fallback paths to ensure reliability:

\n

- **Primary endpoint:** emeditorjp[.]com

\n

- **Mirror endpoint:** emeditorde[.]com, emeditorgb[.]com, emeditorsb[.]com

\n

- **Second-stage delivery:** cachingdrive[.]com

\n

Connections were observed over HTTP, HTTPS, and TCP, indicating deliberate redundancy. This infrastructure delivered a second-stage payload designed to operate entirely in memory.

The second-stage payload targeted browser processes, including chrome.exe and msedge.exe, using **named pipe injection** techniques.

By injecting directly into existing browser processes, the malware avoided creating new processes or dropping additional files. This enabled access to:

\n

- Browser-stored credentials

\n

- Authentication cookies

\n

- Active session tokens for web and enterprise services

\n

The absence of obvious malware artifacts strongly suggests that credential theft and session hijacking were the primary objectives.

\n

- **Potential targets:** Enterprises and individual users installing EmEditor during the affected window.

\n

- **Industries:** Broad, including technology, professional services, and regulated sectors

\n

- **Impact:** Credential compromise, session hijacking, potential lateral movement

\n

- **Scope:** Limited in time but high impact due to trusted distribution channel

\n

Organizations that suspect exposure should take immediate steps to contain potential compromise:

\n

- Isolate affected endpoints from the network

\n

- Block known malicious domains and IP addresses at DNS and firewall layers.

\n

- Force credential resets for users on impacted systems.

\n

- Review active browser sessions and revoke tokens where possible.

\n

- Conduct full endpoint scans using Microsoft Defender XDR

\n

To reduce exposure to supply-chain attacks of this nature, organizations should consider the following measures:

\n

- Enforce multi-factor authentication across cloud and enterprise services.

\n

- Limit browser-stored credentials and encourage password managers with strong protections.

\n

- Monitor software installation activity for anomalous child process behavior.

\n

\n

- Enforce stricter code-signing validation policies.

\n

- Monitor msixexec.exe spawning scripting engines such as PowerShell.

\n

- Apply attack surface reduction rules to limit abuse of living-off-the-land binaries.

\n

Microsoft Defender XDR provides coordinated detection and investigation across endpoints, identities, email, and cloud applications. Relevant protections include:

\n

- Detection of suspicious PowerShell execution chains

\n

- Network-based indicators tied to known malicious infrastructure.

\n

- Behavioral monitoring of browser process injection

\n

- Cross-domain correlation to identify installer abuse patterns.

\n

Customers are encouraged to review applicable detections and hunting guidance within Microsoft Defender XDR to proactively identify similar activity.

Security Copilot customers can use the standalone experience to [create their own prompts](#) or run the following [prebuilt promptbooks](#) to automate incident response or investigation tasks related to this threat:

Microsoft Defender XDR customers can run the following query to find related activity in their networks:

1) Detects malicious MSI downloads originating from WordPress upload paths or matching known hashes.

\n

Indicator

\n\n

Type

\n\n

Description

\n\n

4bea333d3d2f2a32018cd6afe742c3b25bfcc6bfe8963179dad3940305b13c98

\n\n

File hash (SHA-256)

\n\n

Trojanized EmEditor MSI installer delivered via compromised WordPress infrastructure

\n\n

3d1763b037e66bbde222125a21b23fc24abd76ebab40589748ac69e2f37c27fc

\n\n

File hash (SHA-256)

\n\n

Secondary malicious MSI variant associated with the same campaign

\n\n

hxxps://cachingdrive[.]com

\n\n

Domain

\n\n

Second-stage payload delivery infrastructure

\n\n

hxxps://emeditorde[.]com

\n\n

Domain

\n\n

Stager and command-and-control infrastructure (mirror)

\n\n

hxxps://emeditorgb[.]com

\n\n

Domain

\n\n

Stager and command-and-control infrastructure (regional variant)

\n\n
hxxps://emeditorjp[.]com

\n\n
Domain

\n\n
Primary stager and command-and-control endpoint

\n\n
hxxps://emeditorsb[.]com

\n\n
Domain

\n\n
Stager and command-and-control infrastructure (regional variant)

\n\n
147.45.50[.]54

\n\n
IP address

\n\n
Hosting IP associated with cachingdrive[.]com

\n\n
46.28.70[.]245

\n\n
IP address

\n\n
Hosting IP associated with emeditorde[.]com

\n\n
5.101.82[.]159

\n\n
IP address

\n\n
Hosting IP associated with emeditorgb[.]com

\n\n
5.101.82[.]118

\n\n
IP address

\n\n

Hosting IP associated with emeditorjp[.]com

\n

Microsoft Sentinel customers can use the TI Mapping analytics (a series of analytics all prefixed with 'TI maps) to automatically match the malicious domain indicators mentioned in this blog post with data in their workspace. If the TI Map analytics are not currently deployed, customers can install the Threat Intelligence solution from the [Microsoft Sentinel Content Hub](#) to have the analytics rule deployed in their Sentinel workspace.

To hear stories and insights from the Microsoft Threat Intelligence community about the ever-evolving threat landscape, listen to the [Microsoft Threat Intelligence podcast](#).