

# Lazarus Threat Group Attacking Windows Servers to Use as Malware Distribution Points - ASEC

By ATCP

Published: 2023-07-13 · Archived: 2026-04-05 17:34:27 UTC

AhnLab Security Emergency response Center (ASEC) has discovered that Lazarus, a threat group deemed to be nationally funded, is attacking Windows Internet Information Service (IIS) web servers and using them as distribution points for their malware.

The group is known to use the watering hole technique for initial access. [1] The group first hacks Korean websites and modifies the content provided from the site. When a system using a vulnerable version of INISAFE CrossWeb EX V6 visits this website via a web browser, the Lazarus malware (SCSKAppLink.dll) is installed from the distribution site through the INISAFECrossWebEXSvc.exe vulnerability.

While the INITECH vulnerability has already been patched, **vulnerability attacks against systems that have not yet been patched** still continue to this day. After the Lazarus group attacks an IIS web server and obtains control, it will use the server to distribute malware used for INITECH vulnerability attacks. If a system has a vulnerable version of INISAFE CrossWeb EX V3 installed on it, **it must be uninstalled and updated to the latest version** following the security update recommendation below.

- [Initech Product \(INISAFE CrossWEB\) Security Update Recommendation](#)

## 1. Attacks Against Windows IIS Web Servers

Cases of the Lazarus threat group targeting IIS servers had also been covered in the past blog post (May 2023), “Lazarus Group Targeting Windows IIS Web Servers” [2]. It was identified in the attack case at the time that the threat actor used poorly managed or vulnerable web servers as the initial access point. There were also circumstances of RDP being used for lateral movement after the internal reconnaissance process.

Ordinarily, when attackers find a web server with a vulnerable version from scanning, they use the vulnerability suitable for the version to install a WebShell or execute malicious commands. When the threat actor exploits the vulnerability to execute malicious commands or uses WebShell to download/upload files and execute remote commands, the malicious behaviors are performed by w3wp.exe that is the IIS web server process.

The recently identified attack showed that the Lazarus threat group’s malware strains were generated by w3wp.exe (IIS web server process), similar to past cases.

Target Type	File Name	File Size	File Path
Current	cmd.exe	198 KB	%SystemRoot%\system32\cmd.exe
Target	usopriv.exe	3.25 MB	%ALLUSERSPROFILE%\usopriv.exe
Parent	w3wp.exe	21 KB	%SystemRoot%\system32\inetsrv\w3wp.exe
ParentOfParentOfCurrent	svchost.exe	45.38 KB	%SystemRoot%\system32\svchost.exe

  

Process	Module	Target	Behavior
cmd.exe	N/A	usopriv.exe	Creates process
cmd.exe	N/A	svchost.exe	Creates process

## 2. Privilege Escalation Malware JuicyPotato (usopriv.exe)

The malware generated by the w3wp.exe process, usopriv.exe is the JuicyPotato malware packed with Themida. The Potato malware strains are responsible for privilege escalation. There are many types such as JuicyPotato, RottenPotato, and SweetPotato according to the privilege escalation method.

```
C:\#ProgramData>usopriv.exe

JuicyPotatoNG
by decoder_it & splinter_code

JuicyPotatoNG
by decoder_it & splinter_code

Mandatory args:
-t createprocess call: <t> CreateProcessWithTokenW, <u> CreateProcessAsUser, <*> try both
-p <program>: program to launch

Optional args:
-l <port>: COM server listen port (Default 10247)
-a <argument>: command line argument to pass to program (default NULL)
-c <CLSID>: (Default {854A20FB-2D44-457D-992F-EF13785D2B51})
-i : Interactive Console (valid only with CreateProcessAsUser)

Additional modes:
-b : Bruteforce all CLSIDs. !ALERT: USE ONLY FOR TESTING. About 1000 processes will be spawned!
-s : Seek for a suitable COM port not filtered by Windows Defender Firewall
```

While threat actors can control the processes through WebShells or dictionary attacks, they cannot perform the intended malicious behaviors because the w3wp.exe process does not have the appropriate privilege. The case is the same for the sqlservr.exe process in MS-SQL servers. To resolve this problem, threat actors often simultaneously use privilege escalation tools in their attacks.

Particularly, the Potato strains of malware for privilege escalation are mainly used in attacks against IIS web servers and MS-SQL database servers. Potato types escalate privilege by abusing some processes with certain privileges activated. Afterward, the threat actor is able to perform malicious behaviors using the elevated privilege.

The following is a list of commands executed by the threat actor using JuicyPotato installed in infected systems. The whoami command was used to check if privilege escalation had occurred correctly. A log was also found showing that a loader malware which is responsible for the actual malicious behavior had been executed.

Time	Location	Command
2023-6-28 11:35 AM	%ALLUSERSPROFILE%\usopriv.exe	%SystemRoot%\system32\cmd.exe /c whoami > c:\programdata
2023-6-29 7:48 AM	%ALLUSERSPROFILE%\usopriv.exe	%SystemRoot%\system32\cmd.exe /c whoami > c:\programdata
2023-6-29 7:51 AM	%ALLUSERSPROFILE%\usopriv.exe	%SystemRoot%\system32\cmd.exe /c whoami > c:\programdata\nueio.txt
2023-6-29 8:27 AM	%ALLUSERSPROFILE%\usopriv.exe	%SystemRoot%\system32\cmd.exe /c rundll32 c:\programdata\usoshared.dat ,usoprivfunc 4729858204985024133
2023-6-29 8:40 AM	%ALLUSERSPROFILE%\usopriv.exe	%SystemRoot%\system32\cmd.exe /c del c:\programdata\nueio.txt
2023-6-29 3:08 PM	%USERPROFILE%\desktop\ngc\usopriv.exe	%SystemRoot%\system32\cmd.exe /c whoami > c:\users\%ASD%\desktop\ngc\test.txt

Table 1. List of commands executed through the privilege escalation malware

Collected Date	Process	Module	Behavior	Data
2023-06-29 08:29:22	rundll32.exe	usoshared.dat	Opens process	Target Process 
2023-06-29 08:27:26	rundll32.exe	usoshared.dat	Detected fileless attack	Target Process rundll32.exe
2023-06-28 11:35:46	usopriv.exe	N/A	Executes exploitable process	Target Process cmd.exe

### 3. Loader Malware (usoshared.dat)

The threat actor used JuicyPotato to execute a loader. The loader is in DLL format, so rundll32 was used to execute it. A random string was given as the argument.

```
> rundll32 c:\programdata\usoshared.dat ,usoprivfunc 4729858204985024133
```

First, the loader decrypts the file name of the data to be used and obtains the string “{20D1BF68-64EE-489D-9229-95FEFE5F12A4}”. This string is the name of the data file. Files with this name are searched for in a total of three paths. While the files in these paths have not been procured as of yet, it could be identified through the loader malware routine that this malware type is a loader that decrypts encrypted data files and executes them in the memory area.

- A folder containing rundll32.exe
- A folder containing usoshared.dat
- C:\Windows\Installer\

```

00007FF9D8483338 48:8D15 C95C0400 lea rdx,qword ptr ds:[7FF9D84C9008] 00007FF9D84C9008:"{PXZF}5sR-t]m0-o8My-t.W_-Ibc3L0
00007FF9D848333F 48:8D4D 50 lea rcx,qword ptr ss:[rbp+50] [rbp+50]:"{20D1BF68-64EE-489D-9229-95FEFE5F12A4}"
00007FF9D8483343 E8 D8DFFFFF call <usoshared.fn_decodestr>
RIP -> 00007FF9D8483348 90 nop
00007FF9D8483349 48:8D0D C8330200 lea rcx,qword ptr ds:[7FF9D84A6718] 00007FF9D84A6718:"tJZmFxA7ko3IkunmB1"
00007FF9D8483350 E8 6BDDFFFF call <usoshared.fn_getAPI>
00007FF9D8483355 41:88 01020000 mov r8d,201
<
rcx=FFFFFFFF
qword ptr [rbp+50]=[000000414787F5B0 &"{20D1BF68-64EE-489D-9229-95FEFE5F12A4}"]=0000017C1C33B070 "{20D1BF68-64EE-489D-9229-95FEFE5F12A4}"
.text:00007FF9D848333F usoshared.dat:$333F #273F
    
```

If the file {20D1BF68-64EE-489D-9229-95FEFE5F12A4} exists in the above path, the first 3 bytes are read to determine if it is the string “GIF”. It appears that the threat actor disguised the data file as a GIF image file. If the conditions match, the next 4 bytes are read. This contains the size of the data that will be read.

```

(ReadFile)(hFile, Buffer_1, 3i64, &lpNumberOfBytesRead, 0i64);
v17 = 0i64;
do
{
  buf = *(Buffer_1 + v17++);
  if ( buf != aGif[v17 - 1] )           // "GIF"
    goto LABEL_30;
}
while ( v17 != 4 );
ReadFile_1 = fn_getAPI("Mh5KE4Kx");
(ReadFile_1)(hFile, &Buffer_2, 4i64, &lpNumberOfBytesRead, 0i64);
if ( !Buffer_2 )
  goto LABEL_30;
if ( Buffer_2 > 0xC00 )                 // Checking Size
  goto LABEL_31;
Buffer_3 = LocalAlloc(0x40u, 0x1002ui64);
if ( !Buffer_3 )
  goto LABEL_31;
ReadFile_2 = fn_getAPI("Mh5KE4Kx");
(ReadFile_2)(hFile, Buffer_3, Buffer_2, &lpNumberOfBytesRead, 0i64);

```

Because the remaining data is executed in the memory area through the following decryption routine, it is deemed to be the actual encrypted PE. The first obtained data (starting with 0xC00) is given as an argument when executing PE in the memory area, and so is deemed to be the configuration data to be used by the decrypted malware.

```

kernel32 = fn_getModule(L"Kernel32.dll");
HeapAlloc = fn_getProc(kernel32, "HeapAlloc");
kernel32_1 = fn_getModule(L"Kernel32.dll");
GetProcessHeap = fn_getProc(kernel32_1, "GetProcessHeap");
user32 = fn_getModule(L"User32.dll");
fn_getProc(user32, "wsprintf");
hHeap = GetProcessHeap();
result = HeapAlloc(hHeap, 8i64, 32i64);
mem_newAlloc = result;
if ( result )
{
  *result = 0;
  if ( a1 == 3 )
  {
    result[4] = data_sizeOfPE;
    *(result + 1) = data_decodedPE;
  }
  else
  {
    data_decodedPE = *(result + 1);
  }
  if ( fn_checkPE(data_decodedPE) && fn_allocMem(mem_newAlloc) && fn_resolveAPI(mem_newAlloc) )
  {
    if ( fn_runMem(mem_newAlloc, data_config) )
      *mem_newAlloc = 1;
  }
}

```

Offset	Size	Data
0x0000	0x0003	Signature (GIF)
0x0003	0x0004	The size of the configuration data
0x0007	SizeOfConfig	encrypted configuration data

0x0007 + SizeOfConfig	Remainder	The size of the encrypted PE (0x04) and the encrypted PE itself
-----------------------	-----------	---

Table 2. Structure of the encrypted data file

Generally, the Lazarus group uses a loader malware and an encrypted data file together as shown above. As shown above, the process involves a loader in the PE format finding a data file in a certain path. The file will be run after it is decrypted in the memory area. While the data file has not been identified yet, examining past cases reveals that the ultimately executed malware strains are mostly downloaders that download additional malware types or backdoors that can receive commands from the threat actor to perform malicious behaviors.

#### 4. INISAFE Vulnerability Exploitation

According to AhnLab Smart Defense (ASD) logs, INISAFE vulnerability attacks against systems using unpatched past versions of INISAFECrossWebEX are continuously ongoing.

After these attacks, the threat actor attempted to install an additional malware “SCSKAppLink.dll” in the infected system through INISAFE vulnerability attacks. The download URL for “SCSKAppLink.dll” was identified as being the aforementioned IIS web server. This signifies that the threat actor attacked and gained control over IIS web servers before using these as servers for distributing malware.

Target Type	File Name	File Size	File Path
Target	skin[1].htm	3.69 MB	%SystemDrive%\users\%ASD%\appdata\local\microsoft\windows\inetcache\ie4m6vlllh\skin[1].htm
Current	inisafecrosswebexsvc.exe	2.99 MB	%ProgramFiles%(x86)\initech\inisafe web ex client\inisafecrosswebexsvc.exe
GeneratedByCurrent	skin[1].htm	3.69 MB	%SystemDrive%\users\%ASD%\appdata\local\microsoft\windows\inetcache\ie4m6vlllh\skin[1].htm

  

Process	Module	Target	Behavior	Data
inisafecrosswebexsvc.exe	N/A	N/A	Creates executable file	N/A
inisafecrosswebexsvc.exe	N/A	N/A	Connects to network	https://www. . . . .co.kr/

  

Target Type	File Name	File Size	File Path
Current	inisafecrosswebexsvc.exe	2.99 MB	%ProgramFiles%(x86)\initech\inisafe web ex client\inisafecrosswebexsvc.exe
Target	scskapplink.dll	3.69 MB	%SystemDrive%\users\%ASD%\libraries\scskapplink.dll

  

Process	Module	Target	Behavior
inisafecrosswebexsvc.exe	N/A	N/A	Creates executable file

The malware installed through exploiting this vulnerability (“SCSKAppLink.dll”) has not been identified, but it is probably similar to that covered in a previous ASEC Blog post, “New Malware of Lazarus Threat Actor Group Exploiting INITECH Process”[3]. “SCSKAppLink.dll” was identified in the past as being a downloader malware that downloads and executes additional malware strains from an external source. It can install malware types designated by the attacker in the system to gain control.

#### 5. Conclusion

The Lazarus group used various attack vectors for initial access such as [joint certificate vulnerabilities](#) and [3CX supply chain attacks](#). It is one of the most dangerous threat groups highly active worldwide. Thus, corporate security managers must practice strict management by employing attack surface management to identify assets that may be exposed to threat actors and continuously applying the latest security patches.

The threat actor is continuously using vulnerability attacks for initial access to unpatched systems. If a system does not have the latest version of INITECH products installed, the latest update must be applied following the security update recommendation below.

- [Initech Product \(INISAFE CrossWEB\) Security Update Recommendation](#)

Also, V3 should be updated to the latest version so that malware infection can be prevented.

#### **File Detection**

- Exploit/Win.JuicyPotato.C5452409 (2023.07.12.03)
- Trojan/Win.Loader.C5452411 (2023.07.12.03)

#### **Behavior Detection**

- InitialAccess/MDP.Event.M4242

MD5

280152df6b6d3123789138c0a396f30d

d0572a2dd4da042f1c64b542e24549d9

Additional IOCs are available on AhnLab TIP.

Gain access to related IOCs and detailed analysis by subscribing to **AhnLab TIP**. For subscription details, click the banner below.

