Lazarus Continues Heists, Mounts Attacks on Financial Organizations in Latin America

blog.trendmicro.com/trendlabs-security-intelligence/lazarus-continues-heists-mounts-attacks-on-financial-organizations-in-latin-america/
Trend Micro
November 20, 2018

0

by Lenart Bermejo and Joelson Soares

The cybercriminal group Lazarus, and particularly its subgroup Bluenoroff, has a history of attacking financial organizations in Asia and Latin America. There seems to be a resurgence of activity from the group, and recent events show how their tools and techniques have evolved. Just last week they were found stealing millions from ATMs across Asia and Africa. We also recently discovered that they successfully planted their backdoor (detected by Trend Micro as BKDR_BINLODR.ZNFJ-A) into several machines of financial institutions across Latin America.

We determined that these backdoors were installed on the targets' machines on September 19 2018, based mainly on the service creation time of the loader component. We also saw that the attack technique bears some resemblance to a previous 2017 Lazarus attack, analyzed by BAE Systems, against targets in Asia. The use of FileTokenBroker.dll was a key part of the group's attack in 2017, and they seem to have used the same modularized backdoor in the recent incident as well.

Our analysis of the backdoors used in the September 2018 attacks show that AuditCred.dll/ROptimizer.dll was similarly used:

	FileTokenBroker.dll (2017 attack)	**AuditCred.dll/Roptimizer.dll (2018 attack)**
Launch Method	Service	Service
Function	Loader Component	Loader Component
Working directory	%Windows%\System32	%Windows%\System32
Loaded Component Path	%Windows%\System32\en-US	%Program Files%\Common Files\System \ado
Loaded Component Blending	Blends with .mui files	Blend with ActiveX data Object dll files

Table1: Similarities of the Loader components in both incidents

Analysis of backdoors used in 2018

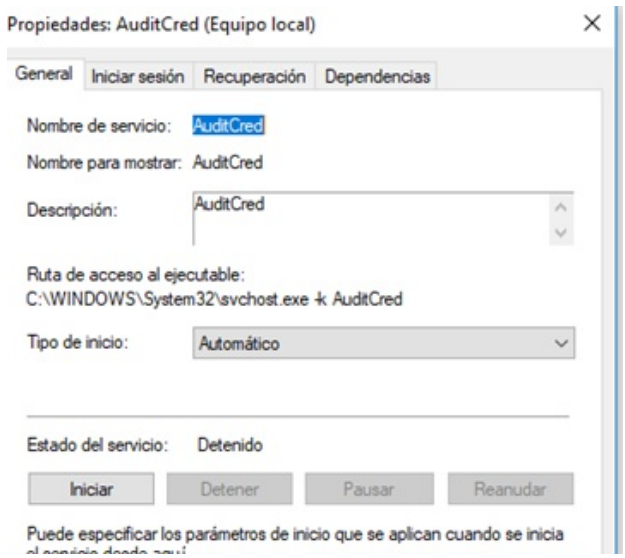
The Lazarus group used a series of backdoors in their 2018 attacks, employing a complicated technique that involves three major components:

- **AuditCred.dll/ROptimizer.dll** (detected by Trend Micro as BKDR_BINLODR.ZNFJ-A) – loader DLL that is launched as a service
- **Msadoz<*n*>.dll** (detected by Trend Micro as BKDR64_BINLODR.ZNFJ-A) – encrypted backdoor
  n = number of characters in the loader dll's filename
- **Auditcred.dll.mui/rOptimizer.dll.mui** (detected by Trend Micro as TROJ_BINLODRCONF.ZNFJ-A) – encrypted configuration file

Figure 1: Loading sequence of the modularized backdoor

The loader DLL is installed as a service and uses different names (AuditCred and ROptimizer) on different machines. However, they still have the same capabilities and are essentially the same file. Its purpose is to load **Msadoz<n>.dll** in order to decrypt and execute it in memory.

Figure 2: AuditCred/ROptimizer Service



If successfully installed, this particular backdoor poses quite a threat to its target. It is capable of the following functions:

- Collect file/folder/drive information
- Download files and additional malware
- Launch/terminate/enumerate process
- Update configuration data
- Delete files
- Inject code from files to other running process
- Utilize proxy

- Open reverse shell
- Run in passive mode — instead of actively connecting to the command and control (C&C) server, the backdoor will open and listen to a port then receive commands through it

Once the backdoor is loaded, it will then load the encrypted configuration file **Auditcred.dll.mui/rOptimizer.dll.mui** to extract the C&C information and connect to it. The connection is necessary for conducting activities; and based on the backdoor's functions, these actions could be quite damaging to targets.

```
0000000006113A00   74 1E              je 6113A20
0000000006113A02   4A 8D 14 20        lea rdx,qword ptr ds:[rax+r12]
0000000006113A06   66 66 0F 1F 84 00 00 0( nop word ptr ds:[rax+rax]
0000000006113A10   0F B6 42 FF        movzx eax,byte ptr ds:[rdx-1]
0000000006113A14   48 8D 52 FF        lea rdx,qword ptr ds:[rdx-1]
0000000006113A18   30 42 01           xor byte ptr ds:[rdx+1],al
0000000006113A1B   41 FF C8           dec r8d
0000000006113A1E   75 F0              jnz 6113A10
0000000006113A20   83 C3 E0           add ebx,FFFFFFE0
0000000006113A23   49 8D 54 24 20     lea rdx,qword ptr ds:[r12+20]
0000000006113A28   49 8B CC           mov rcx,r12
0000000006113A2B   44 8B C3           mov r8d,ebx
0000000006113A2E   E8 7D D8 FF FF     call 61112B0
```

Figure 3: The first step of decryption will perform XOR on one byte using the previous adjacent byte, starting from the last byte and excluding the first byte

```
0000000006111300   41 88 01           mov byte ptr ds:[r9],al
0000000006111303   66 FF C0           inc ax
0000000006111306   49 FF C1           inc r9
0000000006111309   66 41 3B C2        cmp ax,r10w
000000000611130D   72 F1              jb 6111300
000000000611130F   4C 8D 04 24        lea r8,qword ptr ss:[rsp]
0000000006111313   66 66 66 66 66 0F 1F 84 nop word ptr ds:[rax+rax]
0000000006111320   41 0F B6 10        movzx edx,byte ptr ds:[r8]
0000000006111324   40 0F B6 C7        movzx eax,dil
0000000006111328   40 FE C7           inc dil
000000000611132B   0F B6 0C 28        movzx ecx,byte ptr ds:[rax+rbp]
000000000611132F   02 CA              add cl,dl
0000000006111331   40 02 F1           add sil,cl
0000000006111334   40 80 FF 0C        cmp dil,C
0000000006111338   40 0F B6 CE        movzx ecx,sil
000000000611133C   0F B6 04 0C        movzx eax,byte ptr ss:[rsp+rcx]
0000000006111340   41 88 00           mov byte ptr ds:[r8],al
0000000006111343   40 0F B6 C7        movzx eax,dil
0000000006111347   88 14 0C           mov byte ptr ss:[rsp+rcx],dl
000000000611134A   41 0F 44 C4        cmove eax,r12d
000000000611134E   49 FF C0           inc r8
0000000006111351   49 FF CA           dec r10
0000000006111354   0F B6 F8           movzx edi,al
0000000006111357   75 C7              jnz 6111320
0000000006111359   44 0F B6 8C 24 00 01 0( movzx r9d,byte ptr ss:[rsp+100]
0000000006111362   44 0F B6 94 24 01 01 0( movzx r10d,byte ptr ss:[rsp+101]
000000000611136B   85 DB              test ebx,ebx
000000000611136D   74 39              je 61113A8
000000000611136F   90                 nop
0000000006111370   41 FE C1           inc r9b
0000000006111373   49 FF C3           inc r11
0000000006111376   45 0F B6 C1        movzx r8d,r9b
000000000611137A   42 0F B6 14 04     movzx edx,byte ptr ss:[rsp+r8]
000000000611137F   44 02 D2           add r10b,dl
0000000006111382   41 0F B6 CA        movzx ecx,r10b
0000000006111386   0F B6 04 0C        movzx eax,byte ptr ss:[rsp+rcx]
000000000611138A   42 88 04 04        mov byte ptr ss:[rsp+r8],al
000000000611138E   88 14 0C           mov byte ptr ss:[rsp+rcx],dl
0000000006111391   42 0F B6 0C 04     movzx ecx,byte ptr ss:[rsp+r8]
0000000006111396   03 CA              add ecx,edx
0000000006111398   0F B6 C1           movzx eax,cl
000000000611139B   0F B6 0C 04        movzx ecx,byte ptr ss:[rsp+rax]
000000000611139F   41 30 4B FF        xor byte ptr ds:[r11-1],cl
00000000061113A3   48 FF CB           dec rbx
00000000061113A6   75 C8              jnz 6111370
```

Figure 4: The second step uses RC4, using the first 0x20 bytes from the result of the first step as the RC4 key

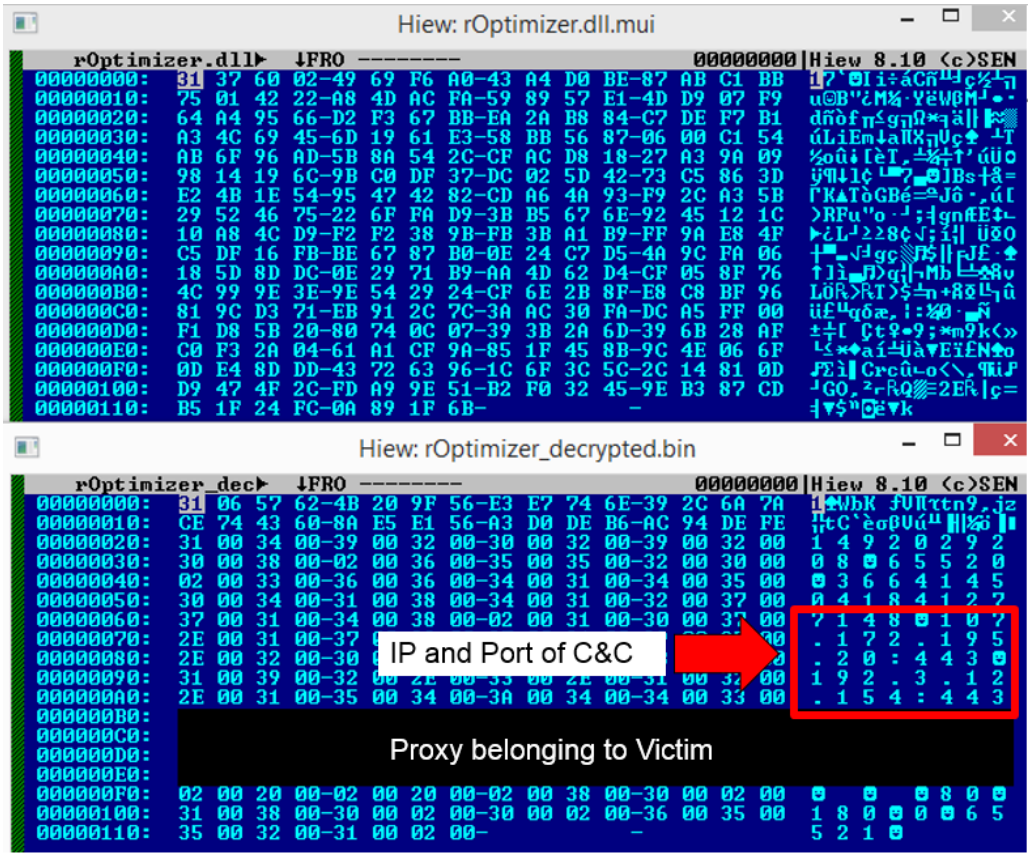


Figure 5: Encrypted (Top) and decrypted (bottom) configuration file

It is also important to note that while the loader component and the configuration file are located in the same directory (%windows%\system32), the encrypted backdoor is located in a different directory (%Program Files%\Common Files\System\ado). This complex setup makes it harder to detect and remove all the backdoors, and is more effective at hiding any activities.

The complexity and the capabilities of these backdoors present a tough problem for the targeted organizations. It is a sophisticated attack that needs equally sophisticated security solutions.

Trend Micro Solutions

The Lazarus group is an experienced organization, methodically evolving their tools and experimenting with strategies to get past an organization's defenses. The backdoors they are deploying are difficult to detect and a significant threat to the privacy and security of enterprises, allowing attackers to steal information, delete files, install malware, and more.

These and other tools used by the Lazarus group can be mitigated by routinely scanning the network for any malicious activity to help prevent the malware from entering and spreading through an organization. In addition, educating employees and other key people in an organization on social engineering techniques can allow them to identify what to look out for when it comes to malicious attacks.

Other mitigation strategies include a multilayered approach to securing the organization's perimeter, which includes hardening the endpoints and employing application control to help prevent malicious applications and processes from being executed.

Trend Micro endpoint solutions such as Trend Micro™ Smart Protection Suites and Worry-Free™ Business Security can protect users and businesses from these threats by detecting malicious files and spammed messages as well as blocking all related malicious URLs. Trend Micro Deep Discovery™ has an email inspection layer that can protect enterprises by detecting malicious attachments and URLs that could lead to malicious downloads.

Trend Micro XGen™ security provides a cross-generational blend of threat defense techniques to protect systems from all types of threats. It features high-fidelity machine learning on gateways and endpoints, and protects physical, virtual, and cloud workloads. With capabilities like web/URL filtering, behavioral analysis, and custom sandboxing, XGen security protects against today's threats that bypass traditional controls; exploit known, unknown, or undisclosed vulnerabilities; either steal or encrypt personally identifiable data; or conduct malicious cryptocurrency mining. Smart, optimized, and connected, XGen security powers Trend Micro's suite of security solutions: Hybrid Cloud Security, User Protection, and Network Defense.

Indicators of Compromise

Command and Control Servers
107[.]172[.]195[.]20
192[.]3[.]12[.]154
46[.]21[.]147[.]161