Lazarus KillDisks Central American casino

By Anton CherepanovPeter Kálnai

Archived: 2026-04-02 10:41:39 UTC

Our analysis shows that the cybercriminals behind the attack against an online casino in Central America, and several other targets in late-2017, were most likely the infamous Lazarus hacking group. In all of these incidents the attackers utilized similar toolsets, including KillDisk; the disk-wiping tool that was executed on compromised machines.

Lazarus toolset

The Lazarus Group was first identified in Novetta's report Operation Blockbuster in February 2016; US-CERT and the FBI call this group Hidden Cobra. These cybercriminals rose into prominence with the infamous case of cyber-sabotage against Sony Pictures Entertainment.

Some of the past attacks attributed to the Lazarus Group attracted the interest of security researchers who relied on Novetta et al's white papers with hundreds of pages describing the tools used in the attacks - the Polish and Mexican banks; the WannaCryptor outbreak; phishing campaigns against US defense contractors, etc – and provides grounds for the attribution of these attacks to the Lazarus Group.

Note that the Lazarus toolset (i.e. the collection of all files that are considered by the security industry as fingerprints of the group's activity) is quite broad, and we believe there are numerous subgroups. Unlike toolsets used by some other cybercriminal groups, none of the source code of any Lazarus tools has ever been disclosed in a public leak.

On top of custom tools, the Lazarus Group also leverages projects that are either available from GitHub or provided commercially.

Lazarus tools in casino attack

In this section, we review some of the tools that were detected on numerous servers and endpoints in the network of an online casino in Central America. These were used in conjunction with the destructive KillDisk samples described later. We explain why we believe they are linked to Lazarus. ESET detects known Lazarus malware as Win32/NukeSped or Win64/NukeSped.

Almost all of these tools are designed to run as a Windows service. Administrator privileges are necessary to achieve this, which means the attackers expected to have to have those privileges at the time of tool design or compilation.

TCP backdoor

Win64/NukeSped.W is a console application that is installed in the system as a service. One of the initial execution steps is dynamically resolving the required DLL names, on the stack:

```
mov     dword ptr [rbp+57h+LibFileName_0], 'nrek'
mov     [rbp+57h+var_64], '23le'
mov     [rbp+57h+var_60], 'lld.'          kernel32.dll
mov     [rbp+57h+var_5C], 0
mov     dword ptr [rbp+57h+LibFileName_1], 'avda'
mov     [rbp+57h+var_44], '23ip'
mov     [rbp+57h+var_40], 'lld.'          advapi32.dll
mov     [rbp+57h+var_3C], 0
mov     dword ptr [rbp+57h+LibFileName_2], 'lhpi'
mov     [rbp+57h+var_54], 'ipap'
mov     [rbp+57h+var_50], 'lld.'          iphlpapi.dll
mov     [rbp+57h+var_4C], 0
mov     dword ptr [rbp+57h+LibFileName_3], 'astw'
mov     [rbp+57h+var_34], '23ip'
mov     [rbp+57h+var_30], 'lld.'          wtsapi32.dll
mov     [rbp+57h+var_2C], 0
mov     dword ptr [rbp+57h+LibFileName_4], 'uces'
mov     [rbp+57h+var_84], '.23r'
mov     [rbp+57h+var_80], 'lld'           secur32.dll
mov     dword ptr [rbp+57h+LibFileName_5], 'ldtn'
mov     [rbp+57h+var_A4], 'ld.l'
mov     [rbp+57h+var_A0], 6Ch             ntdll.dll
mov     dword ptr [rbp+57h+LibFileName_6], 'resu'
mov     [rbp+57h+var_94], '.vne'
mov     [rbp+57h+var_90], 'lld'           userenv.dll
mov     dword ptr [rbp+57h+LibFileName_7], 'wlhs'
mov     [rbp+57h+var_74], '.ipa'
mov     [rbp+57h+var_70], 'lld'           shlwapi.dll
lea     rcx, [rbp+57h+LibFileName_0]    ; lpLibFileName
call    cs:LoadLibraryA
```

Likewise, procedure names of Windows APIs are constructed dynamically. In this particular sample, they are visible in plaintext; in other past samples that we've analyzed they were base64-encoded, encrypted or resolved on the stack character by character:

```
lea     rdx, aWtsenumeratese            ; "WTSEnumerateSessionsA"
mov     rcx, r13                        ; hModule
call    cs:GetProcAddress
mov     cs:WTSEnumerateSessionsA, rax
lea     rdx, aWtsfreememory             ; "WTSFreeMemory"
mov     rcx, r13                        ; hModule
call    cs:GetProcAddress
mov     cs:WTSFreeMemory, rax
lea     rdx, aWtsqueryuserto            ; "WTSQueryUserToken"
mov     rcx, r13                        ; hModule
call    cs:GetProcAddress
mov     cs:WTSQueryUserToken, rax

loc_13F6BA818:                          ; CODE XREF: resolve_WINAPIs+5E1↑j
test    r12, r12
jz      short loc_13F6BA862
lea     rdx, aLsaenumeratelo            ; "LsaEnumerateLogonSessions"
mov     rcx, r12                        ; hModule
call    cs:GetProcAddress
mov     cs:LsaEnumerateLogonSessions, rax
```

Both are typical traits of Lazarus malware. Another typical Lazarus backdoor characteristic is also seen in this backdoor: it listens on a specific port that it ensures is not blocked by the firewall:

```
1  void __fastcall TCP::PortOpening(__int64 a1, unsigned __int16 a2, int bSwitch)
2  {

13   l_bSwitch = bSwitch;
14   v4 = a2;
15   szCommand = 0;
16   memset(&Dst, 0, 0x103ui64);
17   szFmt = "netsh firewall add portopening TCP %d Assistance";
18   if ( !l_bSwitch )
19     szFmt = "netsh firewall delete portopening TCP %d";
20   sprintf(&szCommand, szFmt, v4);

28   if ( CreateProcessA_1(0i64, &szCommand, 0i64, 0i64, 0, 0, 0i64, 0i64, &Start
29   {
30     WaitForSingleObject(hHandle, 0x3A98u);
31     CloseHandle_1(hHandle);
32     CloseHandle_1(hObject);
33     Sleep(0x7D0u);
34   }
35 }
```

The backdoor supports 20 commands whose functionality is similar to previously analyzed Lazarus samples (note that the command names here did not originate from the attackers but were created by an ESET malware analyst):

```
switch ( CommandId )
{
  case 1:
    v10 = cmd__GetCurrentDir(MainObject, (__int64)Param1, (__int64)Param2, (__int64)Param3);
    goto LABEL_59;
  case 2:
    v10 = cmd__ListDisksInfo(MainObject, (__int64)Param1, (__int64)Param2, (__int64)Param3);
    goto LABEL_59;
  case 3:
    v10 = cmd__FileSearch(MainObject, (__int64)Param1, (__int64)Param2, (__int64)Param3);
    goto LABEL_59;
  case 4:
    v10 = cmd__CreateProcessSimple(MainObject, (__int64)Param1, (__int64)Param2, (__int64)Param3);
    goto LABEL_59;
  case 5:
    v10 = cmd__ChangeRealFileTime(MainObject, (__int64)Param1, (__int64)Param2, (__int64)Param3);
    goto LABEL_59;
  case 6:
    v10 = cmd__DropFile(MainObject, (__int64)Param1, (__int64)Param2, Param3);
    goto LABEL_59;
    :
  case 17:
    v10 = cmd__CreateProcessAsLoggedUser(MainObject, (__int64)Param1, (__int64)Param2, (__int64)Param3);
    goto LABEL_59;
  case 18:
    v10 = cmd__InjectIntoExplorer(MainObject, (__int64)Param1, (__int64)Param2, (__int64)Param3);
    goto LABEL_59;
  case 19:
    v10 = cmd__InjectIntoProcIDLoggedUser(MainObject, (__int64)Param1, (__int64)Param2, (__int64)Param3);
    goto LABEL_59;
}
if ( CommandId == 20 )
  break;
```

It creates several files on the file system. The listening port is stored in a text file named %WINDOWS%\Temp\p. The file %WINDOWS%\Temp\perflog.evt contains a list of paths of binary files to be injected, executed or written to the Registry depending upon the beginning character:

```
"*" = executed via process injection
"+" = executed via cmd.exe
" " =  written to the registry at
HKLM\SYSTEM\CurrentControlSet\Services\<ServiceName>\Instance
```

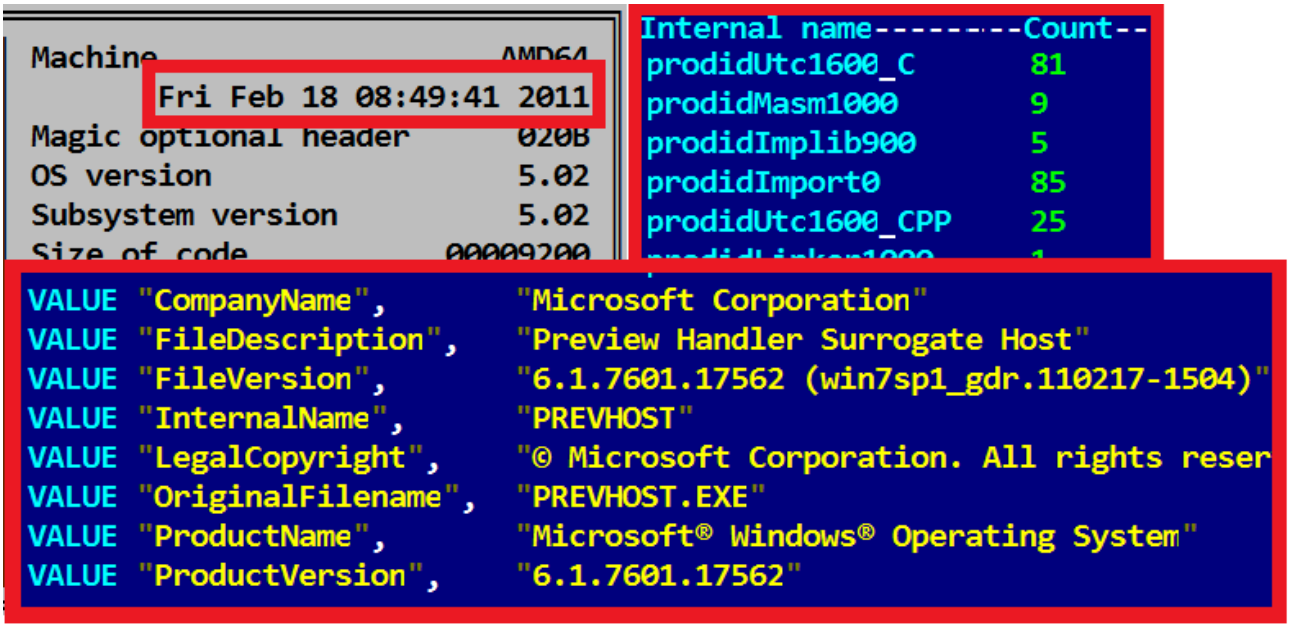
In case of the “+” option, the output data of cmd.exe /c "%s 2>> %s" (or cmd.exe /c "%s >> %s 2>&1") is logged to %WINDOWS%\Temp\perflog.dat.

Session hijacker

Win64/NukeSped.AB is a console application that creates a process as another currently–logged-in user on the victim’s system (similar to command number 17 from the previously described TCP backdoor).

It is a Themida-protected variant of a sample described by Kaspersky. In our case, it was installed as C:\Users\public\ps.exe. It accepts three parameters.

A static look shows the same file properties in both these samples: the same PE compilation timestamp, identical Rich Header linker data (indicating that the linker was Visual Studio 2010 (10.00)), and part of the resources version info matches:



While the PE timestamp and the resources are stolen from the legitimate Microsoft PREVHOST.EXE file from Windows 7 SP1, the linker data was not: the original Microsoft file was compiled and linked by Visual Studio 2008 (9.00).

Our consequent, dynamic analysis confirmed that this file – found in the compromised casino's network – is related to the session hijacker used in the Polish and Mexican attacks.

Loader/installer

This is a simple command line tool accepting several switches. Its purpose is to work with processes (injecting/killing a process by PID or by name), services (terminating/reinstalling a service) or files (drop/remove). Its exact functionality depends on the parameters.

KillDisk variants

KillDisk is a generic detection name that ESET uses for destructive malware with disk wiping capabilities, such as damaging boot sectors and overwriting then deleting (system) files, followed by a reboot to render the machine unusable. Although all KillDisk malware has similar functionality, as a generic detection, individual samples do not necessarily have strong code similarities or relationships. Such generic malware detections usually have many "sub-families", distinguished by the detection suffix (e.g. Win32/KillDisk.NBO in this case). Sub-family variants that do have strong code similarities, are sometimes seen separate cyberattacks and thus can help us make connections, as here. Other cases, for example the directed cyberattacks against high-value targets in Ukraine in December 2015 and December 2016, also employed KillDisk malware, but those samples were from different KillDisk sub-families, so are most likely unrelated to these attacks.

In the Central American online casino case, we detected two variants of Win32/KillDisk.NBO in their network. This malware was detected on over 100 machines in the organization. There are several possible explanations for its deployment, with the attackers covering their tracks after an espionage operation, or its direct use for extortion or sabotage, being the most probable. In any case, the impact against a single organization is large.

Based on our telemetry, the simultaneous use of the detected Win32/KillDisk.NBO variants, and the other known Lazarus malware on the targeted network, we are confident this KillDisk malware was deployed by Lazarus, rather than by another, unrelated attacker.

Our analysis of these two Win32/KillDisk.NBO variants revealed that they share many code similarities. Further, they are almost identical to the KillDisk variant used against financial organizations in Latin America, as described by Trend Micro.

In this online casino case, the KillDisk variants' path was typically: C:\Windows\Temp\dimens.exe

The actual embedded payload is injected into the system process werfault.exe:

```
1  BOOL WinMainEx()
2  {
15   if ( *(_WORD *)au8EmbeddedPayload == IMAGE_DOS_SIGNATURE )
16   {
17     do
18     {
19       v0 = &au8EmbeddedPayload[e_lfanew];
20       if ( *(_DWORD *)&au8EmbeddedPayload[e_lfanew] != IMAGE_NT_SIGNATURE )
21         break;
27       if ( !CreateProcessA( "C:\\Windows\\system32\\werfault.exe",
51       Mem = (char *)VirtualAllocEx(ProcessInformation.hProcess, *((LPVOID *)v0 + 13),
                                      *((_DWORD *)v0 + 20), 0x3000u, 0x40u);
58       if ( Mem )
59       {
60         WriteProcessMemory(ProcessInformation.hProcess, Mem, au8EmbeddedPayload,
81         SetThreadContext(ProcessInformation.hThread, v1);
82         ResumeThread(ProcessInformation.hThread);
88     }
```

One of the variants was protected using the commercial PE protector VMProtect in its 3rd generation, which made unpacking it trickier. The attackers most likely did not buy a VMProtect license but have rather used leaked or pirated copies available on the Internet. Using protectors is common for the Lazarus group: during the Polish and Mexican attacks in February 2017, they made use of Enigma Protector and some of the Operation Blockbuster samples, reported by Palo Alto Networks, used an older version of VMProtect.

Common Lazarus format strings

Among numerous typical characteristics that let us attribute the samples and attacks to Lazarus, one worth pointing out for the sake of other researchers is format strings. The table below lists formatting strings found in the aforementioned samples, as well as in many TCP backdoors linked with Lazarus:

Format String	Lazarus Attack / Report
cmd.exe /c "%s 2>> %s" cmd.exe /c "%s >> %s 2>&1"	This case - online casino in Central America
cm%sx%s"%s%s %s" 2>%s	Operation Blockbuster & WannaCryptor outbreak
c%s.e%sc "%s > %s 2>&1" %sd.e%sc "%s > %s 2>&1"	Operation Blockbuster - The Sequel
%s%s%s "%s > %s 2>&1" md.e xe /c	Operation Blockbuster - The Saga
%sd.e%sc "%s > %s" 2>&1 %sd.e%sc n%ssh%srewa%s ad%s po%sop%sing T%s %d "%s"	Operation Blockbuster
%s /c "%s" >%s 2>&1	Operation Blockbuster
cmd.exe /c "%s" > %s 2>&1	The Polish and Mexican case

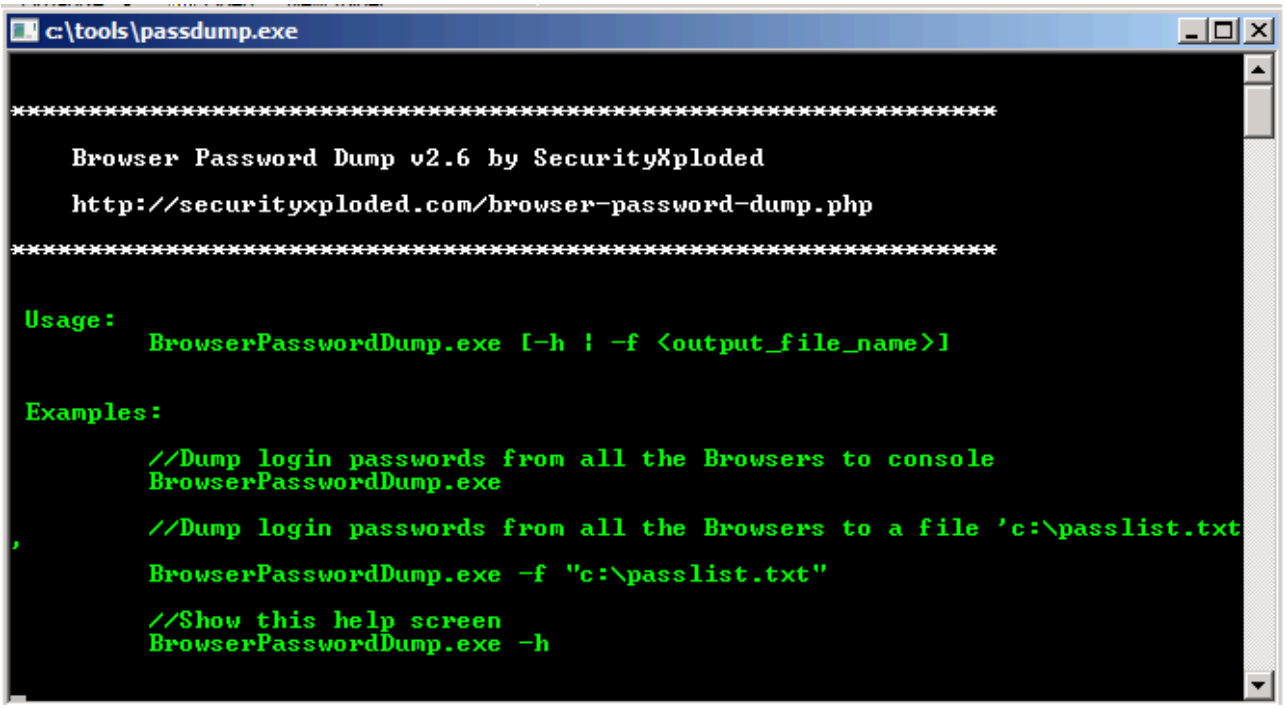
By itself, this might not seem to be a convincing clue, but checking these format strings against all the malware samples ever collected by ESET, the only results are from samples in suspected Lazarus operations. Hence, we conclude that these format strings represent a relevant, static characteristic of the Lazarus Group's modus operandi.

Additional tools

There are (at least) two widely available tools that the attackers in the online casino case also used.

Browser Password Dump

This shady tool serves the purpose of recovering passwords from popular web browsers. However, it is a tool from December 2014, which uses old, well-known techniques. Nevertheless, it can be used effectively on the current, latest versions of Google Chrome (64.0.3282.186), Chromium (67.0.3364.0), Microsoft Edge (41.16299.15.0) and Microsoft Internet Explorer (11.0.9600.17843). It does not work against recent versions of Firefox or Opera.



Mimikatz

These attackers also used a modified version of the infamous open-source tool Mimikatz, which is used for extracting Windows credentials. It accepts one parameter – the name of the file in which to store the output. If no parameter is given then the output file is called ~Temp1212.tmp located in the same directory as Mimikatz. The output contains hashes of the Windows credentials of currently logged-in users. This tool is commonly used by various APT groups and cybercriminals, for example by the Telebots group in the massive DiskCoder.C (aka NotPetya) outbreak, or in Operation Buhtrap.

```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
12   l_argc = argc;
13   l_argv = argv;
14   mimikatz_initOrClean(1);
15   if ( l_argc >= 2 )
16     g_logFileName = (char *)l_argv[1];
17   err = RtlAdjustPrivilege(SE_DEBUG_PRIVILEGE, TRUE, FALSE, &bEnabled);
18   if ( (err & 0x80000000) != 0 )
19     FS::writeFile(
20       (__int64)L"ERROR kuhl_m_privilege_simple ; RtlAdjustPrivilege (%u) %08x\n",
21       SE_DEBUG_PRIVILEGE,
22       err,
23       v7);
24   v10 = 4;
25   pOptionalData = g_pOptionalData;
26   kuhl_m_sekurlsa_enum(pData, (__int64)&pOptionalData);
27   (*(void (**)(void))(qword_13F2992F0 + 8))();
28   CoUninitialize();
29   return 0;
30 }
```

Infection vector

Most of the tools described above are downloaded and installed onto victim systems by malicious droppers and loaders active in the initial stage of the attack. Moreover, we have seen indicators that the attackers leveraged remote access tools, such as Radmin 3 and LogMeIn, in order to control machines remotely.

Conclusion

This recent attack against an online casino in Central America suggests that hacking tools from the Lazarus toolset are recompiled with every attack (we didn't see these exact samples anywhere else). The attack itself was very complex, consisted of several steps, and involved tens of protected tools that, being stand-alone, would reveal little from their dynamics.

Utilizing KillDisk in the attack scenario most likely served one of two purposes: the attackers covering their tracks after an espionage operation, or it was used directly for extortion or cyber-sabotage. In any case, the fact that ESET products detected the malware on over 100 endpoints and servers in the organization signifies a large-scale effort of the attackers.

Special thanks to Dávid Gábriš and Robert Lipovský.

Image Credit: © Julliane Nova

SAMPLES	**#colspan#**
429B750D7B1E3B8DFC2264B8143E97E5C32803FF	Win32/KillDisk.NBO
7DFE5F779E46855B32612D168B9CC5334F25B5F6	Win32/KillDisk.NBO
5042C16076AE6346AF8CF2B40553EEEEA98D5321	Win64/NukeSped.W trojan (VMProtect-ed)
7C55572E8573D08F3A69FB15B7FEF10DF1A8CB33	Win64/NukeSped.W trojan (Themida-protected)
E7FDEAB60AA4203EA0FF24506B3FC666FBFF759F	Win64/NukeSped.Z trojan (Themida-protected)
18EA298684308E50E3AE6BB66D7321A5CE664C8E	Win64/NukeSped.Z trojan (VMProtect-ed)
8826D4EDBB00F0A45C23567B16BEED2CE18B1B6A	Win64/NukeSped.AB trojan (Themida-protected)
325E27077B4A71E6946735D32224CA0421140EF4	Win64/Riskware.Mimikatz.A application

SAMPLES	**#colspan#**
D39311C74DEB60C736982C1AB74D6684DD1E1264	Win32/SecurityXploded.T (VMProtect-ed)
E4B763B4E74DE3EF24DB6F19108E70C494CD18C9	Win32/SecurityXploded.T (Themida-protected)
