(Are you) afreight of the dark? Watch out for Vyveva, new Lazarus backdoor

welivesecurity.com/2021/04/08/are-you-afreight-dark-watch-out-vyveva-new-lazarus-backdoor

April 8, 2021

ESET researchers have discovered a previously undocumented Lazarus backdoor, which they have dubbed Vyveva, being used to attack a freight logistics company in South Africa. The backdoor consists of multiple components and communicates with its C&C server via the Tor network. So far, we have been able to find its installer, loader and main payload – a backdoor with a TorSocket DLL. The previously unknown attack was discovered in June 2020.

Although Vyveva has been used since at least December 2018, its initial compromise vector is still unknown. Our telemetry data suggests targeted deployment as we found only two victim machines, both of which are servers owned by a freight logistics company located in South Africa. The backdoor features capabilities for file exfiltration, timestomping, gathering information about the victim computer and its drives, and other common backdoor functionality such as running arbitrary code specified by the malware's operators. This indicates that the intent of the operation is most likely espionage.

This blogpost provides the first public, technical analysis of Vyveva's components.

Attribution to Lazarus

Vyveva shares multiple code similarities with older Lazarus samples that are detected by ESET products as the NukeSped malware family. However, the similarities do not end there: the use of fake TLS in network communication, command line execution chains, and the way of using encryption and Tor services all point towards Lazarus; hence we can attribute Vyveva to this APT group with high confidence.

An example of the numerous code similarities can be seen in Figure 1 – resolving uniquely named Tor library exports.

- 92F5469DBEFDCEE1343934BE149AFC1241CC8497 msobjs.drx Vyveva backdoor
- BF98EA1326E5F8C351E68C79B5D1E0164C7BE728 taskhosts.exe Win32/NukeSped.HV trojan

```
ReadFile(v4, v6, v5, &NumberOfBytesRead, 0);
v9 = CloseHandle(v4);
if ( v9 != NumberOfBytesRead )
  v9 = NumberOfBytesRead;
if ( v12 != v9 )
  v12 = v9;
if ( v9 != v5 )
  return 0;
decrypt_PE_fix_MZ(v7, v7, v12);
v10 = pe_load_10001000(v7);
if ( v4 != v10 )
  v4 = v10;
if ( v4 )
{
  g_open_ch = find_export(v4, "open_ch");
  g_connect_ch = find_export(v4, "connect_ch");
  g_read_ch = find_export(v4, "read_ch");
  g_write_ch = find_export(v4, "write_ch");
  v11 = find_export(v4, "close_ch");
  g_close_ch = v11;
  if ( !g_open_ch || !g_connect_ch || !g_read_ch || !g_write_ch || !v11 )
    return 0;
  result = 1;
```

```
if ( v6 )
{
  ReadFile(v4, v6, v5, &NumberOfBytesRead, 0);
  CloseHandle(v4);
  sub_403B00(v14, 4096);
  v15 = 0;
  if ( sub_408C70(v7, v5) )
  {
    v8 = sub_403C80(v14);
    v9 = pe_load_403280(v8);
    v10 = v9;
    if ( v9 )
    {
      g_open_ch = find_export(v9, "open_ch");
      g_connect_ch = find_export(v10, "connect_ch");
      g_read_ch = find_export(v10, "read_ch");
      g_write_ch = find_export(v10, "write_ch");
      v11 = find_export(v10, "close_ch");
      g_close_ch = v11;
      if ( g_open_ch )
      {
        if ( g_connect_ch && g_read_ch && g_write_ch && v11 )
        {
```

Figure 1. Hex-Rays decompilation showing similarity between Vyveva (left) and NukeSped sample (right)

Technical analysis

Up until now, we have managed to find three of the multiple components comprising Vyveva – its installer, loader and backdoor. The installer is the earliest chronological stage found and since it expects other components to be already present on the machine, it suggests the existence of an earlier, unknown stage – a dropper. The loader serves to decrypt the backdoor using a simple XOR decryption algorithm.

Figure 2 provides a closer look at the functionality of the installer, the backdoor, and the Tor library.

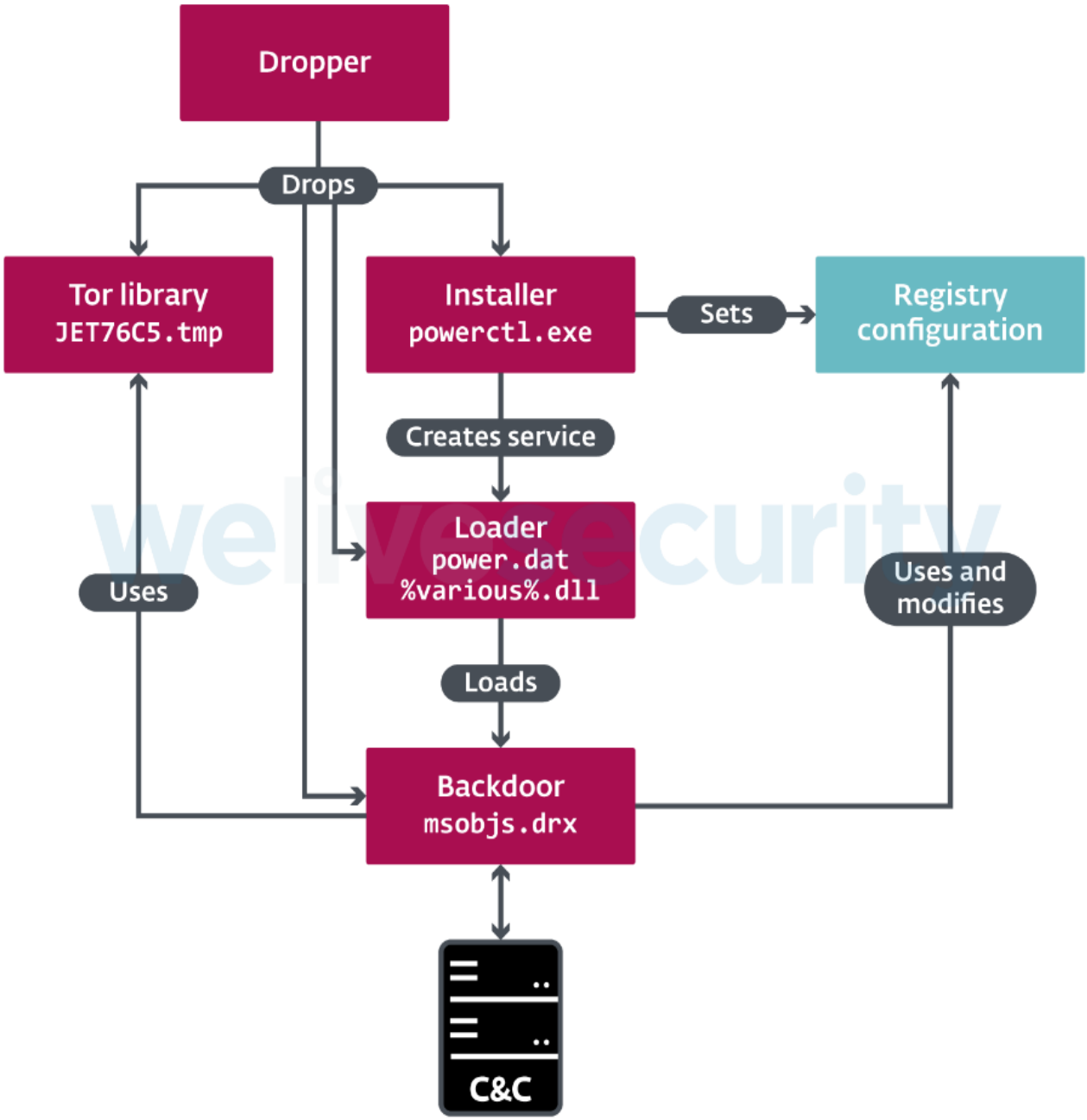

Figure 2. Overview of Vyveva components

Installer

The main purposes of the installer are twofold: it creates a service that ensures persistence of the backdoor loader, and it stores the embedded, default backdoor configuration in the registry.

To create a legitimate-looking service, its attributes, such as service name and display name, are formed using a combination of words from the attributes of existing services, which are randomly selected. It is also possible to specify these attributes to the installer via command line parameters -dll, -svc, -disp, -desc, and -group. We observed the following in the wild, with these parameters:

<SYSDIR>\powerctl.exe -svc powerctl -dll powerctl.dll

As for the latter task, the installer first sets the configuration infection ID, which uniquely identifies each victim, to a randomly generated value, and then stores it in the registry, as shown in Figure 3.

[HKLM\SOFTWARE\Microsoft\DirectX]
  UsageMask = <CONFIG_DATA>

Figure 3. Configuration registry value

One of the entries in the configuration is a list of encrypted C&C servers: for example, the installer sample we analyzed is configured with the following C&Cs:

- 4bjt2rceijktwedi[.]onion:80
- cwwpxpxusw07b6tr[.]onion:80

Backdoor functionality

The backdoor, Vyveva's main component, connects to C&C servers and executes commands issued by the threat actors. It features 23 commands, some of which are asynchronous and executed in their own threads. Most of them are ordinary commands for file and process operations or information gathering, but there is also a less common command for file timestomping. It can copy creation/write/access time metadata from a "donor" file to a destination file or use a random date in the years 2000—2004.

Other noteworthy commands are Vyveva's file upload command, and command 0x26. The file upload command is capable of exfiltrating directories recursively and supports file extension filtering – for example, Office documents only. As for command 0x26, it indicates the existence of another, unknown component that we have not yet observed at the time of writing.

The full list of commands is shown in Table 1.

Table 1. Vyveva backdoor commands

ID	Description
0x03	Reply to "ping" from server
0x10	Get information about computer – username, computer name, IP, code page, OS version, OS architecture, tick count, time zone, current directory
0x11	Get information about drives – type, size, name, serial number, filesystem type
0x12	Write data to specified file, optionally timestomp.

ID	Description
0x13	Upload specified file or directory • File – size, last write time, content • Directory stats – total files size, file count, directory count - For each entry – name, attributes - Directories – recurse into directories - Files – size, last write time, content Options • Use compression for file content (zlib 1.2.5) • File extension filter (whitelist/blacklist) • Recursion flag
0x14	Get listing of specified directory • name, attributes, write time • Directories – is nonempty • Files – size
0x15	Set current directory to specified directory
0x16	Create specified process
0x17	Get information about running processes – PID, PPID, executable file path
0x18	Terminate process(es) by PID or executable file path
0x19	Create process with redirected output and upload the output The command uses a format string which hints at execution through cmd.exe • "%param0% /c "%param1% > %tmp_fpath%" 2>&1" If the output is empty, unique string "\x0D\x0A" is uploaded instead
0x1A	Delete specified path. File deletion methods: • delete only • overwrite & move & delete
0x1B	Copy creation/write/access time metadata from source file or directory to destination file or directory. If the source doesn't exist, random time in year 2000-2004 is used for creation & last write time, access time is unchanged.
0x1C	Get info about specified path: • File – attributes, creation/write/access time, type, size • Directory / Drive – total files size, file count, directory count (with optional extension filtering and recursion)
0x1D	Set current configuration blob, save to registry
0x1E	Get current configuration blob
0x1F	Enable/disable drive watchdog (configuration field enable_drive_watchdog)
0x20	Enable/disable session watchdog (configuration field enable_session_watchdog)
0x21	Set configuration value related to delay of backdoor execution (configuration field delay_until_time)
0x23	Store data used by asynchronous command (related to commands 0x12, 0x13)
0x24	Stop executing asynchronous command (related to commands 0x12, 0x13)

ID	Description
0x25	Set configuration value related to delay between failed C&C connection attempts (configuration field wait_minutes)
0x26	If <SYSDIR>\wsdchngr.drx exists • Delete configuration registry value • Delete backdoor file (self delete) • Delete loader file • Read, decrypt, PE-load wsdchngr.drx and call SamIPromote export in a new thread • Exit current thread

Of particular interest are the backdoor's watchdogs, which can be optionally enabled or disabled. There is a drive watchdog used to monitor newly connected and disconnected drives, and a session watchdog monitoring the number of active sessions (i.e. logged-on users). These components can trigger a connection to the C&C server outside the regular, preconfigured three-minute interval, and on new drive and session events.

Configuration

The configuration of the backdoor, which is initially set by the installer, is read from the registry value (shown in Figure 3). When the configuration is modified by a C&C command, the value stored in the registry is updated. An example configuration and its structure are shown in Figure 4.

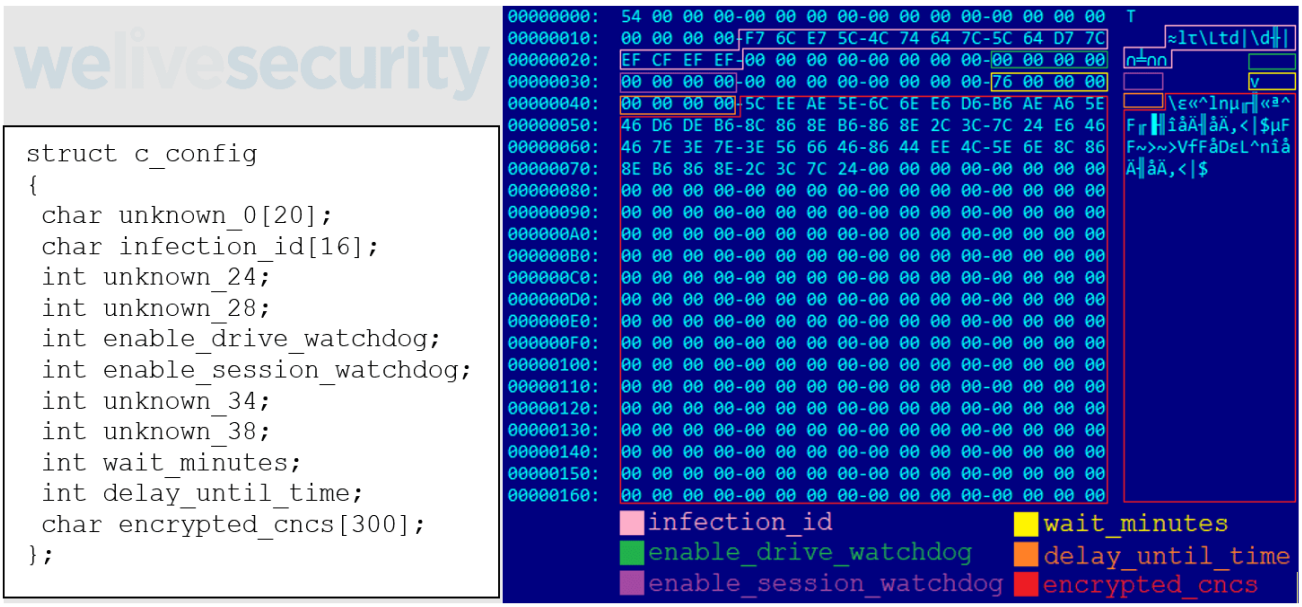

Figure 4. Configuration structure and annotated example

The wait_minutes field specifies the time to wait before next connection to the C&C after a failed connection attempt. If the execution of the backdoor needs to be delayed until a particular time and date, it can be specified in the delay_until_time field. The encrypted_cncs field is an encrypted string, which contains semicolon-separated C&Cs.

Tor library

Vyveva uses the Tor library, which is based on the official Tor source code, to communicate with a C&C server selected at random from the configuration. It contacts the C&C at three-minute intervals, sending information about the victim computer and its drives before receiving commands. The

backdoor's export directory contains the TorSocket.dll with self-explanatory exports close_ch, connect_ch, open_ch, read_ch, write_ch.

Conclusion

Vyveva constitutes yet another addition to Lazarus's extensive malware arsenal. Attacking a company in South Africa also illustrates the broad geographical targeting of this APT group.

For any inquiries, or to make sample submissions related to the subject, contact us at *threatintel@eset.com*.

Indicators of Compromise (IoCs)

Samples

SHA-1	Filename	ESET detection name	Description
DAD50AD3682A3F20B2F35BE2A94B89E2B1A73067	powerctl.exe	Win32/NukeSped.HX	Installer
69529EED679B0C7F1ACC1FD782A4B443CEC0CF83	powerctl.dll	Win32/NukeSped.HX	Loader (x86)
043ADDFB93A10D187DDE4999D78096077F26E9FD	wwanauth.dll	Win64/NukeSped.EQ	Loader (x64)
1E3785FC4FE5AB8DAB31DDDD68257F9A7FC5BF59	wwansec.dll	Win32/NukeSped.HX	Loader (x86)
4D7ADD8145CB096359EBC3E4D44E19C2735E0377	msobjs.drx	-	Backdoor (encrypted)
92F5469DBEFDCEE1343934BE149AFC1241CC8497	msobjs.drx	Win32/NukeSped.HX	Backdoor (decrypted with fixed MZ header)
A5CE1DF767C89BF29D40DC4FA6EAECC9C8979552	JET76C5.tmp	-	Backdoor Tor library (encrypted)
66D17344A7CE55D05A324E1C6BE2ECD817E72680	JET76C5.tmp	Win32/NukeSped.HY	Backdoor Tor library (decrypted with fixed MZ header)

Filenames

%WINDIR%\System32\powerctl.exe
%WINDIR%\SysWOW64\powerctl.exe
%WINDIR%\System32\power.dat
%WINDIR%\SysWOW64\power.dat

%WINDIR%\System32\wwanauth.dll
%WINDIR%\SysWOW64\wwanauth.dll
%WINDIR%\System32\wwansec.dll
%WINDIR%\SysWOW64\wwansec.dll
%WINDIR%\System32\powerctl.dll
%WINDIR%\SysWOW64\powerctl.dll

%WINDIR%\System32\JET76C5.tmp
%WINDIR%\SysWOW64\JET76C5.tmp
%WINDIR%\System32\msobjs.drx
%WINDIR%\SysWOW64\msobjs.drx

MITRE ATT&CK techniques

This table was built using version 8 of the MITRE ATT&CK framework.

Tactic	ID	Name	Description
Execution	T1569.002	System Services: Service Execution	Vyveva loader executes via a service.
	T1106	Native API	Vyveva backdoor uses the CreateProcessA API to execute files.
Persistence	T1543.003	Create or Modify System Process: Windows Service	Vyveva installer creates a new service to establish persistence for its loader.
Defense Evasion	T1140	Deobfuscate/Decode Files or Information	Vyveva decrypts strings and components (backdoor, Tor library).
	T1070.006	Indicator Removal on Host: Timestomp	Vyveva backdoor can timestomp files.
	T1036.004	Masquerading: Masquerade Task or Service	Vyveva installer can create a service with attributes mimicking existing services.
	T1112	Modify Registry	Vyveva stores its configuration in the registry.
	T1027	Obfuscated Files or Information	Vyveva has encrypted strings and components.
Discovery	T1083	File and Directory Discovery	Vyveva backdoor can obtain file and directory listings.
	T1057	Process Discovery	Vyveva backdoor can list running processes.
	T1082	System Information Discovery	Vyveva backdoor can obtain system information, including computer name, ANSI code page, OS version and architecture.
	T1016	System Network Configuration Discovery	Vyveva backdoor can obtain the local IP address of the victim computer.
	T1033	System Owner/User Discovery	Vyveva backdoor can obtain victim's username.

Tactic	ID	Name	Description
	T1124	System Time Discovery	Vyveva backdoor can obtain system time and time zone.
Collection	T1560.002	Archive Collected Data: Archive via Library	Vyveva backdoor can compress files with zlib before sending to C&C.
	T1005	Data from Local System	Vyveva backdoor can collect files from computer.
	T1025	Data from Removable Media	Vyveva backdoor can notify C&C about newly inserted removable media and collect files from them.
Command and Control	T1573.001	Encrypted Channel: Symmetric Cryptography	Vyveva backdoor encrypts C&C traffic using XOR.
	T1573.002	Encrypted Channel: Asymmetric Cryptography	Vyveva backdoor communicates with C&C via Tor.
Exfiltration	T1041	Exfiltration Over C2 Channel	Vyveva exfiltrates data to C&C server.

8 Apr 2021 - 11:30AM

Newsletter

Discussion