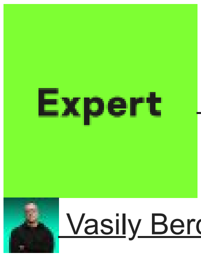
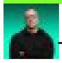
EAGERBEE backdoor

securelist.com/eagerbee-backdoor/115175/

Authors

- 

Saurabh Sharma
-  Vasily Berdnikov

Introduction

In our recent investigation into the EAGERBEE backdoor, we found that it was being deployed at ISPs and governmental entities in the Middle East. Our analysis uncovered new components used in these attacks, including a novel service injector designed to inject the backdoor into a running service. Additionally, we discovered previously undocumented components (plugins) deployed after the backdoor's installation. These enabled a range of malicious activities such as deploying additional payloads, exploring file systems, executing command shells and more. The key plugins can be categorized in terms of their functionality into the following groups: Plugin Orchestrator, File System Manipulation, Remote Access Manager, Process Exploration, Network Connection Listing and Service Management.

In this blog, we provide a detailed analysis of the EAGERBEE backdoor's capabilities, focusing on the service injector, Plugin Orchestrator module and associated plugins. We also explore potential connections of the EAGERBEE backdoor with the CoughingDown threat group.

Initial infection and spread

Unfortunately, the initial access vector used by the attackers remains unclear. However, we observed them executing commands to deploy the backdoor injector named “tsvipsrv.dll” along with the payload file ntusers0.dat, using the SessionEnv service to run the injector, as can be seen below.

```
//change the creation, last access and write time, timestamp of the file to "1/8/2019 9:57"
attrib.exe -s -h -a C:\users\public\ntusers0.dat
powershell.exe -Command "='1/8/2019 9:57'; = 'C:\users\public\ntusers0.dat';(Get-Item
).creationtime = ;(Get-Item ).lastaccesstime = ;(Get-Item ).lastwritetime = "

//set the attributes of the file (EAGERBEE backdoor) to archive (+a), system file (+s) and
//hidden (+h)
attrib.exe +s +h +a C:\users\public\ntusers0.dat

//set the attributes of the file (loader) to archive (+a), system file (+s) and hidden
//(+h)
attrib.exe +s +h +a system32\tsvipsrv.dll

//the malware runs now because of a DLL hijacking vulnerability, as the libraries in the
//system32 directory where the malicious library is located are the first to load
net.exe stop sessionenv
cmd.exe /c "sc config sessionenv Start= auto"
net.exe start sessionenv

attrib.exe -s -h -a C:\users\public\ntusers0.dat
net.exe use \\<<internal ip>>\c$ <password> /user:<username>
attrib.exe +s +h +a C:\users\public\ntusers0.dat
attrib.exe +s +h +a \\172.17.1.127\c$\users\public\ntusers0.dat
attrib.exe -s -h -a system32\tsvipsrv.dll
attrib.exe +s +h +a system32\tsvipsrv.dll
attrib.exe +s +h +a \\172.17.1.127\c$\windows\system32\tsvipsrv.dll
attrib.exe -s -h -a \\172.17.1.127\c$\windows\system32\tsvipsrv.dll
attrib.exe +s +h +a \\172.17.1.127\c$\windows\system32\
```

Malware components

Service injector

The service injector targets the Themes service process. It first locates and opens the process, then allocates memory within it to write EAGERBEE backdoor bytes (stored in C:\users\public\ntusers0.dat) along with stub code bytes. The stub code is responsible for decompressing the backdoor bytes and injecting them into the service process memory.

To execute the stub code, the injector replaces the original service control handler with the address of the stub code in the service process memory. The stub is then triggered by sending a SERVICE_CONTROL_INTERROGATE control code to the service. After the stub completes its execution, the injector cleans up by removing the stub code from the service memory and restoring the original service control handler.

When we found the backdoor in the infected system, it was named dllloader1x64.dll. It can create a mutex with the name mstoolFtip32W if one doesn't exist yet. After that, it starts collecting information from the system: the NetBIOS name of the local computer, OS information (major and minor version numbers, build number, platform identifier, and information about product suites and the latest service pack installed on the system), product type for the operating system on the local computer, processor architecture, and list of IPv4 and IPv6 addresses.

The backdoor has an execution day and time check. It compares the current system day and hour to the hardcoded string 0-6:00:23;6:00:23;, where the numbers mean the following:

- 0: start day of the week;
- 6: end day of the week;
- 00: start hour;
- 23: end hour.

If the execution day and hour do not match, it sleeps for 15 seconds and checks again. In the cases we've seen, the backdoor is configured to run 24/7.

The backdoor configuration is either stored in C:\Users\Public\iconcache.mui or hardcoded within the binary. If stored in the file, the first byte serves as the XOR key to decode the remaining data. When hardcoded, the configuration is decoded using a single-byte XOR key (0x57). This configuration includes the command-and-control (C2) hostname and port.

The backdoor retrieves the proxy host and port information for the current user by reading the registry key Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyServer. If proxy details are available, the backdoor connects through the proxy; otherwise it connects to the C2 server directly.

To establish communication, the backdoor creates a TCP socket capable of operating over both IPv4 and IPv6. If the C2 port has an "s" appended, an SSL session is initiated. Depending on the configuration, it may use the SCHANNEL security package, which supports SSL and TLS encryption on Windows. In this mode, it can validate server credentials (passive mode) or use local client credentials to prepare an outgoing token (active mode).

Once a connection is established, the backdoor transmits the previously collected victim-specific details to the C2 server. The server responds with a string followed by a payload known as the Plugin Orchestrator. If the response string matches a hardcoded value in the backdoor (unique to each sample), the backdoor retrieves the raw address of the first export method in the received payload and invokes it. Notably, at this stage, the payload (Plugin Orchestrator) is not yet mapped into memory.

Plugin Orchestrator

The payload downloaded by the EAGERBEE backdoor is a plugin orchestrator in the form of a DLL file with the internal name “ssss.dll” which exports a single method: “m”. As previously mentioned, EAGERBEE does not map the plugin orchestrator DLL directly into memory. Instead, it retrieves the raw address of the “m” export method and invokes it.

The “m” method of the plugin orchestrator DLL is responsible for injecting the orchestrator into memory and subsequently calling its entry point. In addition to the victim-specific data already collected, the plugin orchestrator gathers and reports to the C2 server the following additional information:

- The NetBIOS name of the domain;
- Current usage of physical and virtual memory;
- System locale and time zone settings;
- Windows character encoding;
- The current process identifier;
- Identifiers for any loaded plugins.

After transmitting this information, the plugin orchestrator also reports whether the current process has elevated privileges. It then collects details about all running processes on the system, including:

- Process identifiers;
- The number of execution threads started by each process;
- The identifier of the parent process;
- The fully qualified path of each process executable.

Once the information is sent, the plugin orchestrator waits for commands to execute. The following commands are supported:

Command	Description
06	This command supports several sub-commands: 2: Receive and inject plugins into memory. There can be multiple plugins loaded one after another. Each plugin has an identifier. 3: Unload a specific plugin from memory, remove the plugin from the list, and free the plugin code bytes. 4: No operation. 5: Remove all plugins from the list and free the plugin code bytes.
07 and 09	Check if the plugin is loaded or not. If the plugin is loaded, then call the specified export method of the plugin. If the plugin is not loaded, then check if the plugin has been received, then load it, and call the specified export method. Otherwise, just make a plugin entry.

Plugins

The plugins are DLL files and export three methods using ordinals. The plugin orchestrator first calls the exported method of the plugin with the ordinal number 3. This method is responsible for injecting the plugin DLL into memory. After that, the orchestrator calls the exported method of the plugin with the ordinal number 1, which is the DllMain method. This method initializes the plugin with the required data structures. Finally, it calls the exported method of the plugin with the ordinal number 2. This method implements the functionality of the plugin.

All the plugins are responsible for receiving and executing commands from the orchestrator. Below, we provide brief descriptions of the analyzed plugins and the commands supported by each of them.

File Manager Plugin

This plugin performs a wide range of file system functions, including:

- Listing drives, files and folders in the system;
- Renaming, moving, copying and deleting files;
- Setting ACLs to manage file and folder permissions;
- Reading and writing files to and from the system;
- Injecting additional payloads into memory.

The table below contains commands it accepts.

Command	Description
0x02	Check and enable SeDebugPrivilege, SeBackupPrivilege, SeRestorePrivilege and SeTakeOwnershipPrivilege for the current process.
0x06	• List files and folders at the specified path or at some of the following locations: DESKTOP, MYDOCUMENTS, RECYCLE.BIN, FAVORITES, STARTUP, RECENT, “C:\Windows\Prefetch” and the window credential manager storage folder. • Get information about USB storage devices that have been connected to the computer by querying the registry key HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR.
0x07	Get information about drives.
0x08	Delete multiple directories or files.
0x09	Create a directory at the specified location.
0x0A (10)	Rename an existing directory/file to a new directory/file.
0x0B (11)	Move or copy an existing directory/file to a new directory/file.

0x0C (12)	Move or copy multiple existing directories/files to new directories/files.
0xD (13)	Reflectively inject the received executable and DLL into memory.
0x0F (15)	• Get a list of files and folders at a specified location recursively. • Read a file by dumping file sectors of the specified file directly from disk. • Write a file.
0x14 (20)	Launch the passed command line via the CreateProcessW API. The module can also launch the passed command line via CreateProcessAsUserW to run in the security context of the user represented by the token of specified process ID.
0x22 (34)	Adjust the security (DACL) for the user groups LOCAL SYSTEM, AUTHENTICATED USERS, DOMAIN ADMINISTRATOR and DOMAIN USER to grant access to specified file or directory.
0x23 (35)	Load a DLL at the specified path via LoadLibraryW.
0x24 (36)	Set the label of a file system volume.
0x26 (38)	• Copy an existing file to a new file. • Change the existing and new file time parameters (last write time, last access time and creation time) to those of user32.dll.

Process Manager

This plugin manages process-related activities such as:

- Listing running processes in the system;
- Launching new modules and executing command lines;
- Terminating existing processes.

It accepts four commands.

Command	Description
0x10 (16)	Terminate the process with the specified process ID.
0x11 (17)	Run the passed command line via the CreateProcessW API. Process Manager can also launch the specified module via CreateProcessAsUserW to run in the security context of the user represented by the token of specified process ID.
0x1E (30)	Get information about the list of running processes in the system. The module also collects user accounts associated with the processes.

0x26 (38)	Set file attribute.

Remote Access Manager

This plugin facilitates and maintains remote connections, while also providing command shell access.

Command	Description
0x0B (11)	Perform the operations below to enable and persist an RDP session: ○ Set remote desktop services to autostart. ○ Keep the Windows remote access service (RAS) session opened after logging off. ○ Enable remote desktop connections. ○ Enable concurrent (multiple) RDP sessions. ○ After performing the above settings, start the remote desktop service (TermService).
0x0D (13)	Download a file from the specified URL and write to the specified file path. Then start the remote desktop service (TermService).
0x1D (29)	• Start the command shell (cmd.exe). The module can also run cmd.exe by injecting its code into the process C:\Windows\System32\dllhost.exe. • Read data from the command shell and send it to the C2 server.
0x1E (30)	If the command shell process is not running, then start the command shell (cmd.exe) and write the received command data from C2 to the command shell.
0x21 (33)	Terminate the thread to read the command output from the command shell console. Then terminate the command shell process.

The attackers launch the command shell by injecting cmd.exe into the DllHost.exe process. The commands below were seen executed by the threat actor:

```
//list all users and users in the local admin group
net user
net localgroup administrators

//obtain system- and account-related information; the "dsquery" command implies that the
//attacker got hold of a Windows server machine with the Active Directory Domain Services
//(AD DS) server role installed.
dsquery computer
dsquery server
dsquery users
dsquery user
systeminfo
ping  -n 1 <<computer name>>

//establish a connection to a shared resource using stolen credentials
net use \\<<ip in the network>>\admin$ <password> /user:<username>

//archive the information from the shared resource
rar.exe  a -v100M idata001.rar -ta"20240101000000" -r -x"*.mp3" -x"*.dll" -x"*.exe" -
x"*.zip" -x"*.mxf" -x"*.rar" "\\<<ip in the network>>\c$\Users\<<user name>>\Documents"
"\\<<ip in the network>>\c$\Users\<<user name>>\Desktop"
rar.exe  a -v100M idata001.rar -ta"20240101000000" -r -x"*.mp3" -x"*.dll" -x"*.exe" -
x"*.zip" -x"*.mp4" -x"*.rar" "\\<<ip in the network>>\c$\Users\<<user name>>\Documents"
"<<ip in the network>>\c$\Users\<<user name>>\Desktop"
```

Service Manager

This plugin manages system services, including installing, starting, stopping, deleting and listing them.

Command	Description

0x11 (17)	Create Service entries. The module can create the following types of services: ○ SERVICE_WIN32_SHARE_PROCESS: shares a process with other services. ○ SERVICE_WIN32_OWN_PROCESS: runs inside its own process.
0x12 (18)	Stop and delete the service.
0x13 (19)	Start a service.
0x14 (20)	Stop a service.
0x1E (30)	Enumerate all services (active and inactive) to collect the following information about services: the service name, display name and service status information.

Network Manager

This plugin lists the network connections in the system.

Command	Description
0x1E (30)	Get information about the list of IPv4 and IPv6 TCP and UDP connections: • State • Local address • Local port • Remote address • Remote port • Owning PID

Attribution

EAGERBEE was deployed in several organizations in East Asia. Two of these organizations were breached via the infamous ProxyLogon vulnerability (CVE-2021-26855) in Exchange servers, after which malicious webshells were uploaded and utilized to execute commands on the breached servers.

In May 2023, our telemetry indicated the execution of multiple commands to start and stop system services at one of the affected organizations in East Asia. The attackers abused the legitimate Windows services MSDTC, IKEEXT and SessionEnv to execute malicious DLLs: oci.dll, wlbsctrl.dll and TSVIPSrv.dll, respectively.

```
tasklist.exe
net stop IKEEXT
net start IKEEXT

net start msdtc
net stop msdtc
net start msdtc

NETSTAT.EXE -ano
tasklist.exe
ARP.EXE -a

net.exe use \\[[IP REDACTED]]\admin$
ipconfig.exe /all

net.exe stop IKEEXT

//all privileges are assigned to the service IKEEXT, which loads the malicious DLL
reg.exe add hklm\SYSTEM\CurrentControlSet\Services\IKEEXT /v RequiredPrivileges /t
REG_MULTI_SZ /d
SeAuditPrivilege\0SeBackupPrivilege\0SeRestorePrivilege\0SeTakeOwnershipPrivilege\0SeImper
sonatePrivilege\0SeTcbPrivilege\0SeAssignPrimaryTokenPrivilege\0SeManageVolumePrivilege\0S
eCreateSymbolicLinkPrivilege\0SeShutdownPrivilege /f

net.exe start IKEEXT
net.exe start IKEEXT

NETSTAT.EXE -ano
net.exe view

net.exe stop IKEEXT
net.exe start IKEEXT
net.exe start IKEEXT

net.exe start sessionenv
net.exe stop sessionenv

net.exe stop SessionEnv
net.exe start SessionEnv
net.exe start SessionEnv
net.exe start SessionEnv
net.exe start SessionEnv
net.exe start SessionEnv
net.exe stop SessionEnv
net.exe stop SessionEnv
```

According to our telemetry, the DLLs loaded and executed by the services IKEEXT and SessionEnv are loaders in nature, loading the EAGERBEE backdoor into memory. Similar EAGERBEE loaders targeting Japanese organizations have been described by another security vendor. Examples of files loaded by these services are provided below.

IKEEXT

MD5	File Name	Compilation Time	EAGERBEE Payload File
5633cf714bfa4559c97fc313650f86dd	wlbsctrl.dll	Monday, 23.05.2022 14:02:39 UTC)	C:\Users\Public\Videos\<*name*>.mui
3ccd5827b59ecd0043c112c3aafb7b4b	wlbsctrl.dll	Sunday, 01.01.2023 20:58:58 UTC	%tmp%*g.logs

SessionEnv

MD5	File Name	Compilation Time	EAGERBEE Payload File
67565f5a0ee1deffe0f368860be78e1e	TSVIPSrv.dll	Wednesday, 25.05.2022 15:38:06 UTC	C:\Users\Public\Videos\<*name*>.mui
00d19ab7eed9a0ebcaab2c4669bd34c2	TSVIPSrv.dll	Sunday, 01.01.2023 20:50:01	%tmp%*g.logs

The service MSDTC loaded and executed a DLL file named “ oci.dll”. By analyzing this file, we established that it was the CoughingDown Core Module.

MD5	File Name	Compilation Time	Description
f96a47747205bf25511ad96c382b09e8	oci.dll	Thursday, 24.02.2022 05:18:05 UTC	CoughingDown Core Module

We found several clues linking the EAGERBEE backdoor to the CoughingDown group:

1. One of the aforementioned DLLs, oci.dll (MD5 f96a47747205bf25511ad96c382b09e8), which is executed by abusing the legitimate MSDTC service, has a 25% match with CoughingDown samples according to the Kaspersky Threat Attribution Engine (KTAE). Analysis of the DLL reveals that it is a Core Module of multi-plugin malware developed by CoughingDown in late September 2020 and that there is indeed a significant code overlap (same RC4 key, same command numbers).
2. This Core Module was configured to use the IP addresses 45.90.58[.]103 and 185.82.217[.]164 as its C2. The IP address 185.82.217[.]164 is known to be used as an EAGERBEE C2 as reported by other security vendors.

Conclusions

Malware frameworks continue to advance as threat actors develop increasingly sophisticated tools for malicious activities. Among these is EAGERBEE, a malware framework primarily designed to operate in memory. This memory-resident architecture enhances its stealth capabilities, helping it evade detection by traditional endpoint security solutions. EAGERBEE also obscures its command shell activities by injecting malicious code into legitimate processes, such as dllhost.exe, and executing it within the context of explorer.exe or the targeted user’s session. These tactics allow the malware to seamlessly integrate with normal system operations, making it significantly more challenging to identify and analyze.

In the East Asian EAGERBEE attacks, the organizations were penetrated via the ProxyLogon vulnerability. ProxyLogon remains a popular exploit method among attackers to gain unauthorized access to Exchange servers. Promptly patching this vulnerability is crucial to securing your network perimeter.

Because of the consistent creation of services on the same day via the same webshell to execute the EAGERBEE backdoor and the CoughingDown Core Module, and the C2 domain overlap between the EAGERBEE backdoor and the CoughingDown Core Module, we assess with medium confidence that the EAGERBEE backdoor is related to the CoughingDown threat group.

However, we have been unable to determine the initial infection vector or identify the group responsible for deploying the EAGERBEE backdoor in the Middle East.

IOC

Service Injector
183f73306c2d1c7266a06247cedd3ee2

EAGERBEE backdoor compressed file
9d93528e05762875cf2d160f15554f44

EAGERBEE backdoor decompress
c651412abdc9cf3105dfbafe54766c44

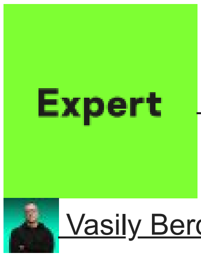
EAGERBEE backdoor decompress and fix
26d1adb6d0bcc65e758edaf71a8f665d

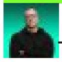
Plugin Orchestrator
cbe0cca151a6ecea47cfaa25c3b1c8a8
35ece05b5500a8fc422cec87595140a7

Domains and IPs
62.233.57[.]94
82.118.21[.]230
194.71.107[.]215
151.236.16[.]167
www.socialentertainments[.]store
www.rambiler[.]com
5.34.176[.]46
195.123.242[.]120
195.123.217[.]139

- APT
- Backdoor
- CoughingDown
- Cyber espionage
- EagerBee
- Malware
- Malware Descriptions
- Malware Technologies
- Targeted attacks

Authors

- 

 Saurabh Sharma
- 
 Vasily Berdnikov

EAGERBEE, with updated and novel components, targets the Middle East

Your email address will not be published. Required fields are marked *