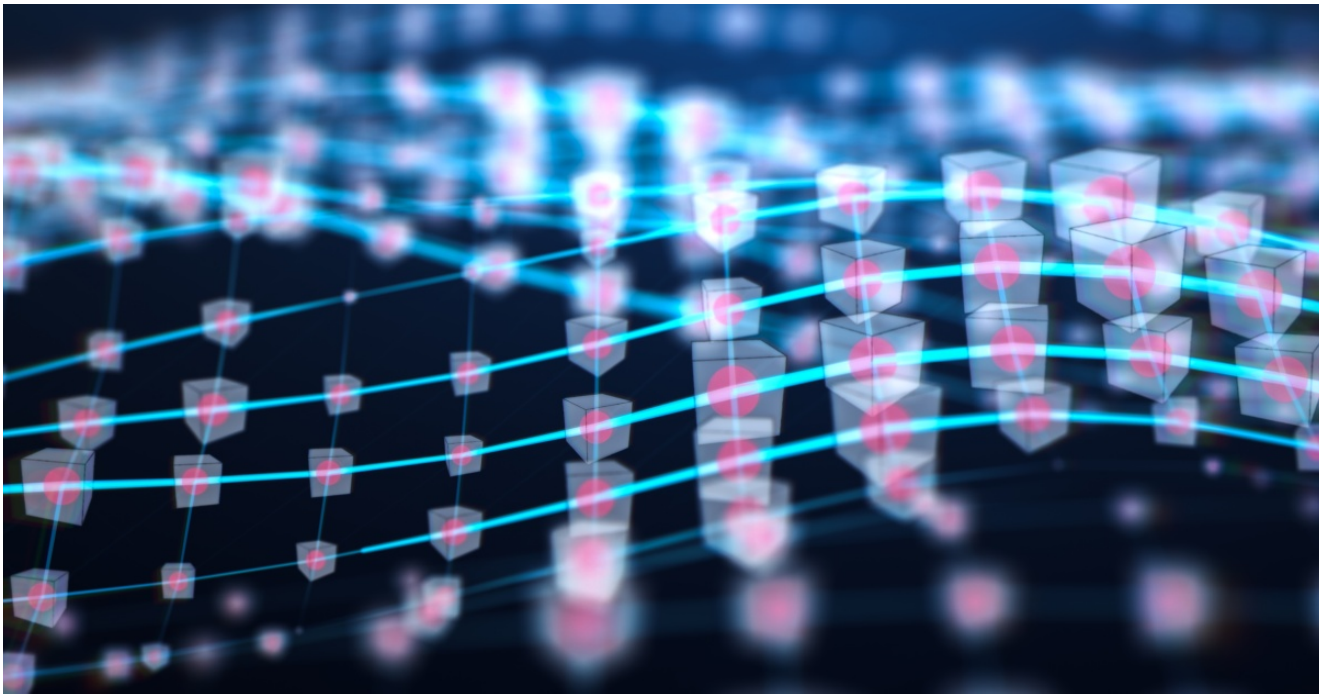
Verblecon: Sophisticated New Loader Used in Low-level Attacks

symantec-enterprise-blogs.security.com/blogs/threat-intelligence/verblecon-sophisticated-malware-cryptocurrency-mining-discord



An unknown attacker is using a complex and powerful new malware loader in relatively unsophisticated and low-reward attacks, indicating they may not realize the potential capabilities of the malware they are deploying.

The malware, Trojan.Verblecon, is being used in attacks that appear to have installing cryptocurrency miners on infected machines as their end goal. There are some indications the attacker may also be interested in stealing access tokens for chat app Discord. However, the capabilities of this malware indicate that it could be highly dangerous if leveraged in ransomware or espionage campaigns.

Verblecon was first spotted by analysts from Symantec, a division of Broadcom Software, in January 2022. This blog will detail the capabilities of the malware.

Technical breakdown

The malware is loaded as a server-side polymorphic JAR file. The fact that the file is polymorphic means that, due to encryption and obfuscation, the code of the malware payload looks different each time it is downloaded. Attackers generally pack malware in this way in an effort to evade detection by security software.

The malware samples analyzed by Symantec were fully obfuscated, in the code flow, strings, and symbols. The samples themselves may be based on publicly available code.

Once started, the malware checks its command-line arguments. It requires at least one command-line argument to execute, which could be the infection or campaign ID initially e.g.

"CSIDL_SYSTEM_DRIVE\program files\java\jre1.8.0_301\bin\javaw.exe" -jar "CSIDL_PROFILE\appdata\local\temp\rpvbh.jar" masonkhonsari

and

"CSIDL_SYSTEM_DRIVE\program files\java\jre1.8.0_301\bin\javaw.exe" -jar "CSIDL_PROFILE\appdata\local\temp\rpvbh.jar" 923ec15ffa4474ca7bf200bfb90e782d

Additionally, it also attempts to determine if its own process is being debugged by checking for the following Java command-line arguments:

- "-xbootclasspath"
- "-xdebug"
- "-agentlib"
- "-javaagent:"
- "-xrun:"
- "-verbose"
- "-agentpath:"

Next, it attempts to detect if it is being opened in a virtual or sandbox environment, which would indicate it is likely being opened on a security researcher’s machine.

First, it checks for the following directories:

- "%ProgramFiles(X86)%\VMware\VMware Tools"
- "%ProgramFiles(X86)%\Oracle\VirtualBox Guest Additions"

It also obtains the machine MAC address and attempts to check for the following prefixes, which may indicate the file is being opened on a virtual machine:

- "00:05:69"
- "00:0C:29"
- "00:1C:14"
- "00:50:56"
- "08:00:27"
- "00:16:3E"
- "00:1C:42"
- "0A:00:27"

Following those checks, it executes the following command to obtain a list of running processes:

```
tasklist.exe /fo csv /nh
```

It then appears to check these processes against a set list:

- "vboxservice.exe"
- "vboxtray.exe"
- "xenservice.exe"
- "vmtoolsd.exe"
- "vmwaretray.exe"
- "vmwareuser.exe"
- "vgauthservice.exe"
- "vmacthlp.exe"
- "vmsrvc.exe"
- "vmusrvc.exe"
- "prl_cc.exe"
- "prl_tools.exe"
- "qemu-ga.exe"
- "vmcomputeagent.exe"
- "sandboxie"
- "vdagent"
- "vdservice"
- "fiddler"
- "joeboxserver.exe"
- "joeboxcontrol.exe"
- "blnsvr.exe"

It then also checks for the following files:

- "%Windows%\system32\windanr.exe"
- "%Windows%\system32\drivers\VBoxMouse.sys"
- "%Windows%\system32\drivers\VBoxGuest.sys"
- "%Windows%\system32\drivers\VBoxSF.sys"
- "%Windows%\system32\drivers\VBoxVideo.sys"
- "%Windows%\system32\vboxdisp.dll"
- "%Windows%\system32\vboxhook.dll"
- "%Windows%\system32\vboxmrxnp.dll"
- "%Windows%\system32\vboxogl.dll"
- "%Windows%\system32\vboxoglarrayspu.dll"
- "%Windows%\system32\vboxoglcrutil.dll"
- "%Windows%\system32\vboxoglerrorspu.dll"
- "%Windows%\system32\vboxoglfeedbackspu.dll"

- "%Windows%\system32\vboxoglpackspu.dll"
- "%Windows%\system32\vboxoglpassthroughspu.dll"
- "%Windows%\system32\vboxservice.exe"
- "%Windows%\system32\vboxtray.exe"
- "%Windows%\system32\VBoxControl.exe"
- "%Windows%\system32\Drivers\Vmmouse.sys"
- "%Windows%\system32\Drivers\vm3dgl.dll"
- "%Windows%\system32\Drivers\vmdum.dll"
- "%Windows%\system32\Drivers\vm3dver.dll"
- "%Windows%\system32\Drivers\vmtray.dll"
- "%Windows%\system32\Drivers\VMToolsHook.dll"
- "%Windows%\system32\Drivers\vmmousever.dll"
- "%Windows%\system32\Drivers\vmhgfs.dll"
- "%Windows%\system32\Drivers\vmGuestLib.dll"
- "%Windows%\system32\Drivers\VmGuestLibJava.dll"
- "%Windows%\system32\Driversvmhgfs.dll"
- "[java.lang.System.getProperty("user.home")]\Desktop\moutonheart.wav"

Next, it appears to check the user name against the following:

- java.lang.System.getProperty("user.name") == "WDAGUtilityAccount"
- java.lang.System.getProperty("user.name").startsWith("hal-")

Then it executes the following command:

```
reg query "HKU\S-1-5-19"
```

It is unclear how the output is processed, however, there are some strings that could be related to this or other registry checks:

- "HARDWARE\ACPI\DSDT\"
- "HARDWARE\ACPI\FADT\"
- "HARDWARE\ACPI\RSDT\"
- "SOFTWARE\Oracle\"
- "SYSTEM\ControlSet001\Services\"
- "SYSTEM\ControlSet001\Services\"
- "SOFTWARE\Microsoft\Virtual Machine\Guest\"
- "SOFTWARE\VMware, Inc.\"
- "SOFTWARE\"
- "VBOX__"
- "VBOX__"
- "VirtualBox Guest Additions"
- "VBoxGuest"
- "VBoxMouse"

- "VBoxService"
- "VBoxSF"
- "VBoxVideo"
- "Parameters"
- "VMware Tools"
- "Wine"

If satisfied with these checks, it may copy itself as one of the following files:

- "%ProgramData%[INFECTION_ID][INFECTION_ID].jar"
- "%ALL_USERS_HOME%[INFECTION_ID][INFECTION_ID].jar"
- "%LOCALAPPDATA%[INFECTION_ID][INFECTION_ID].jar"

And then create one of the following files to use as a loadpoint:

- "%HOMEPATH%\Library\LaunchAgents[INFECTION_ID].plist"
- "%Windows%\System32\Tasks[INFECTION_ID]"

[INFECTION_ID] is computed as follows:

```
hashlib.md5(b"%PROCESSOR_IDENTIFIER%%COMPUTERNAME%[USER_NAME]").hexdigest()
```

Then it periodically attempts to connect to the following URLs:

- "hxxps://gaymers[.]ax/"
- "hxxp://[DGA_NAME][.]tk/"

[DGA_NAME] is apparently generated using the following method: