# Glupteba Campaign Hits Network Routers and Updates C&C Servers with Data from Bitcoin Transactions


- Posted on:September 4, 2019 at 4:57 am
- Posted in:Exploits, Malware, Vulnerabilities
- Author:
  Trend Micro


0

***by Jaromir Horejsi and Joseph C. Chen***

We recently caught a malvertising attack distributing the malware Glupteba. This is an older malware that was previously connected to a campaign named Operation Windigo and distributed through exploit kits to Windows users. In 2018, a security company reported that the Glupteba botnet may have been independent from Operation Windigo and had moved to a pay-per-install adware service to distribute it in the wild. The activities of the actors behind Glupteba have been varied: they were suspected of providing proxy services in the underground, and were identified as using the EternalBlue exploit to move into local networks and run Monero (XMR) cryptocurrency miners.

After looking into the recent variant of the Glupteba dropper delivered from the malvertising attack, we found that the dropper downloaded two undocumented components aside from the Glupteba malware:

- A browser stealer that can steal sensitive data, for example, browsing history, website cookies, and account names and passwords from browsers and send the information to a remote server.
- A router exploiter that attacks MikroTik routers in local network with the CVE-2018-14847 vulnerability. It will schedule a task on the router for command and control (C&C) and upload the stolen administrator credentials to a remote server. A compromised router will be configured as a SOCKS proxy to relay malicious traffic, matching the original purpose of the Glupteba botnet on Windows.

In addition, an interesting feature we found inside the Glupteba dropper can retrieve the latest C&C domain from Bitcoin transactions. We explain this feature further in the next sections. It seems the operators are still improving their malware and may be trying to extend their proxy network to internet of things (IoT) devices.

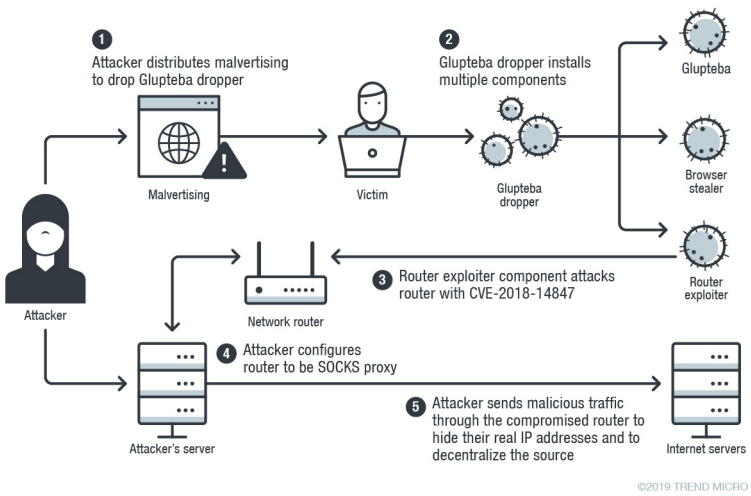



*Figure 1. Glupteba campaign attack flow*

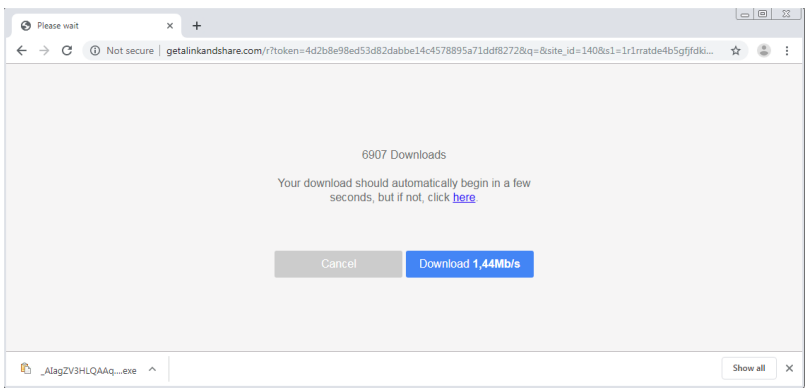
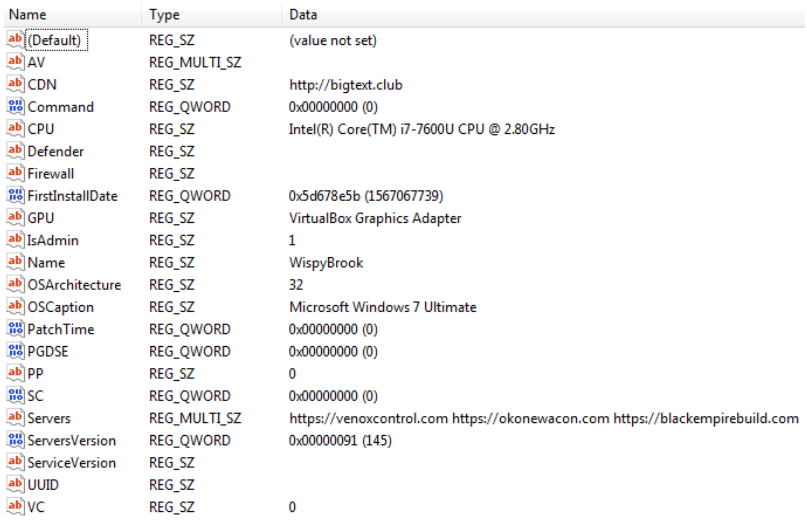



*Figure 2. Pop-up malvertising on file-sharing websites downloaded Glupteba dropper*

***Analysis of the Glupteba dropper***

The downloaded dropper binary is packed with a custom packer, written in Go programming language, and compiled to executable. The dropper first initializes 'config information' by acquiring current application information, operating information, hardware information, as well as some information hardcoded in binary. It

creates a registry key *HKEY_USERS\<sid>\Software\Microsoft\TestApp* to store all the acquired information. The result of running the config initialization function is shown in the figure below.



| Name | Type | Data |
|---|---|---|
| (Default) | REG_SZ | (value not set) |
| AV | REG_MULTI_SZ | |
| CDN | REG_SZ | http://bigtext.club |
| Command | REG_QWORD | 0x00000000 (0) |
| CPU | REG_SZ | Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz |
| Defender | REG_SZ | |
| Firewall | REG_SZ | |
| FirstInstallDate | REG_QWORD | 0x5d678e5b (1567067739) |
| GPU | REG_SZ | VirtualBox Graphics Adapter |
| IsAdmin | REG_SZ | 1 |
| Name | REG_SZ | WispyBrook |
| OSArchitecture | REG_SZ | 32 |
| OSCaption | REG_SZ | Microsoft Windows 7 Ultimate |
| PatchTime | REG_QWORD | 0x00000000 (0) |
| PGDSE | REG_QWORD | 0x00000000 (0) |
| PP | REG_SZ | 0 |
| SC | REG_QWORD | 0x00000000 (0) |
| Servers | REG_MULTI_SZ | https://venoxcontrol.com https://okonewacon.com https://blackempirebuild.com |
| ServersVersion | REG_QWORD | 0x00000091 (145) |
| ServiceVersion | REG_SZ | |
| UUID | REG_SZ | |
| VC | REG_SZ | 0 |

*Figure 3. Registries created by the Glupteba dropper*

Then, the function *sendParentProcesses* acquires machine_guid from the registry, as well as distributor id and campaign id from the file name, product identification (PID), and names of parent processes. It then embeds this information in a POST request, encrypts it with an AES cipher, and uploads to the C&C server: *hxxps://<server>/api/parent-processes*.

After that, the dropper checks if process is elevated and running as a SYSTEM user. If process is not elevated, it tries to exploit the fodhelper method to get it elevated. If it is elevated but not running as a SYSTEM user, it uses the "Run as Trusted Installer" method, likely inspired by this code, which uses a stolen winlogon process token to run process as SYSTEM.

The main dropper binary has embedded a few rootkit drivers used for hiding files and processes (WinMon32.sys, WinMon64.sys, WinMonFs32.sys, WinMonFs64.sys, WinMonprocessmonitor32.sys, WinMonProcessMonitor64.sys, WinmonSystemMonitor-10-64.sys, WinmonSystemMonitor-7-10-32.sys, WinmonSystemMonitor-7-64.sys) and a few other tools taken from GitHub used to help installing the necessary drivers (dsefix.exe and patch.exe.)

Function *executeTask* processes these main commands:

| | |
|---|---|
| hide | Hide task PID using embedded WinMon |
| update | Terminate and remove current version, replace with new version |
| cleanup | Uninstall |

Function *mainInstall* checks for installed antivirus (AV) programs, adds firewall rules, and adds defender exclusions.

Function *mainPoll* regularly polls the C&C server for new commands. It sends a POST request to hxxps://<server>/api/poll. POST parameters look like the following (before encryption):

challenge=e94e354daf5f48ca&cloudnet_file=1&cloudnet_process=1&lcommand=0&mrt=1&pgdse=0&sb=1&sc=0&uuid=&version=145&wup_process=1&wu

The query is AES 256-encrypted.

Finally, function *handleCommand* implements backdoor functions.

| **Function** | **Task** |
|---|---|
| update | |
| get_app_name | |
| is_admin | |
| process_is_running | Queries "SELECT Name FROM Win32_Process WHERE Name =" |
| exec | |
| download | |
| run | |
| run-v2 | |
| exit | |
| update | Download and execute file |
| update-data | POST internal config to /bots/update-data |
| update-cloudnet | Download file from hxxp://nxtfdata[.]xyz/cl.exe, replaces *cloudnet.exe* file, which is Glupteba |
| stop-wup | Stop XMR mining |
| stop-wupv | Stop XMR mining |
| stop-mrt | |
| notify | Establish heartbeat, notification to URL with a given time interval, notifyHTTP, notifyH, notifyG, notifyS, notifyTCP, notifyTLS, notifyUDP |
| notify-host | Host for notification |
| event-exists | if Global\\<event name> exists |
| mutex-exists | if Global\\<mutex name> exists |
| registry-get-startup | HKEY_USERS\%s\Software\Microsoft\Windows\CurrentVersion\Ru |

| | |
|---|---|
| | n |
| verify-signature | Verify signature of PE file |
| registry-get-startup-signatures | Verify signatures of PE files from startup |
| verify-processes-signatures | Enumerate processes, verify signatures |
| get-unverified-files | Calls VerifyProcessesSignatures and RegistryGetStartupSignatures, reports unverified files |
| get-stats-wup | Query hxxp://localhost:3433/, GET cryptominer stats, *wup.exe* is the open-source miner for XMR |
| upload-file | File upload |
| update-service | Download and run service |
| get-logfile-proxy | Read file \\proxy\\t |
| install | Download and run file, sendInstallReport |
| get-logfile-i2pd | Read file \\i2pd\\i2pd.log |
| sc | Take screenshot |
| update-cdn | Update C2 |
| discover-electrum | Use hardcoded Electrum wallet; read blockchain transaction data |
| discover-blockchaincome | Use hardcoded Bitcoin address, discover new C2 domain encrypted in bitcoin transaction data |

***Notable C&C update capability***

The backdoor mostly has standard capabilities, but one interesting feature stands out: This malware can update its C&C server address through the blockchain via the function *discoverDomain*.

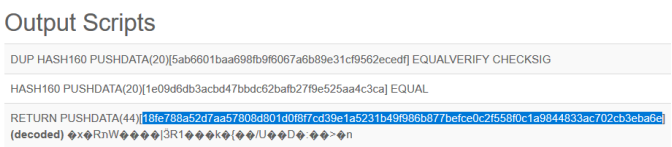
The *discoverDomain* function can be run either by sending a backdoor command, or automatically by the dropper. *DiscoverDomain* first enumerates Electrum Bitcoin wallet servers using a publicly available list, then tries to query the blockchain script hash history of the script with a hardcoded hash.

```
{
    "jsonrpc": "2.0",
    "id": 1,
    "method": "blockchain.scripthash.get_history",
    "params": [
        "f3ebe86400fc08e24f3db53f43dd82a8fd7152cc7a
    ]
}
```

This command then reveals all the related transactions.

```
[{
    "tx_hash": "8a7c43d0bbf01cdf3bb28de48e76d2e1d6e339a063251fce30cb83ae50c2096a",
    "height": 581123
}, {
    "tx_hash": "55e8fe62bcc41ec465c3f1f2804a7457cf5d82443a15a30d88fefc3f55ad2f29",
    "height": 581372
}]
```

Then each transaction is parsed, searching for the OP_RETURN instruction.




The pieces of data followed by OP_RETURN instruction are then used as parameters for AES decryption routine — the first 12 bytes are used as the AES GCM tag, and the following 32 bytes are the encrypted data. The 32-byte long AES key is hardcoded in binary file.

```
003C7EC0: D8 72 7A 0E 9D A3 E9 8B|2E 4E 14 CE 5A 6C F3 3E  | Ørz..éé■.N.ÎZló>
003C7ED0: F2 6C 62 31 56 2A 33 93|CA 46 56 29 D6 65 03 CF  | òlb1V*3■ÊFV)Öe.Ï
```

Therefore, 0f8f7cd39e1a5231b49f986b877befce0c2f558f0c1a9844833ac702cb3eba6e gets decoded to venoxcontrol[.]com, which is the current C&C server at the time of writing this publication.

This technique makes it more convenient for the threat actor to replace C&C servers. If they lose control of a C&C server for any reason, they simply need to add a new Bitcoin script and the infected machines obtain a new C&C server by decrypting the script data and reconnecting.

***Browser stealer component***

One observed component from the recent Glupteba variant is called "updateprofile", which is a browser profile, cookies, and password extractor. The Chrome, Opera, and Yandex browsers are targeted — cookies, history, and other profile files are zipped and uploaded to the information collection server to path */api/log*. Similar to the main dropper, this component is also written in Go, compiled to be executable, and packed with a UPX packer.

Another version of the browser stealer is called "vc.exe". Its goal is to extract browser passwords and cookies and post the extracted data to the information collection server to path */bots/post-en-data?uuid=*.

***Router exploiter component***

Another component we found downloaded by the Glupteba dropper is a router exploiter, which is also developed with Go language. It looks into the default gateway of the victim's network. The list of default IP gateways is obtained by calling WMI command "SELECT DefaultIPGateway FROM Win32_NetworkAdapterConfiguration WHERE IPEnabled = true".

In addition to these addresses, the following three default addresses are added: 192.168.88.11, 192.168.0.1, 192.168.1.1.

Once the component successfully connects to the device listening on port 8291, it then attempts to exploit the device with the CVE-2018-14847 vulnerability, which affects the RouterOS system used on MikroTik routers. The exploit code was likely inspired by this code on exploit-db. It allows the attackers to grab the administrator's credentials from unpatched routers. The grabbed account names and passwords are stored in a JSON object, encrypted, and POSTed to /api/router path of the C&C server.

Once credentials are successfully obtained, a task is added to the scheduler of the router. There are three methods implemented to add a scheduler task: using WinBox protocol, using SSH, or using API.

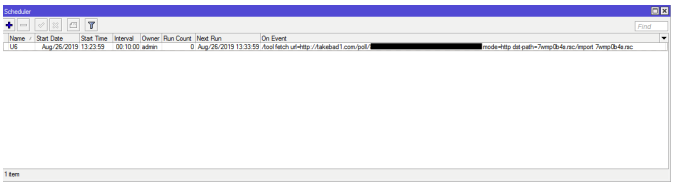


*Figure 4. Example of scheduled task added by Glupteba campaign to compromised MikroTik routers*

The router exploiter component scheduled a task named "U6" on compromised routers for command and control. The task will regularly check a C&C URL every 10 minutes and execute the content downloaded from it. The C&C URL is appended with a unique UUID, which is the same as the bot ID of Glupteba on the victim's machine. The URL usually returns an HTTP 404 error; but when the bot master sends a command, it returns an RSC file, which is the format of the RouterOS configuration.

The file is the command from the bot master to control the router. In the attack we analyzed, we saw the C&C server send multiple RSC files step by step to configure the compromised router to be a SOCKS proxy. Here are the steps:

- The first configuration changed the period of "U6" task schedule from 10 minutes to every 15 seconds.

```
:do { /system scheduler set U6 interval=00:00:15 } on-error={ :put "U6 not found"}
```

- The second configuration disabled services, including winbox, telnet, api, and api-ssl. This is most likely to prevent the router from being compromised by other attackers through the same vulnerability. Then it opened the SSH and SOCKS services, which are listening on randomly assigned ports, and created a firewall rule to accept external connection to the SOCKS port.

```
/ip service disable winbox
/ip service disable telnet
/ip service disable api
/ip service disable api-ssl
/ip service set ssh port=21781
/ip socks set port=2079
/ip firewall filter add action=accept chain=input disabled=no dst-port=2079 protocol=tcp place-before=1
```

- The third configuration removed the existing SOCKS access list on the compromised router.

```
{ /ip socks access print; :local countRules [/ip socks access print count-only]; :for i from 0 to $countRules step=1 do= {/
ip socks access remove $i} }
```

- The fourth configuration added a new SOCKS access list to limit the service so that it only accepts connections from specified IP ranges. These ranges are probably where the attackers' servers are.

```
/ip socks access add src-address=5.188.42.0/24 action=allow
/ip socks access add src-address=85.119.151.0/24 action=allow
/ip socks access add src-address=77.238.240.0/24 action=allow
/ip socks access add src-address=192.243.53.0/24 action=allow
/ip socks access add src-address=95.213.221.0/24 action=allow
/ip socks access add src-address=159.255.24.0/24 action=allow
/ip socks access add src-address=31.184.210.0/24 action=allow
/ip socks access add src-address=178.239.168.0/24 action=allow
/ip socks access add src-address=193.188.22.205/32 action=allow
/ip socks access add src-address=146.0.78.6/32 action=allow
/ip socks access add src-address=31.172.128.25/32 action=allow
/ip socks access add src-address=10.0.0.0/8 action=allow
/ip socks access add src-address=0.0.0.0/0 action=deny
```

***Relayed traffic on compromised routers***

After the above-mentioned setups, the compromised router became a SOCKS proxy for the attackers to relay traffic. We monitored a compromised router to see what kind of traffic is transferred. The first remote connection routed through the SOCKS proxy is from a server, which likely belongs to the attackers. This server queries "http://ip-api[.]com/json", which returns the IP address of the current SOCKS proxy server. This query is sent repeatedly, probably to monitor the SOCKS proxy service.

After the first check on the router status, we started seeing different servers with two types of traffic connected to the proxy. The first one is spam traffic. We saw a remote server establish SMTP connections to different mail servers through the SOCKS proxy of compromised routers. If a mail server accepted the connection, that remote server started to send spam mail. The spam mail delivered seems to be related to the notorious "Canadian Pharmacy" spam.

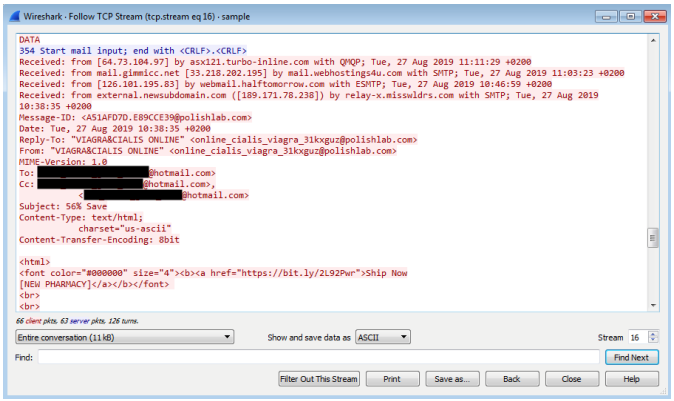



*Figure 5. Example of spam traffic sent through a compromised router*

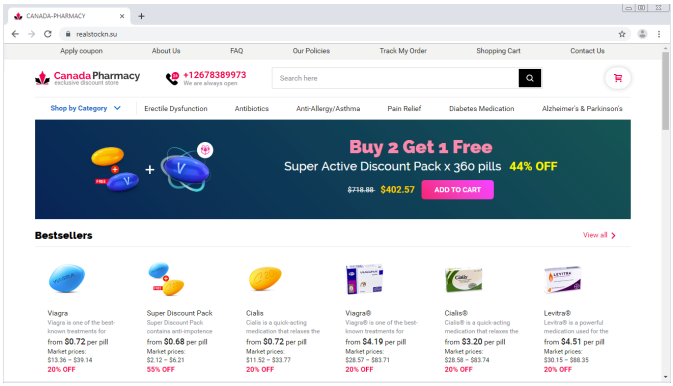



*Figure 6. "Canada Pharmacy" website redirected from spam mail*

Besides the spam traffic, we saw other traffic from a set of remote servers that were repeatedly connecting to Instagram. However, the traffic sent through was protected by HTTPS encryption. We can't decrypt it and don't know what exactly these connections are for. One theory is that it is the password-reuse attack hitting Instagram. It was previously reported to be one type of malicious traffic proxied through the Glupteba botnet.

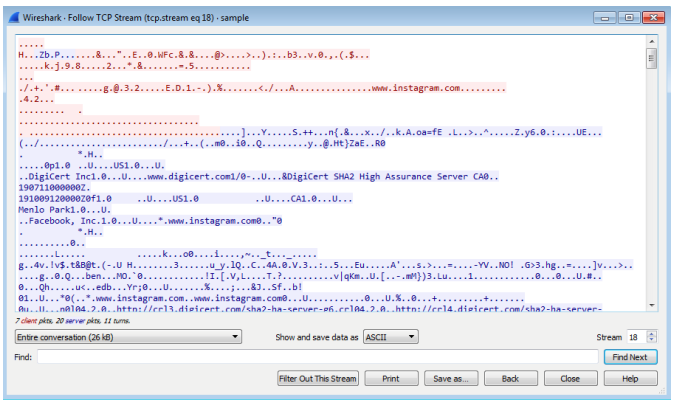



*Figure 7. Example of Instagram connection (HTTPS-encrypted) relayed by a compromised router*

As mentioned, Glupteba still seems to be evolving and adding capabilities. New techniques, such as updating C&C servers through data obtained from Bitcoin transactions, show that the malicious actors behind this malware are adopting little-used techniques to try and keep their malware active. Since it is already proven to be an information stealer and a proxy for malicious spam, users and enterprises should be wary of this threat.

***Security recommendations***

Malvertising is a widespread threat that can affect users and businesses alike. A multilayered approach to security is important — from the gateway, endpoints, networks, and servers. Trend Micro solutions powered by XGen™ security, such as Trend Micro™ Security and Trend Micro Network Defense, can detect related malicious files and URLs and protect users' systems. Trend Micro Smart Protection Suites and Trend Micro Worry-Free™ Business Security, which have behavior monitoring capabilities, can additionally protect from these types of threats by detecting malicious files, as well as blocking all related malicious URLs.

Security should be top of mind when setting up routers — most devices across homes and offices are connected to these devices and can be affected if a router is compromised. Although manufacturers play important roles in securing routers and other devices, users and businesses can adopt good security practices to defend against threats. Also, deploying tools that provide additional security to home networks and devices connected to them further strengthens defenses.

For a full list of the Indicators of Compromise for this malware, please see this document.