



دانشکده مهندسی کامپیوتر

استاد درس: دکتر ابوالفضل دیانت

بهار ۱۴۰۳

پروژه ی ایریدیوم

درس امنیت

گزارش پروژه

ستاره باباجانی – ملیکا محمدی فخار

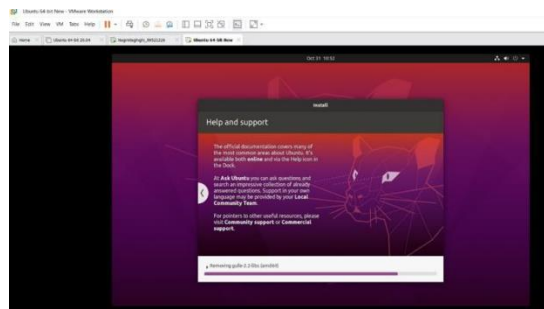
۹۹۵۲۲۰۸۶ – ۹۹۵۲۱۱۰۹

۱ اهداف

در این پروژه قرار است ابتدا یک فایل `bash` نوشته که کل واسطه های شبکه را بشناسد و تمامی IP های موجود در یک کلاس IP را تست کند. سپس شماره پورت باز آن را در یک فایل `csv` ذخیره می کنیم. به `Device` های یافت شده متصل شده و فهرستی از نام کاربری و رمز عبور متداول روی آنها تست میشود. این فهرست از یک فایل `csv` که در کنار برنامه قرار می گیرد، خوانده خواهد شد. پس از اتصال به دستگاه قربانی، فایل `bash` دوم در آن قرار میگیرد. برنامه دوم در دستگاه قربانی در بازه های مشخصی اطلاعات متعددی را به یک خدمت گزار مشخص ارسال می کند. در سمت خدمت گزار، هم یک برنامه مینویسیم که اطلاعات دریافت شده را در یک برنامه تحت وب نمایش دهد.

۲ گام اول

در ابتدا دستگاه قربانی مد نظر را انتخاب میکنیم. برای این کار ما چند ماشین مجازی جدید ساختیم تا در مراحل بعد یکی از آنها به عنوان دستگاه قربانی عمل کند.



شکل ۱: ماشین مجازی

حال در دستگاه اصلی (ریموت) ترمینال باز کرده و یک فایل `bash` به نام `p1.sh` میسازیم و تلاش میکنیم کدی بنویسیم تا یک کلاس IP را از ورودی بگیرد و روی کل IP های آن تست انجام دهد. در ادامه بررسی کند که کدام یک `Active` و پورت `SSH` آن باز می باشد.

پس نیاز است به دستگاه قربانی رفته و از قصد پورت `SSH` آن را باز بگذاریم. برای اینکار ابتدا دستور زیر را اجرا میکنیم:

```
sudo apt install openssh-server
```



که همانطور که میبینید خروجی کامند به صورت زیر به درستی اجرا شده است:

```

negin@ubuntu: ~/Desktop
negin@ubuntu:~/Desktop$ sudo apt install openssh-server
[sudo] password for negin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 libfwupdplugin1 libxmlb1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id

```

شکل ۲: اجرای درست کامند

سپس به فایل کانفیگ در آدرس `/etc/ssh/sshd_config` رفته و با `nano` یا `vim` آن را باز میکنیم و خطی را که در آن عبارت `#Port 22` نوشته شده است را آنکامنت میکنیم. سپس دو کامند زیر را ران کرده تا پورت 22 باز شود:

```

sudo systemctl restart ssh
sudo ufw allow 22

```

حال برنامه `bash` باز کرده و برنامه ای مینویسیم که کلاس IP را از ورودی بگیرد و روی کل IP های آن تست انجام دهد و هرکدام که `Active` بود، با دستور `nmap` بررسی کند که آیا پورت SSH آن نیز باز است یا خیر. با استفاده از دستور `ifconfig` در دستگاه قربانی، میدانیم IP آن برابر 192.168.134.133 است. پس با اجرای برنامه، میبینیم خروجی کاملاً درست است و پورت SSH آن IP باز است:

```

Ip address 192.168.134.129 is inactive
Ip address 192.168.134.130 is inactive
Ip address 192.168.134.131 is inactive
IP address 192.168.134.132 is active
No open SSH port found on 192.168.134.132
IP address 192.168.134.133 is active
Open SSH port(s) on 192.168.134.133: 22
Ip address 192.168.134.134 is inactive
Ip address 192.168.134.135 is inactive

```

شکل ۳: باز بودن پورت SSH

گام دوم

پورت های باز هر IP در گام قبل پیدا شد و حال کافیهست پس از یافتن شماره پورت باز، آنها را در فایل `csv` ذخیره کنیم. در برنامه نوشته شده، ما نتیجه پورت های باز را در فایل `csv` به نام `ssh_results` ریخته ایم.

گام سوم

حال سعی میکنیم به Device های یافت شده متصل شویم. برای انجام این کار، ما یک فایل به نام `common_passwords.csv` تهیه کرده ایم که در آن چند نام کاربر و رمز عبور متداول را ذخیره کرده ایم. برای مثال، نام کاربری `Admin` و رمز 1234 جزو یکی از پرکاربرترین رمزها است. همانطور که داخل کد میبینید، فایل `csv` خوانده شده و تمام نام کاربری ها و رمزهای داخل آن، روی دستگاه قربانی با استفاده از کامند زیر تست شده اند:

```

"sshpass -p "$password" ssh "$username@$ip_address

```

گام چهارم و پنجم

پس از اینکه تمام نام کاربری و رمزها تست شدند و با موفقیت به دستگاه قربانی متصل شدیم، باید مقداری اطلاعات مفید را مانند IP آن، مدل CPU و اطلاعات memory و System را استخراج کنیم. برای اینکار از دستورات مهمی مانند lscpu برای اطلاعات cpu و مدل آن استفاده می کنیم. که با کمک دستور head و tail آن اطلاعاتی که بنظرمان مهم تر است را استخراج می کنیم. تمام اطلاعات بدست آمده از این دستورات در فایل به نام info.txt ذخیره میشود.

گام ششم

در این گام قصد داریم اطلاعات استخراج شده از دستگاه قربانی را در قالب یک جدول نمایش دهیم. برای این کار از بک و فرانت استفاده میکنیم.

فلذا یک پروژه جنگو میسازیم. در این پروژه یک مدل می سازیم که شامل IP و username و cpu model و memory info و system info میباشد. در ادامه API گت و پست آماده می کنیم که از گت برای نمایش اطلاعات و از پست ذخیره ی اطلاعات استفاده خواهد شد.

```
4 class Device(models.Model):
5     IP_Address = models.CharField(max_length=1024)
6     UserName = models.CharField(max_length=1024)
7     CPU_model = models.CharField(max_length=1024)
8     Memory_info = models.CharField(max_length=1024)
9     System_info = models.CharField(max_length=1024)
10     def __str__(self):
11         return self.CPU_model
```

شکل ۴: نمونه ی مدل در پروژه

System information	Memory information	CPU model
lpr 13 09:27:15 UTC 2021 x86_64 x86_64 GNU/Linux	4015896	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz

شکل ۵: بخشی از خروجی پروژه

CPU model	Username	IP Address
11th Gen Intel® Core™ i5-1135G7 @ 2.40GHz	Melika	192.168.134.133

شکل ۶: بخشی دیگر از خروجی پروژه

توجه: با توجه به اینکه لپتاپی که در اختیار داشتیم، ۸ گیگ رم بیشتر نداشت و امکان بالا آوردن ماشین مجازی روی آن میسر نبود، از ماشین مجازی بر روی لپتاپ یکی از دوستان استفاده کردیم.