

Michał Wawrzyńczak

Zadanie 1

Pierw postawiłem wirtualne maszyny z systemami Windows i Security0nion.

Następnie pobrałem Sysmon'a i zainstalowałem na hoscie Windows.

```
C:\Users\user\Downloads\Sysmon>sysmon.exe -accepteula -i config.xml

System Monitor v10.42 - System activity monitor
Copyright (C) 2014-2019 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.22
Sysmon schema version: 4.23
Configuration file validated.
Sysmon installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon.....
Sysmon started.

C:\Users\user\Downloads\Sysmon>
```

Następnie pobrałem z internetu przykładowy config i skonfigurowałem Sysmona.

```
sysmon.exe -accepteula -c config.xml
```

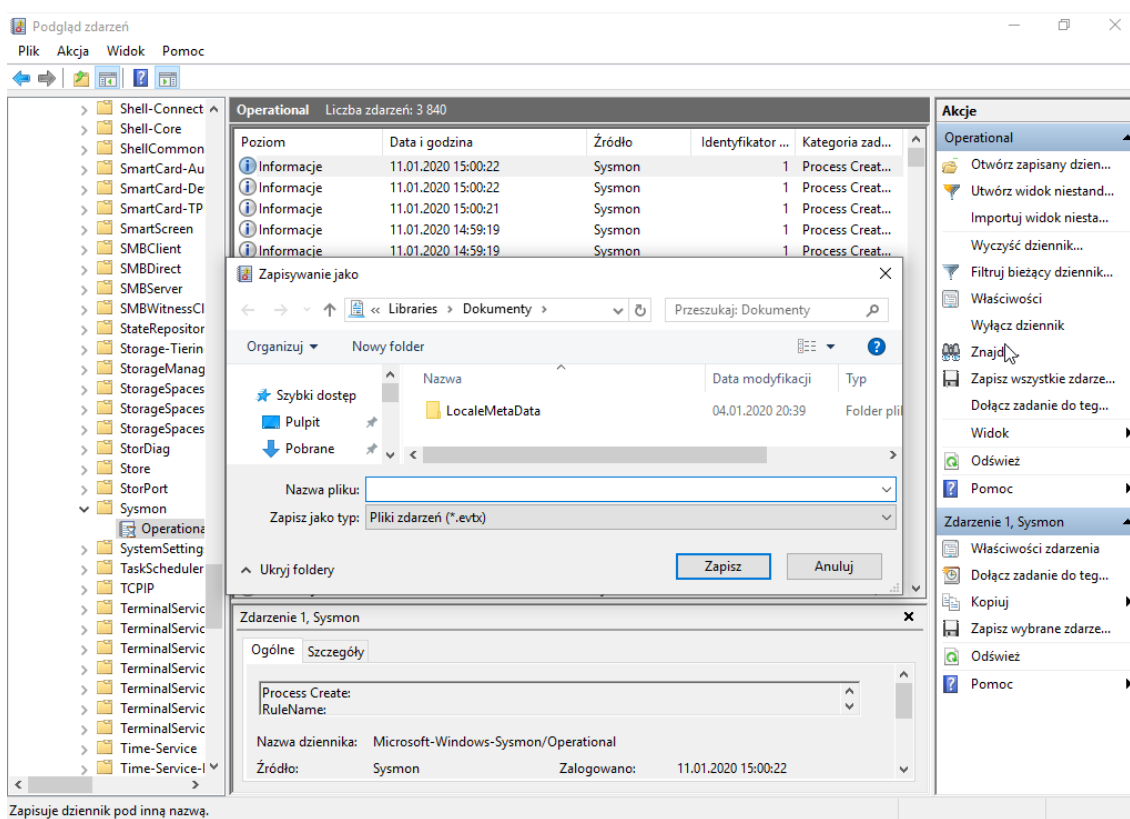
```
C:\Users\user\Downloads\Sysmon>sysmon -c

System Monitor v10.42 - System activity monitor
Copyright (C) 2014-2019 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Current configuration:
- Service name: Sysmon
- Driver name: SysmonDrv
- HashingAlgorithms: MD5,SHA256
- Network connection: enabled
- Image loading: disabled
- CRL checking: enabled
- Process Access: disabled

Rule configuration (version 4.22):
- ProcessCreate
  ParentCommandLine filter: is value: '"C:\Program Files\Microsoft Monitoring Agent\Agent\MonitoringHost.exe" -Embedding'
  CommandLine filter: begin with value: '"C:\Windows\system32\wermgr.exe" "-queuereporting_svc"'
  CommandLine filter: begin with value: 'C:\Windows\system32\DllHost.exe /Processid'
  CommandLine filter: begin with value: 'C:\Windows\system32\wbem\wmiprvse.exe -Embedding'
  CommandLine filter: begin with value: 'C:\Windows\system32\wbem\wmiprvse.exe -secured -Embedding'
  CommandLine filter: is value: 'C:\Windows\system32\wermgr.exe -upload'
```

Po uruchomieniu usługi mogłem już podejrzeć logi w programie EventViwer, a także zapisac je do pliku



Następnie pobrałem program winlogbeat, aby skonfigurować automatyczne wysyłanie logów do hosta SecurityOnion i odebranie ich w Kibanie

```
PowerShell.exe -ExecutionPolicy UnRestricted -File .\install-service-winlogbeat.ps1
```

```
PS C:\Users\User\Downloads\winlogbeat-7.5.1-windows-x86_64\winlogbeat-7.5.1-windows-x86_64> PowerShell.exe -ExecutionPolicy UnRestricted -File .\install-service-winlogbeat.ps1
```

| Status | Name | DisplayName |
|---------|------------|-------------|
| Stopped | winlogbeat | winlogbeat |

Ustawiłem potrzebne dane w pliku konfiguracyjnym winlogbeat'a po czym uruchomiem usługę.

-Kibana

```
setup.kibana:
host: [https://192.168.12.10/app/kibana]
```

-Elasticsearch

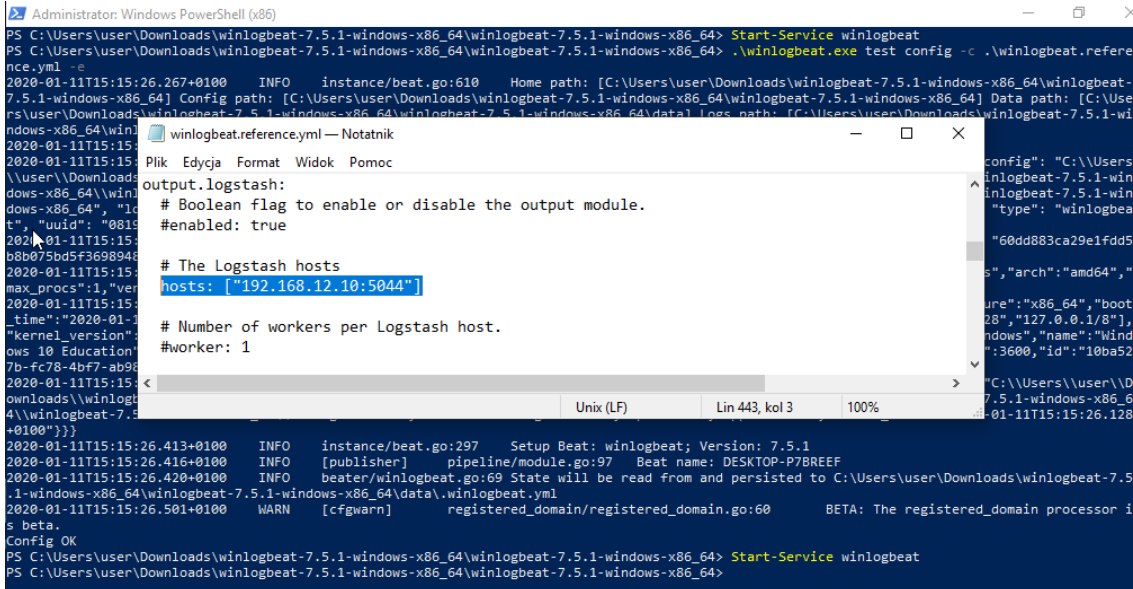
```
#output.elasticsearch:
#hosts: ["192.168.12.10:9200"]
```

-Logstash

```
output.logstash:
hosts: ["192.168.12.10:5044"]
```

-Sprawdzenie poprawności configu

```
.\winlogbeat.exe test config -c .\winlogbeat.reference.yml -e
```



```
Administrator: Windows PowerShell (x86)
PS C:\Users\user\Downloads\winlogbeat-7.5.1-windows-x86_64\winlogbeat-7.5.1-windows-x86_64> Start-Service winlogbeat
PS C:\Users\user\Downloads\winlogbeat-7.5.1-windows-x86_64\winlogbeat-7.5.1-windows-x86_64> .\winlogbeat.exe test config -c .\winlogbeat.reference.yml -e
2020-01-11T15:15:26.267+0100 INFO instance/beat.go:610 Home path: [C:\Users\user\Downloads\winlogbeat-7.5.1-windows-x86_64\winlogbeat-7.5.1-windows-x86_64] Config path: [C:\Users\user\Downloads\winlogbeat-7.5.1-windows-x86_64\winlogbeat-7.5.1-windows-x86_64] Data path: [C:\Users\user\Downloads\winlogbeat-7.5.1-windows-x86_64\winlogbeat-7.5.1-windows-x86_64\data] Logs path: [C:\Users\user\Downloads\winlogbeat-7.5.1-windows-x86_64\winlogbeat-7.5.1-windows-x86_64\data\logs]
winlogbeat.reference.yml — Notatnik
Plik Edycja Format Widok Pomoc
output.logstash:
  # Boolean flag to enable or disable the output module.
  #enabled: true

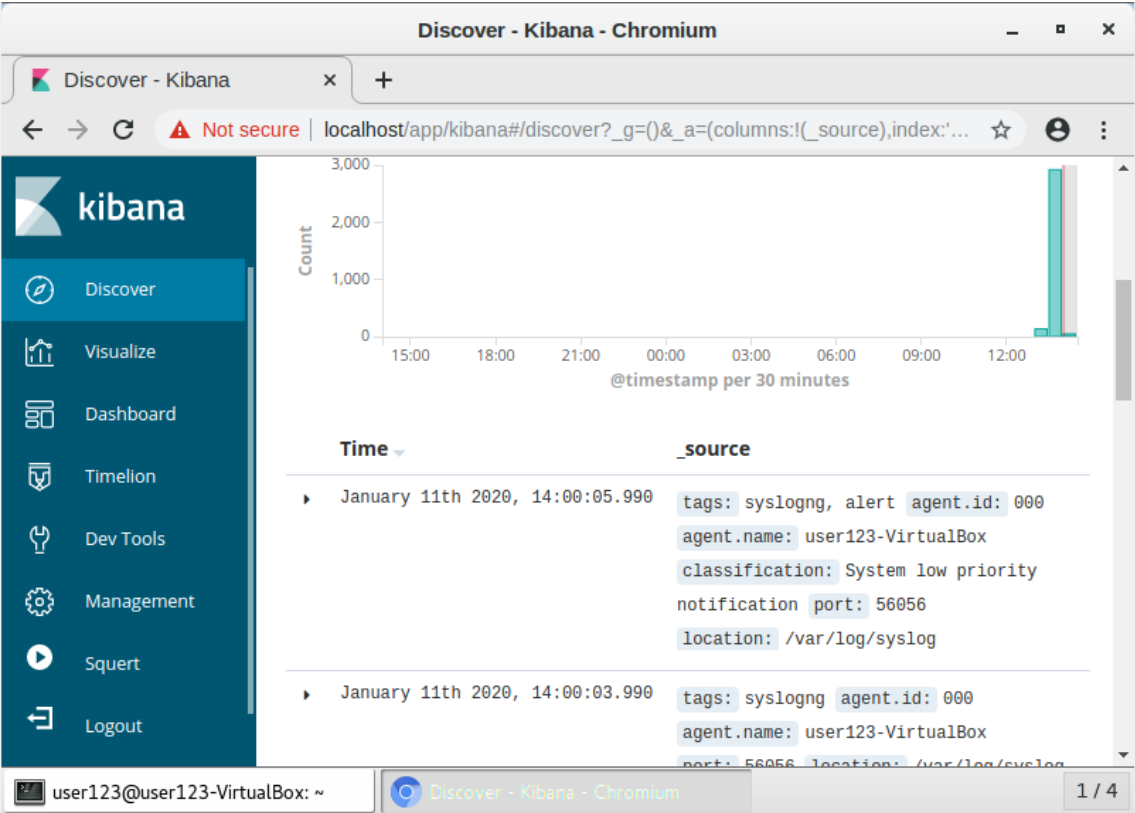
  # The Logstash hosts
  hosts: ["192.168.12.10:5044"]

  # Number of workers per Logstash host.
  #worker: 1

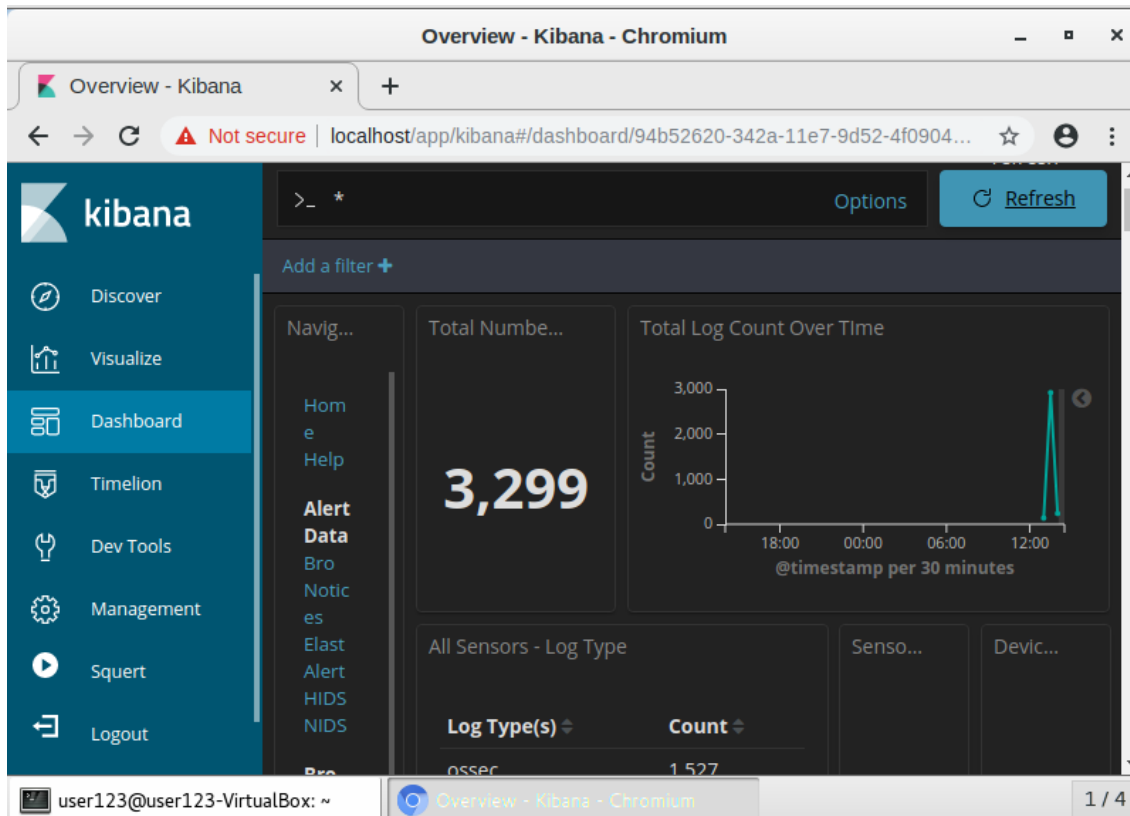
2020-01-11T15:15:26.413+0100 INFO instance/beat.go:297 Setup Beat: winlogbeat; Version: 7.5.1
2020-01-11T15:15:26.416+0100 INFO [publisher] pipeline/module.go:97 Beat name: DESKTOP-P7BREEF
2020-01-11T15:15:26.420+0100 INFO beater/winlogbeat.go:69 State will be read from and persisted to C:\Users\user\Downloads\winlogbeat-7.5.1-windows-x86_64\winlogbeat-7.5.1-windows-x86_64\data\winlogbeat.yml
2020-01-11T15:15:26.501+0100 WARN [cfgwarn] registered_domain/registered_domain.go:60 BETA: The registered_domain processor is beta.
Config OK
PS C:\Users\user\Downloads\winlogbeat-7.5.1-windows-x86_64\winlogbeat-7.5.1-windows-x86_64> Start-Service winlogbeat
PS C:\Users\user\Downloads\winlogbeat-7.5.1-windows-x86_64\winlogbeat-7.5.1-windows-x86_64>
```

Teraz mogłem przejść już do SecurityOniona i sprawdzić czy logi z Windowsa są dostarczane. Uruchomiłem Kibane i w zakładce Discovery po wybraniu danych z logstash, zaobserwowałem, że logi z Windowsa nie są odbierane. Postanowiłem więc wyłączyć

firewalle na obu maszynach, pocz ym mogłem już zaobserwować odebrane logi.

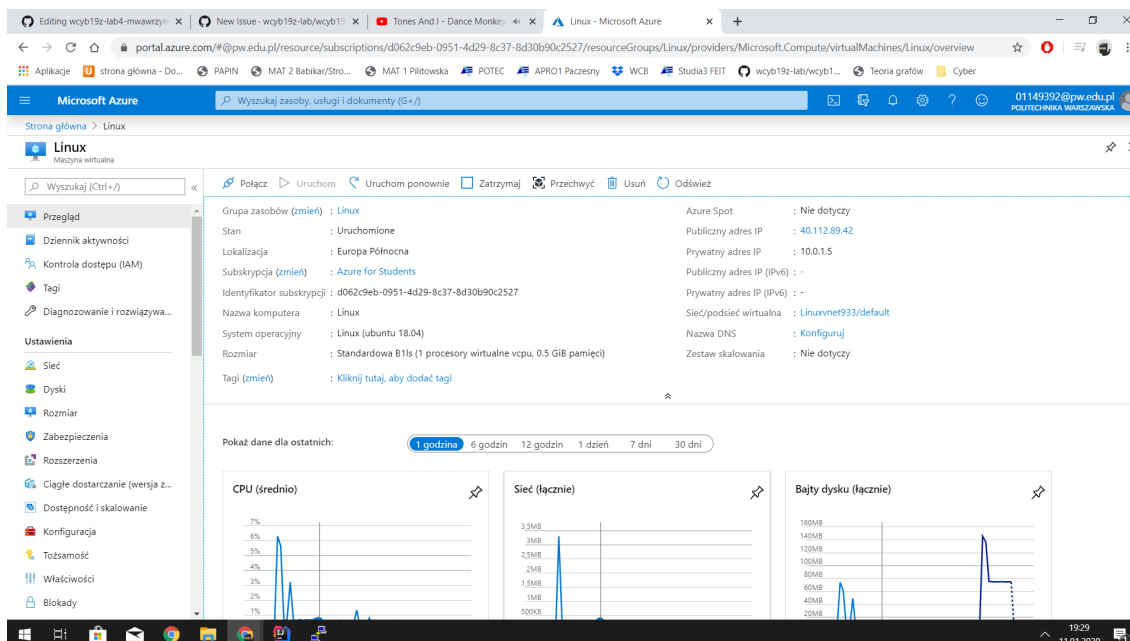


Teraz, można już przeglądać i analizować logi systemowe z Windowsa.



Zadanie 2

Utworzyłem maszynę wirtualną z systemem operacyjnym Linux w chmurze Azure



Przy pomocy programu PuTTY używając protokołu ssh zalogowałem się, uzyskując dostęp do terminala maszyny

```
user123@Linux: ~  
login as: user123  
user123@13.69.187.188's password:  
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-1027-azure x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Sat Jan 11 17:37:18 UTC 2020  
  
System load:  0.3               Processes:            109  
Usage of /:   4.0% of 28.90GB   Users logged in:     0  
Memory usage: 77%              IP address for eth0: 10.0.1.4  
Swap usage:   0%  
  
0 packages can be updated.  
0 updates are security updates.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

**Następnie używając następujących poleceń **

```
sudo iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT  
sudo iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT  
sudo iptables -A OUTPUT -p tcp -m tcp --dport 80 -j ACCEPT  
sudo iptables -A OUTPUT -p tcp -m tcp --dport 443 -j ACCEPT
```

```
user123@Linux:~$ sudo iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT  
user123@Linux:~$ sudo iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT  
user123@Linux:~$ sudo iptables -A OUTPUT -p tcp -m tcp --dport 80 -j ACCEPT  
user123@Linux:~$ sudo iptables -A OUTPUT -p tcp -m tcp --dport 443 -j ACCEPT  
user123@Linux:~$ sudo iptables -S  
-P INPUT ACCEPT  
-P FORWARD ACCEPT  
-P OUTPUT ACCEPT  
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT  
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT  
-A OUTPUT -p tcp -m tcp --dport 80 -j ACCEPT  
-A OUTPUT -p tcp -m tcp --dport 443 -j ACCEPT  
user123@Linux:~$
```

Używając kolejnego polecenia dopuściłem ruch polecenia

```
sudo iptables -A INPUT -p tcp -s 185.49.203.47 -m tcp --dport 22 -j ACCEPT  
sudo iptables -A OUTPUT -p tcp -s 185.49.203.47 -m tcp --dport 22 -j ACCEPT
```

```
user123@Linux:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -s 185.49.203.47/32 -p tcp -m tcp --dport 22 -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A OUTPUT -s 185.49.203.47/32 -p tcp -m tcp --dport 22 -j ACCEPT
user123@Linux:~$
```

Wykorzystując kolejne 2 polecenia zablokowałem komunikację na nieużywanych portach

```
sudo iptables -P INPUT DROP
sudo iptables -P OUTPUT DROP
```

Sprawdziłem jakie porty są konieczne do komunikacji w protokole MQTT a następnie użyłem poleceń i odblokowałem ruch na tych portach

```
sudo iptables -A INPUT -p tcp -m tcp --dport 1883 -j ACCEPT
sudo iptables -A INPUT -p tcp -m tcp --dport 8883 -j ACCEPT
sudo iptables -A OUTPUT -p tcp -m tcp --dport 1883 -j ACCEPT
sudo iptables -A OUTPUT -p tcp -m tcp --dport 8883 -j ACCEPT
```

```
user123@Linux:~$ sudo iptables -A INPUT -p tcp -m tcp --dport 1883 -j ACCEPT
user123@Linux:~$ sudo iptables -A INPUT -p tcp -m tcp --dport 8883 -j ACCEPT
user123@Linux:~$ sudo iptables -A OUTPUT -p tcp -m tcp --dport 1883 -j ACCEPT
user123@Linux:~$ sudo iptables -A OUTPUT -p tcp -m tcp --dport 8883 -j ACCEPT
user123@Linux:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -s 185.49.203.47/32 -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 1883 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 8883 -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A OUTPUT -s 185.49.203.47/32 -p tcp -m tcp --dport 22 -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 1883 -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 8883 -j ACCEPT
user123@Linux:~$
```