

Sprawozdanie_1 WCYB_Lab_4

Marcin Dadura nr: 303_688

Do realizacji zadania należy:

Zadanie 1

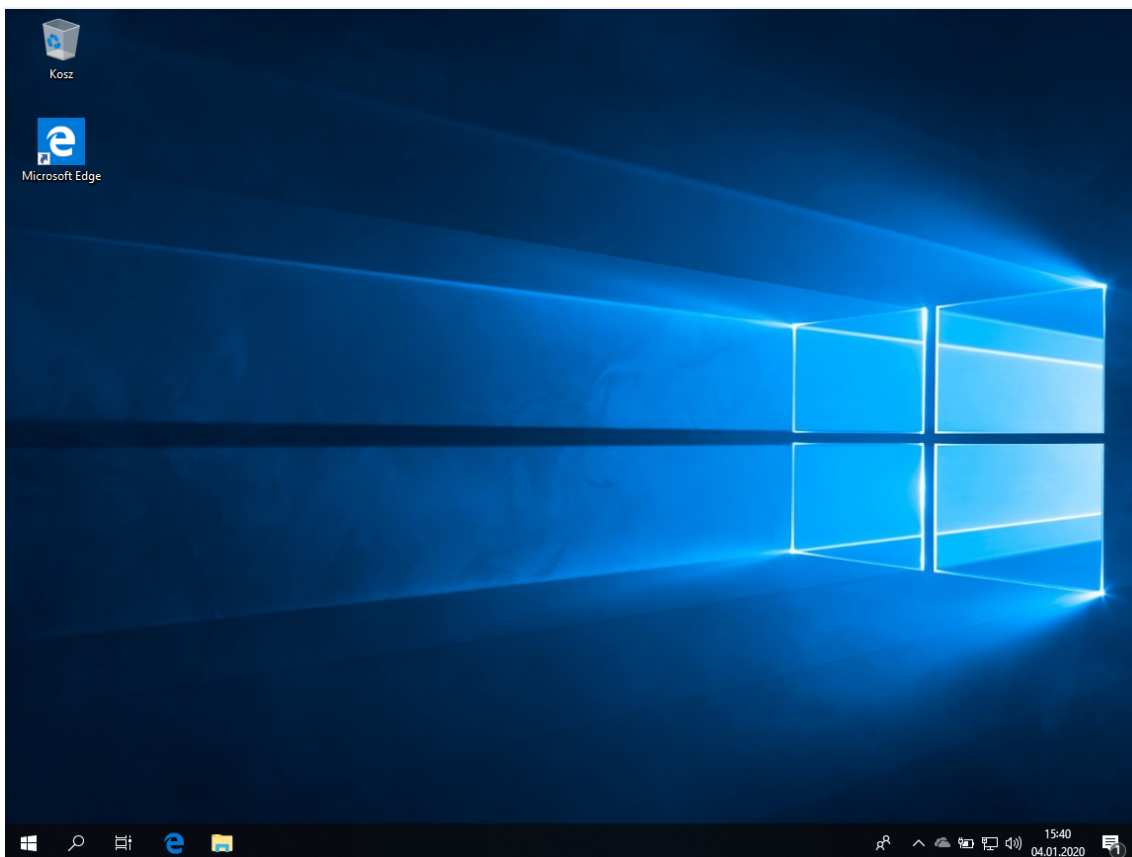
Do realizacji zadania należy:

a) zainstalować wirtualny host Security Onion zgodnie z tym, co wykonawano podczas laboratorium. b) zainstalować wirtualny host Windowsa 10 w wersji Education/Professional/Enterprise - licencja jest dostępna w Azure for Students (sekcja Education) c) Skonfigurować generowanie logów systemowych systemu Windows - Sysmon. W tym kroku może być przeprowadzone to testowo - do pliku. d) Skonfigurować wysyłanie logów sysmon do Security Onion. e) Zaobserwować działanie za pomocą UI dostępnego w Security Onion - Kibana. f) Przeanalizować zawartość informacyjną logów sysmon pod kątem wykrywania zagrożeń w cyberprzestrzeni.

a),b) Zainstalować wirtualny host Security Onion zgodnie z tym, co wykonawano podczas laboratorium, zainstalować wirtualny host Windowsa 10 w wersji Education/Professional/Enterprise - licencja jest dostępna w Azure for Students (sekcja Education)

Po 3 próbach instalacji Security Onion i Win10 zaczęły działać. Korzystam z licencji dostępnej na [Azure](#). Zainstalowana wersja systemu operacyjnego Windows to Education 64-bit.





c) Skonfigurować generowanie logów systemowych systemu Windows - Sysmon. W tym kroku może być przeprowadzone to testowo - do pliku.

Sysmon pobrałem ze strony [Microsot](#). Do konfiguracji użyłem pliku .xml z [Gita](#).

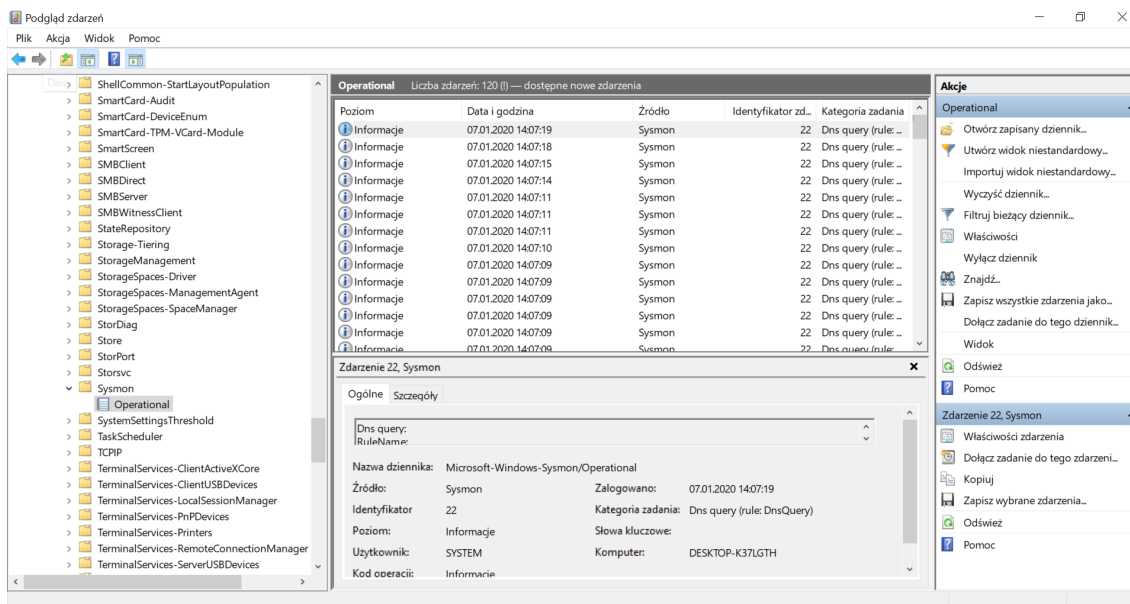
Do instalacji Sysmon użyłem

```
sysmon.exe -accepteula -i sysmonconfig-export.xml
```

gdzie `sysmonconfig-export.xml` to plik konfiguracyjny, który zamieszczam w repozytorium. Do update-owania Sysmon użyłem `sysmon.exe -c sysmonconfig-export.xml`.

Sysmon skonfigurowałem testowo, aby odczytać logi należy wejść w:

- Podgląd zdarzeń
- Dziennik aplikacji i usług
- Microsoft
- Windows
- Sysmon



d) Skonfigurować wysyłanie logów sysmon do Security Onion.

Do wysyłania logów do SecurityOnion-a użyłem programu `winlogbeat`, ze strony <https://www.elastic.co/downloads/beats/winlogbeat>. Do instalacji użyłem komendy wpisanej do Powershella : `PowerShell.exe -ExecutionPolicy UnRestricted -File .\install-service-winlogbeat.ps1`.

```
PS C:\Users\Windows10Edu\Desktop\winlogbeat\winlogbeat-7.5.1-windows-x86_64> PowerShell.exe -ExecutionPolicy UnRestricted -File .\install-service-winlogbeat.ps1

Security warning
Run only scripts that you trust. While scripts from the internet can be useful, this script can potentially harm your
computer. If you trust this script, use the Unblock-File cmdlet to allow the script to run without this warning
message. Do you want to run
C:\Users\Windows10Edu\Desktop\winlogbeat\winlogbeat-7.5.1-windows-x86_64\install-service-winlogbeat.ps1?
[D] Do not run [R] Run once [S] Suspend [?] Help (default is "D"): r

Status  Name      DisplayName
-----
Stopped winlogbeat winlogbeat
```

W pliku konfiguracyjnym `winlogbeat.reference.yml` skonfigurowałem następujące rzeczy:

- Dla kibana

```
setup.kibana:
host: [https://192.168.56.108/app/kibana]
```

- Zakomentujemy część z elasticsearch

```
#output.elasticsearch:
#hosts: ["192.168.56.108:9200"]
```

- Dla logstash

```
output.logstash:
hosts: ["192.168.56.108:5044"]
```

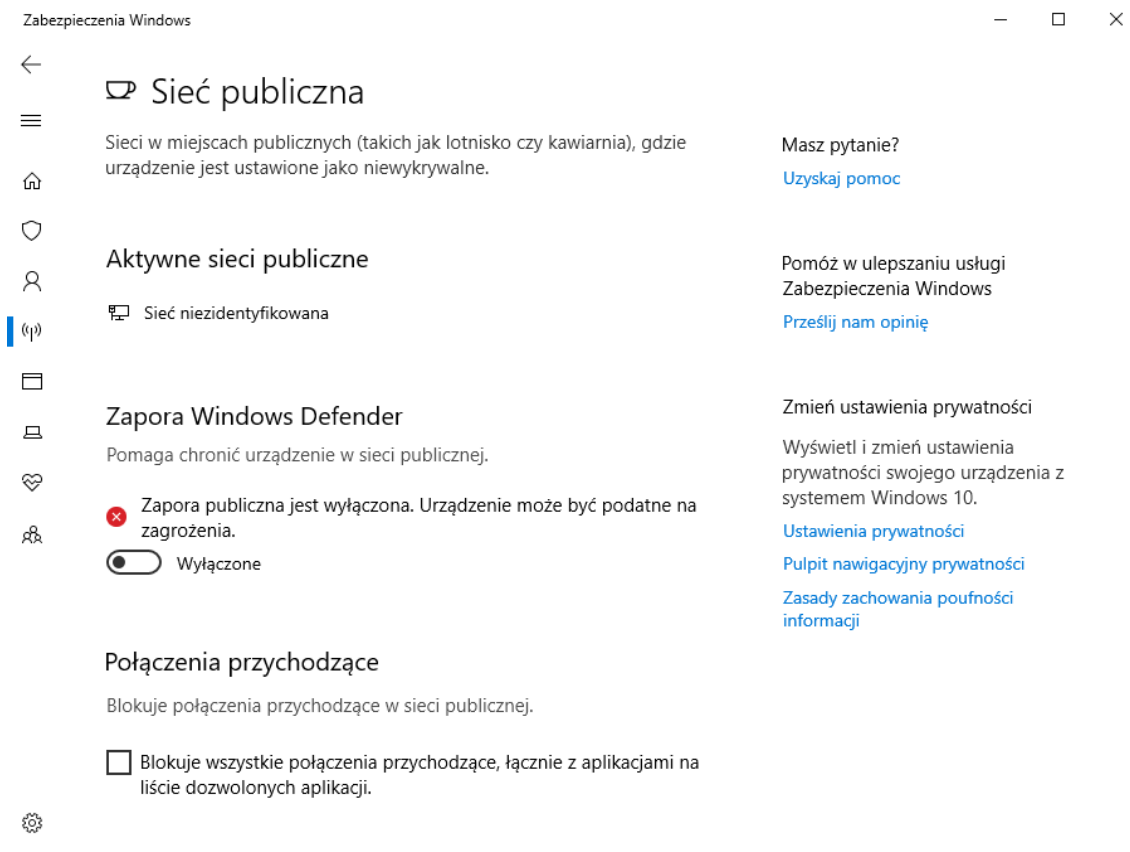
Następnie sprawdzamy poprawność konfiguracji winlogbeat -a poprzez komendę(Config OK oznacza, że konfiguracja przeszła pomyślnie):

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

```
PS C:\Users\Windows10Edu\Desktop\winlogbeat\winlogbeat-7.5.1-windows-x86_64> .\winlogbeat.exe test config -c .\winlogbeat.reference.yml -e
2020-01-08T23:11:13.015+0100 INFO instance/beat.go:610 Home path: [C:\Users\Windows10Edu\Desktop\winlogbeat\winlogbeat-7.5.1-windows-x86_64] Config path: [C:\Users\Windows10Edu\Desktop\winlogbeat\winlogbeat-7.5.1-windows-x86_64] Data path: [C:\Users\Windows10Edu\Desktop\winlogbeat\winlogbeat-7.5.1-windows-x86_64\data] Logs path: [C:\Users\Windows10Edu\Desktop\winlogbeat\winlogbeat-7.5.1-windows-x86_64\logs]
2020-01-08T23:11:13.180+0100 INFO instance/beat.go:618 Beat ID: b9b5dd55-c366-46b7-83cf-138c015137cd
2020-01-08T23:11:13.182+0100 INFO [beat] instance/beat.go:941 Beat info {"system_info": {"beat": {"path": "C:\\Users\\Windows10Edu\\Desktop\\winlogbeat\\winlogbeat-7.5.1-windows-x86_64", "data": "C:\\Users\\Windows10Edu\\Desktop\\winlogbeat\\winlogbeat-7.5.1-windows-x86_64\\data", "home": "C:\\Users\\Windows10Edu\\Desktop\\winlogbeat\\winlogbeat-7.5.1-windows-x86_64", "logs": "C:\\Users\\Windows10Edu\\Desktop\\winlogbeat\\winlogbeat-7.5.1-windows-x86_64\\logs"}, "type": "winlogbeat", "uuid": "b9b5dd55-c366-46b7-83cf-138c015137cd"}}}
2020-01-08T23:11:13.186+0100 INFO [beat] instance/beat.go:950 Build info {"system_info": {"build": {"commit": "60dd883ca29e1fdd5b8b075bd5f3698948b1d44d", "libbeat": "7.5.1", "time": "2019-12-16T22:05:28.000Z", "version": "7.5.1"}}}
2020-01-08T23:11:13.187+0100 INFO [beat] instance/beat.go:953 Go runtime info {"system_info": {"go": {"os": "windows", "arch": "amd64", "max_procs": 1, "version": "go1.12.12"}}}
2020-01-08T23:11:13.213+0100 INFO [beat] instance/beat.go:957 Host info {"system_info": {"host": {"architecture": "x86_64", "boot_time": "2020-01-08T22:49:37.86+01:00", "name": "DESKTOP-ERILMFH", "ip": [{"fe80::acd6:74ea:7d12:dd78/64", "192.168.56.106/24"}, {"1:128", "127.0.0.1/8"}], "kernel_version": "10.0.17763.107 (WinBuild.160101.0800)", "mac": [{"08:00:27:40:62:f6"}], "os": {"family": "windows", "platform": "windows", "name": "Windows 10 Education", "version": "10.0", "major": 10, "minor": 0, "patch": 0, "build": "17763.107"}, "timezone": "CET", "timezone_offset_sec": 3600, "id": "ffc4c2d5-25f4-4860-bb34-ee436792ce23"}}}
2020-01-08T23:11:13.235+0100 INFO [beat] instance/beat.go:986 Process info {"system_info": {"process": {"cwd": "C:\\Users\\Windows10Edu\\Desktop\\winlogbeat\\winlogbeat-7.5.1-windows-x86_64", "exe": "C:\\Users\\Windows10Edu\\Desktop\\winlogbeat\\winlogbeat-7.5.1-windows-x86_64\\winlogbeat.exe", "name": "winlogbeat.exe", "pid": 4864, "ppid": 1480, "start_time": "2020-01-08T23:11:12.362+0100"}}}
2020-01-08T23:11:13.244+0100 INFO instance/beat.go:297 Setup Beat: winlogbeat; Version: 7.5.1
2020-01-08T23:11:13.247+0100 INFO [publisher] pipeline/module.go:97 Beat name: DESKTOP-ERILMFH
2020-01-08T23:11:13.248+0100 INFO beater/winlogbeat.go:69 State will be read from and persisted to C:\Users\Windows10Edu\Desktop\winlogbeat\winlogbeat-7.5.1-windows-x86_64\data\winlogbeat.yml
2020-01-08T23:11:13.375+0100 WARN [cfgwarn] registered_domain/registered_domain.go:60 BETA: The registered_domain processor is beta.
Config OK
```

Następnie wyłączamy Firewall-a w:

- windwos

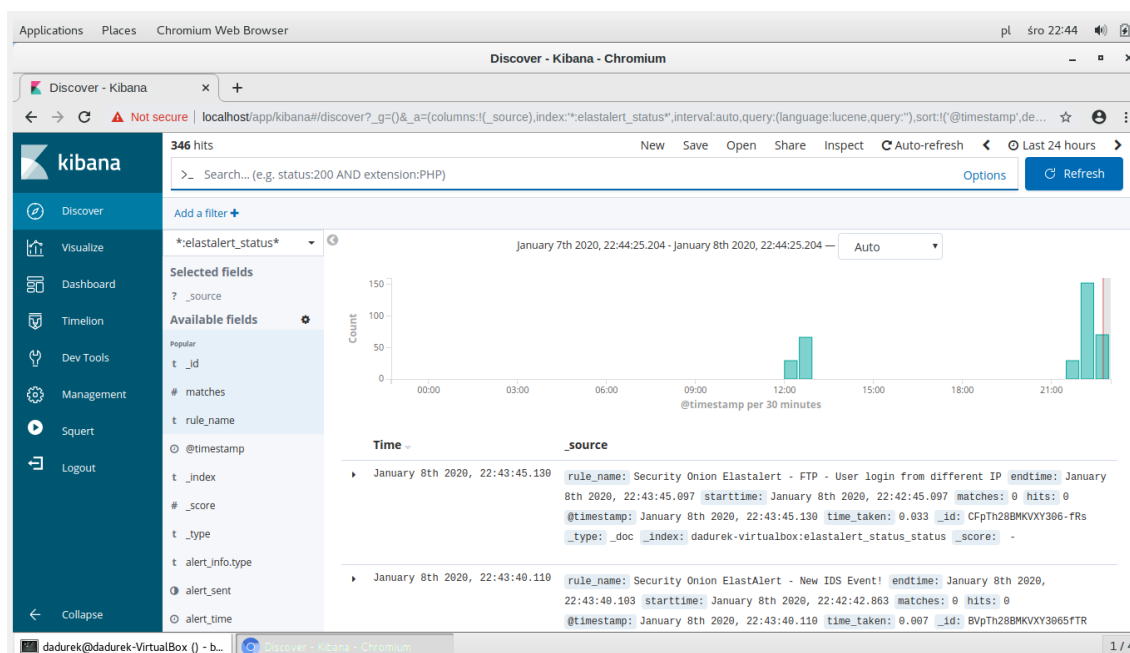


- Onion używamy komendy: `sudo ufw disable`.

Kolejnym punktem jest uruchomienie usługi `winlogbeat` w Windows w PowerShell poprzez komendę: `Start-Service winlogbeat`.

e) Zaobserwować działanie za pomocą UI dostępnego w Security Onion - Kibana.

Po odpaleniu Kibana -y i wejściu w zakładkę `Discover` możemy zaobserwować wysyłanie logów:

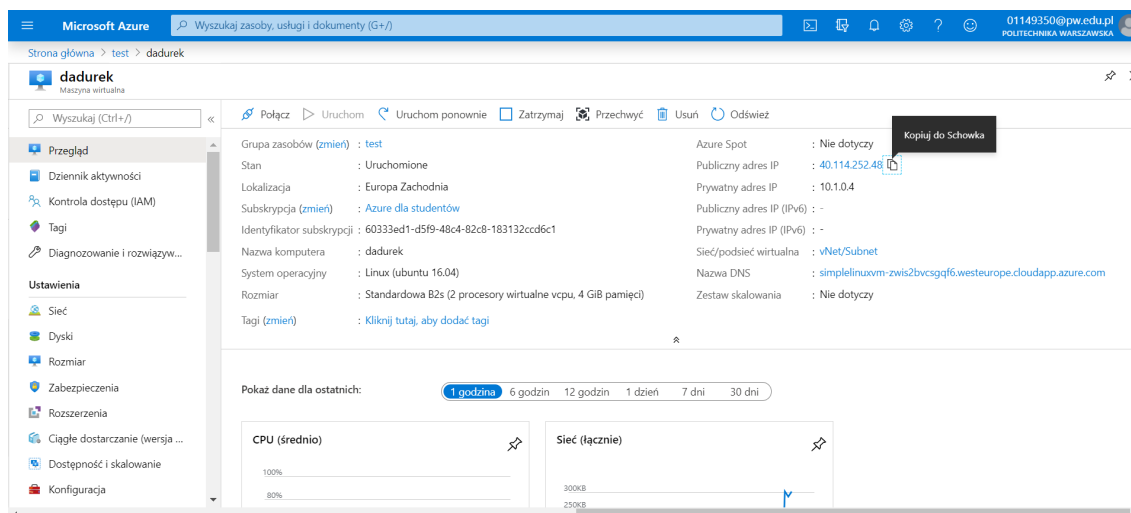


f) Przeanalizować zawartość informacyjną logów sysmon pod kątem wykrywania zagrożeń w cyberprzestrzeni.

Zadanie 2

1) W ramach możliwości konta Azure for Students ustanowić darmową maszynę wirtualną z systemem operacyjnym Linux.

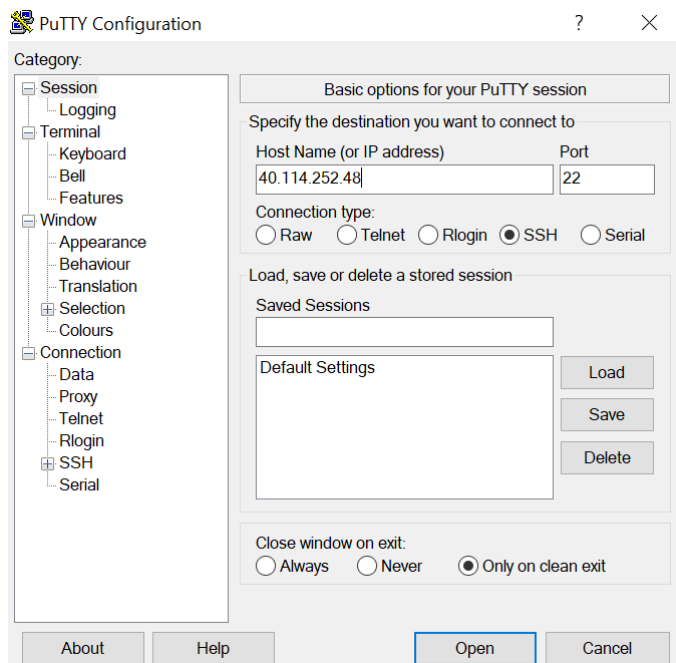
Stworzyłem vm na portalu Azzure.



2) Sonfigurować reguły firewalla:

a) dopuścić ruch na porcie 80 oraz 443 (HTTP) z dowolnej maszyny b) dopuścić ruch dla usługi SSH tylko ze swojej maszyny (swój adres publiczny IP można znaleźć np. na stronie: <https://www.myip.com>) c) zablokować wszystkie nieużywane porty d) dopuścić ruch dla protokołu MQTT (sprawdzić, co jest potrzebne)

Do zalogowania się do maszyny virtualnej użyłem programu PuTTY. Logowanie przez PuTTY wygląda następująco. Wpisuję IP hosta virtualnej maszyny oraz podaje protokół SSH, port 22. Firewall wirtualnej maszyny ma zdefiniowaną podczas tworzenia możliwość logowania się przez SSH.



Po zalogowaniu się używając loginu oraz hasła mam dostęp do terminala systemu.

```
dadurek@dadurek: ~  
login as: dadurek  
dadurek@40.114.252.48's password:  
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-1066-azure x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
0 packages can be updated.  
0 updates are security updates.  
  
New release '18.04.3 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Tue Jan  7 20:02:10 2020 from 89.64.18.75  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
dadurek@dadurek:~$
```

a) Dopuszczać ruch na porcie 80 oraz 443 (HTTP) z dowolnej maszyny.

Aby sprawdzić aktualne zasady firewalla należy wpisać komendę `iptables -S`.

```
dadurek@dadurek: ~  
dadurek@dadurek:~$ sudo iptables -S  
-P INPUT ACCEPT  
-P FORWARD ACCEPT  
-P OUTPUT ACCEPT  
dadurek@dadurek:~$
```

Aby dopuścić ruch na należy użyć komend:

- na porcie 80: `sudo iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT`
- na porcie 443: `sudo iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT`

```
dadurek@dadurek:~$ sudo iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT  
dadurek@dadurek:~$ sudo iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT  
dadurek@dadurek:~$ 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  
dadurek@dadurek:~$
```

Aby sprawdzić, jakie protokoły są aktualnie akceptowane należy użyć komendy `sudo iptables -L -n`. Załączonego poniżej screena wynika, że ruch na tych portach został dozwolony.


```
dadurek@dadurek:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     tcp  --  0.0.0.0/0              0.0.0.0/0          tcp dpt:80
ACCEPT     tcp  --  0.0.0.0/0              0.0.0.0/0          tcp dpt:443

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
dadurek@dadurek:~$
```

b) Dopuszczyć ruch dla usługi SSH tylko ze swojej maszyny (swój adres publiczny IP można znaleźć np. na stronie: <https://www.myip.com>)

Adres publiczny maszyny, na której się znajduję to: 162.158.103.131

Aby dopuścić ruch dla usługi SSH tylko ze swojej maszyny należy użyć komendy: `sudo iptables -A INPUT -p tcp -s 162.158.103.131 -m tcp --dport 22 -j ACCEPT`.

```
dadurek@dadurek:~$ sudo iptables -A INPUT -p tcp -s 162.158.103.131 -m tcp --dport 22 -j ACCEPT
dadurek@dadurek:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -s 162.158.103.131/32 -p tcp -m tcp --dport 22 -j ACCEPT
dadurek@dadurek:~$
```

c) zablokować wszystkie nieużywane porty

Aby zablokować całą resztę ruchu sieciowego należy użyć komend:

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

d) dopuścić ruch dla protokołu MQTT (sprawdzić, co jest potrzebne)

Aby dopuścić ruch dla protokołu MQTT należy zezwolić na ruch na porcie 1883 oraz 8883. Należy zrobić to komendami:

- `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`

Po konfiguracji firewall-a należy zapisać konfigurację komendą: `sudo iptables-save | sudo tee /etc/sysconfig/iptables`

A następnie zrestartować komendą: `sudo service iptables restart`

3) Znaleźć best practices hardeningu serwera Linux. Następnie przeprowadzić procedurę hardeningu maszyny w Azure. Obowiązkowo uwzględnić:

- SSH certificates logins
- Fail2ban
- oraz wybrać 2 inne dowolne działania prowadzące go hardeningu systemu. Uzasadnić wybór.

Best practices hardeningu serwera linux:

- używać silnych loginów oraz haseł (minimum 8 znaków, w tym duże znaki i znaki specjalne)
- dezaktywować logowanie na roota przez ssh

- zmniejszyć ilość użytkowników z możliwością zdalnego dostępu
- używanie niestandardowego portu dla SSH, zamiast standardowego 22
- limitowanie dostępu do ssh przez sprecozywanie dokładnego adresu IPz którego chcemy mieć zdalny dostęp (punkt b z poprzedniego zadania)
- ustawienie czasu, po którym sesja zakańcza się po braku aktywności (Idle Timeout Interval)
- używać klucza do autoryzacji ssh(SSH certificates logins)
- zablokować próby bruteforce-owania haseł do SSH poprzez Fail2Ban
- zablokować porty których nie używamy
- zablokować możliwość wysyłania flag, w tym pingowania

Konfiguracja klucza SSH:

Należy wpisać komendę `ssh-keygen`

```
dadurek@dadurek:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/dadurek/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/dadurek/.ssh/id_rsa.
Your public key has been saved in /home/dadurek/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:PmvZBTT2dVWLSzVzhuI16deXqRhAEjGtHRer7lFNtEo dadurek@dadurek
The key's randomart image is:
+---[RSA 2048]---+
|      ==. .o =B|
|      o+*. B.*|
|      oo+E X +o|
|      .+.O ooo|
|      S. +o+. .|
|      .. ....|
|      o= .|
|      +oo|
|      ...|
+----[SHA256]-----+
dadurek@dadurek:~$
```

Dzięki tej komendzie stworzyliśmy publiczny oraz prywatny klucz dostępny w katalogu `.ssh`.

```
dadurek@dadurek:~$ cd .ssh
dadurek@dadurek:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
dadurek@dadurek:~/.ssh$
```

Aby wyświetlić te klucze można użyć komend `cat id_rsa` i `cat id_rsa.pub`.

Aby móc się logować do SSH poprzez klucz publiczny należy mieć go u siebie w folderze `.ssh` oraz dezaktywować logowanie przez hasło.

Aby dezaktywować logowanie przez hasło należy użyć komendy: `sudo nano/etc/ssh/sshd_config`

Należy w tym pliku zmienić `PasswordAuthentication yes` na `PasswordAuthentication no`.

```
dadurek@dadurek: ~  
GNU nano 2.5.3 File: /etc/ssh/sshd_config  
# Package generated configuration file  
# See the sshd_config(5) manpage for details  
  
# What ports, IPs and protocols we listen for  
Port 22  
# Use these options to restrict which interfaces/protocols sshd will bind to  
#ListenAddress ::  
#ListenAddress 0.0.0.0  
Protocol 2  
# HostKeys for protocol version 2  
HostKey /etc/ssh/ssh_host_rsa_key  
HostKey /etc/ssh/ssh_host_dsa_key  
HostKey /etc/ssh/ssh_host_ecdsa_key  
HostKey /etc/ssh/ssh_host_ed25519_key  
#Privilege Separation is turned on for security  
UsePrivilegeSeparation yes  
  
# Lifetime and size of ephemeral version 1 server key  
KeyRegenerationInterval 3600  
ServerKeyBits 1024  
  
# Logging  
SyslogFacility AUTH  
LogLevel INFO  
  
# Authentication:  
LoginGraceTime 120  
PermitRootLogin without-password  
StrictModes yes  
  
RSAAuthentication yes  
PubkeyAuthentication yes  
#AuthorizedKeysFile %h/.ssh/authorized_keys  
PasswordAuthentication no  
# Don't read the user's ~/.rhosts and ~/.shosts files  
IgnoreRhosts yes  
# For this to work you will also need host keys in /etc/ssh_known_hosts  
RhostsRSAAuthentication no  
  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Następnie należy zrestartować ssh poprzez komendę: `service ssh reload`

Aby zalogować się do servera Ubuntu poprzez PuTTY, należy w aplikacji PuTTYGEN dodać klucz prywatny. W zakładce Conversions zaimportować klucz i zapisać go. Następnie w PuTTY w zakładce SSH, AUTH należy w polu "Private key file for a authentication" wyszukać wcześniej zapisany plik z PuTTYGEN.

PuTTY Configuration

Category:

- Session
 - Logging
- Terminal
 - Keyboard
 - Bell
 - Features
- Window
 - Appearance
 - Behaviour
 - Translation
 - Selection
 - Colours
- Connection
 - Data
 - Proxy
 - Telnet
 - Rlogin
 - SSH
 - Serial

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address) Port

dadurek@40.114.252.48 22

Connection type:

☐ Raw ☐ Telnet ☐ Rlogin ☒ SSH ☐ Serial

Load, save or delete a stored session

Saved Sessions

UbuntuSSHkey

Default Settings
UbuntuSSHkey

Load

Save

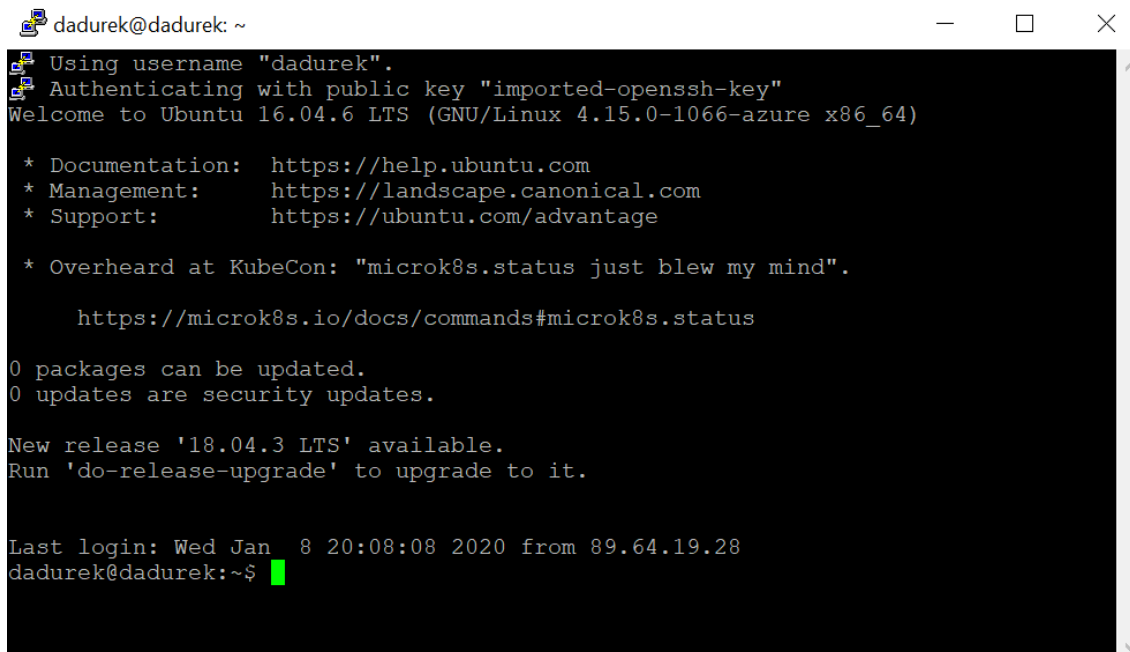
Delete

Close window on exit:

☐ Always ☐ Never ☒ Only on clean exit

About Help Open Cancel

Możemy cieszyć się logowanie poprzez klucz:

A terminal window titled 'dadurek@dadurek: ~' with standard window controls. The terminal output shows the login process for 'dadurek' using a public key. It displays the Ubuntu version (16.04.6 LTS), system information (GNU/Linux 4.15.0-1066-azure x86_64), and various links for documentation, management, and support. It also mentions a KubeCon anecdote and provides a link to microk8s status. The terminal shows that 0 packages can be updated and 0 security updates are available. A new release '18.04.3 LTS' is mentioned as available. The last login is recorded as 'Wed Jan 8 20:08:08 2020 from 89.64.19.28'. The prompt 'dadurek@dadurek:~\$' is shown with a green cursor.

```
dadurek@dadurek: ~
Using username "dadurek".
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-1066-azure x86_64)

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

 * Overheard at KubeCon: "microk8s.status just blew my mind".

    https://microk8s.io/docs/commands#microk8s.status

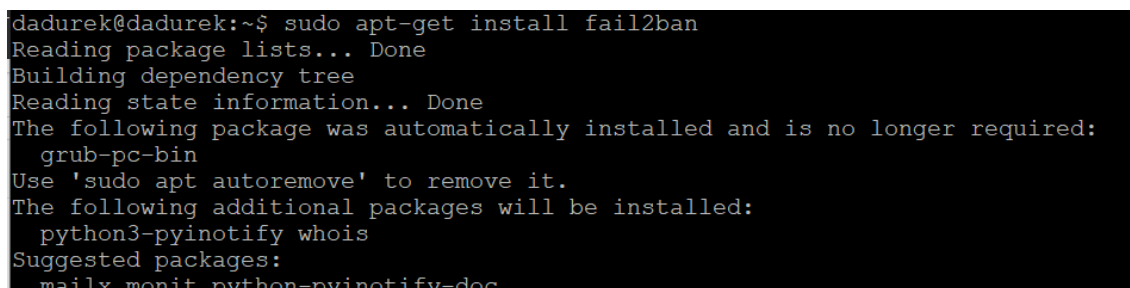
0 packages can be updated.
0 updates are security updates.

New release '18.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Jan 8 20:08:08 2020 from 89.64.19.28
dadurek@dadurek:~$
```

Fail2ban

Należy najpierw zainstalować Fail2Ban poprzez komendę `sudo apt-get install fail2ban`.

A terminal window showing the command 'sudo apt-get install fail2ban' being executed. The output shows the package lists being read, the dependency tree being built, and state information being read. It indicates that 'grub-pc-bin' was automatically installed and is no longer required, suggesting its removal with 'sudo apt autoremove'. It also lists additional packages to be installed: 'python3-pyinotify' and 'whois'. Suggested packages include 'mailx', 'monit', and 'python-pyinotify-doc'.

```
dadurek@dadurek:~$ sudo apt-get install fail2ban
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  python3-pyinotify whois
Suggested packages:
  mailx monit python-pyinotify-doc
```

Następnie należy użyć komendy: `cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local`.

Konfigurujemy linie:

- `ignoreip` - ip które nie mogą się łączyć
- `bantime` - czas po krótej bezczynności odstawiamy bana
- `maxretry` - mówi ile prób, po krótych dostajemy bana

```
dadurek@dadurek: /etc/fail2ban
GNU nano 2.5.3      File: jail.local      Modified

# ban a host which matches an address in this list. Several addresses can be
# defined using space separator.
ignoreip = 127.0.0.1/8

# External command that will take an tagged arguments to ignore, e.g. <ip>,
# and return true if the IP is to be ignored. False otherwise.
#
# ignorecommand = /path/to/command <ip>
ignorecommand =

# "bantime" is the number of seconds that a host is banned.
bantime = 3600

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime = 600

# "maxretry" is the number of failures before a host get banned.
maxretry = 5

# "backend" specifies the backend used to get files modification.
# Available options are "pyinotify", "gamin", "polling", "systemd" and "auto".
# This option can be overridden in each jail as well.
```

(część dalsza konfiguracji)

- należy przejść do części JAILS , w której zmieniamy standardowy port SSH na port 22222

```
#
# JAILS
#

#
# SSH servers
#

[sshd]
enable = true
port = 22222
logpath = %(sshd_log)s
```

Następnie restartujemy używając komendy: `sudo systemctl restart fail2ban` .

Po zmianie portu na 22222 należałoby teraz odblokować ten port w Firewall-u i zablokować port 22. Fail2Ban sam odblokowuje port, na który zmienimy jednakże sami

musimy zablokować port 22 używając odpowiedniej do tego komendy: `sudo iptables -A INPUT -p tcp --dport 22 -j DROP .`

```
dadurek@dadurek:/etc/fail2ban$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-N f2b-sshd
-A INPUT -p tcp -m multiport --dports 22222 -j f2b-sshd
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A f2b-sshd -j RETURN
dadurek@dadurek:/etc/fail2ban$
```

Zablokowanie wysyłania pustych pakietów z flagą NULL, pakietów SYN oraz pakietów z flagą XMAS.

Takie pakiety często używane są przez kaerów do badani sieci przez hakerów pod kątem badani portów. Do firewall-a należy dodać wykluczenie komendami:

- `sudo iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP .`
- `sudo iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP` (flaga SYN, blokuje aby uchronić przez wysyłaniem pakietów, które otwierają port i mogą przeciążyć serwer)
- `sudo iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP` (flaga XMAS)

```
dadurek@dadurek:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-N f2b-sshd
-A INPUT -p tcp -m multiport --dports 22222 -j f2b-sshd
-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j DROP
-A INPUT -p tcp -m tcp ! --tcp-flags FIN,SYN,RST,ACK SYN -m state --state NEW -j DROP
-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG FIN,SYN,RST,PSH,ACK,URG -j DROP
-A f2b-sshd -j RETURN
dadurek@dadurek:~$
```

Zablokowanie możliwości pingowania.

Pingowanie jest często wykorzystywane przez hakerów do odtworzenia topologii sieci, która powinna być tajemnicą każdej korporacji, każdego cyber-bezpiecznika. Robimy to komendą:

```
sudo iptables -I INPUT -m state --state ESTABLISHED,RELATED -j DROP
```