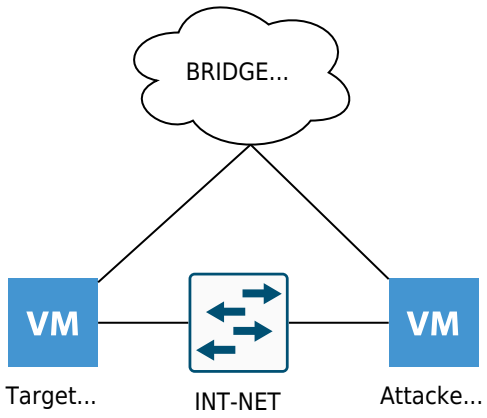
FTP Bruteforce (Patator i CICFlowMeter)

Schemat i opis środowiska testowego



Opis:

Środowisko składa się z dwóch maszyn wirtualnych (VM) działających na Ubuntu Server:

- Maszyna atakująca: uruchomiony program Patator do ataku brute-force na usługę FTP oraz tcpdump do przechwytywania ruchu.
- Maszyna ofiara: zainstalowany i skonfigurowany serwer FTP (vsftpd).
- Sieć: maszyny podłączone do tego samego switcha wirtualnego w Hyper-V.

Każda z Maszyn ma dysk o rozmiarze 8GB (Wymagania minimalne ubuntu mówią o minimalnie 5GB). Dyski są w formacie vhdx, wykorzystywanym przez hyper-v.

Dyski maszyn można pobrać z linków poniżej:

Oprogramowanie wykorzystane do wykonania eksperymentu

Komponent	**Wersja**	**Nazwa**
System operacyjny maszyn wirtualnych	24.04.2 LTS	Ubuntu Server
Hypervisor	Wersja: 10.0.26100.1882	Hyper-V
Serwer FTP	3.0.3	VSFTPD
Narzędzie ataku	0.7	Patator
Przechwytywanie ruchu	4.99.1	tcpdump
Analiza Ruchu	0.4.2	CICFlowMeter
Słownik do ataku	ok. 14 MB, 14344392 haseł	rockyou.txt

Instalacja i konfiguracja

Maszyna ofiara

```
administrator@target:~$ sudo apt update -y
[sudo] password for administrator:
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
58 packages can be upgraded. Run 'apt list --upgradable' to see them.
administrator@target:~$ sudo apt install vsftpd -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ssl-cert
The following NEW packages will be installed:
  ssl-cert vsftpd
0 upgraded, 2 newly installed, 0 to remove and 58 not upgraded.
Need to get 137 kB of archives.
After this operation, 380 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/main amd64 ssl-cert all
1.1.2ubuntu1 [17.8 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 vsftpd amd64
3.0.5-0ubuntu3.1 [120 kB]
Fetched 137 kB in 0s (424 kB/s)
Preconfiguring packages ...
Selecting previously unselected package ssl-cert.
(Reading database ... 86807 files and directories currently installed.)
Preparing to unpack .../ssl-cert_1.1.2ubuntu1_all.deb ...
Unpacking ssl-cert (1.1.2ubuntu1) ...
Selecting previously unselected package vsftpd.
Preparing to unpack .../vsftpd_3.0.5-0ubuntu3.1_amd64.deb ...
Unpacking vsftpd (3.0.5-0ubuntu3.1) ...
Setting up ssl-cert (1.1.2ubuntu1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ssl-cert.service
→ /usr/lib/systemd/system/ssl-cert.service.
Setting up vsftpd (3.0.5-0ubuntu3.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service →
/usr/lib/systemd/system/vsftpd.service.
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...
```

```
Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
administrator@target:~$
```

Plik konfiguracyjny `/etc/vsftpd.conf`:

```
administrator@target:~$ sudo nano /etc/vsftpd.conf
administrator@target:~$ cat /etc/vsftpd.conf
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
#
# Run standalone?  vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=YES
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=NO
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
#local_umask=022
#
```

```
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you
will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# If enabled, vsftpd will display directory listings with the time
# in  your  local  time  zone.  The default is to display GMT. The
# times returned by the MDTM FTP command are also affected by this
# option.
use_localtime=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
#
# If you want, you can arrange for uploaded anonymous files to be owned by
# a different user. Note! Using "root" for uploaded files is not
# recommended!
#chown_uploads=YES
#chown_username=whoever
#
# You may override where the log file goes if you like. The default is shown
# below.
#xferlog_file=/var/log/vsftpd.log
#
# If you want, you can have your log file in standard ftpd xferlog format.
# Note that the default log file location is /var/log/xferlog in this case.
#xferlog_std_format=YES
#
# You may change the default value for timing out an idle session.
#idle_session_timeout=600
#
# You may change the default value for timing out a data connection.
#data_connection_timeout=120
#
# It is recommended that you define on your system a unique user which the
# ftp server can use as a totally isolated and unprivileged user.
#nopriv_user=ftpsecure
#
# Enable this and the server will recognise asynchronous ABOR requests. Not
```

```
# recommended for security (the code is non-trivial). Not enabling it,
# however, may confuse older FTP clients.
#async_abor_enable=YES
#
# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do
ASCII
# mangling on files when in ASCII mode.
# Beware that on some FTP servers, ASCII support allows a denial of service
# attack (DoS) via the command "SIZE /big/file" in ASCII mode. vsftpd
# predicted this attack and has always been safe, reporting the size of the
# raw file.
# ASCII mangling is a horrible feature of the protocol.
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
#ftpd_banner=Welcome to blah FTP service.
#
# You may specify a file of disallowed anonymous e-mail addresses.
Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd.banned_emails
#
# You may restrict local users to their home directories.  See the FAQ for
# the possible risks in this before using chroot_local_user or
# chroot_list_enable below.
#chroot_local_user=YES
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
# (Warning! chroot'ing can be very dangerous. If using chroot, make sure
that
# the user does not have write access to the top level directory within the
# chroot)
#chroot_local_user=YES
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd.chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror"
assume
# the presence of the "-R" option, so there is a strong case for enabling
it.
#ls_recurse_enable=YES
#
```

```
# Customization
#
# Some of vsftpd's settings don't fit the filesystem layout by
# default.
#
# This option should be the name of a directory which is empty.  Also, the
# directory should not be writable by the ftp user. This directory is used
# as a secure chroot() jail at times vsftpd does not require filesystem
# access.
secure_chroot_dir=/var/run/vsftpd/empty
#
# This string is the name of the PAM service vsftpd will use.
pam_service_name=vsftpd
#
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=NO

#
# Uncomment this to indicate that vsftpd use a utf8 filesystem.
#utf8_filesystem=YES
administrator@target:~$
```

Tworzenie użytkownika FTP z hasłem a7s8d6a8s7d6a8s7d68s7:

```
administrator@target:~$ sudo adduser ftpuser
info: Adding user `ftpuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `ftpuser' (1001) ...
info: Adding new user `ftpuser' (1001) with group `ftpuser (1001)' ...
info: Creating home directory `/home/ftpuser' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ftpuser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `ftpuser' to supplemental / extra groups `users' ...
info: Adding user `ftpuser' to group `users' ...
administrator@target:~$
```

Restart usługi:

```
administrator@target:~$ sudo systemctl restart vsftpd
administrator@target:~$ sudo systemctl status vsftpd.service
● vsftpd.service - vsftpd FTP server
     Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled;
preset: e>
     Active: active (running) since Thu 2025-06-19 15:00:47 UTC; 6s ago
    Process: 4065 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty
(code=exited>
   Main PID: 4068 (vsftpd)
      Tasks: 1 (limit: 4602)
     Memory: 704.0K (peak: 1.5M)
        CPU: 7ms
     CGroup: /system.slice/vsftpd.service
             └─4068 /usr/sbin/vsftpd /etc/vsftpd.conf

Jun 19 15:00:47 target systemd[1]: Starting vsftpd.service - vsftpd FTP
server.>
Jun 19 15:00:47 target systemd[1]: Started vsftpd.service - vsftpd FTP
server.
administrator@target:~$
```

Maszyna atakująca

Instalacja Narzędzi:

```
sudo apt update
sudo apt install git python3-pip tcpdump -y
```

Instalacja Patatora:

```
git clone https://github.com/lanjelot/patator.git
cd patator
pip install -r requirements.txt
```

Instalacja CICFlowMeter w wersji Python:

```
cd ~
git clone https://github.com/ahlashkari/CICFlowMeter.git
cd CICFlowMeter
pip install -r requirements.txt
```

Pobranie słownika do ataku:

```
sudo apt install seclists unzip -y
cp /usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt.tar.gz ~/
cd ~
tar -xzf rockyou.txt.tar.gz
```

Test Komunikacji

```
administrator@target:~$ ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=0.180 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=0.229 ms
64 bytes from 10.10.10.2: icmp_seq=3 ttl=64 time=0.269 ms
64 bytes from 10.10.10.2: icmp_seq=4 ttl=64 time=0.286 ms
64 bytes from 10.10.10.2: icmp_seq=5 ttl=64 time=0.266 ms
64 bytes from 10.10.10.2: icmp_seq=6 ttl=64 time=0.293 ms
64 bytes from 10.10.10.2: icmp_seq=7 ttl=64 time=0.284 ms
^C
--- 10.10.10.2 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6182ms
rtt min/avg/max/mdev = 0.180/0.258/0.293/0.037 ms
administrator@target:~$
```

Przeprowadzenie Ataku

Uruchomienie przechwytywania:

```
sudo tcpdump -i eth0 port 21 -w ftp_attack.pcap
```

Atak Patator:

```
./patator ftp_login host=192.168.0.105 user=ftpuser password=FILE0
0=rockyou.txt -x ignore:mesg='Login incorrect' --rate-limit=5
```

- `–rate-limit=5`: ogranicza tempo ataku do 5 prób/sekundę (by nie przytłoczyć serwera)
- `rockyou.txt`: używa ogólnodostępnego słownika

⚠ W tym przypadku atak się nie powiedzie, ponieważ utworzone hasło nie znajduje się w rockyou.txt – i o to właśnie chodzi: chodzi o symulację nieudanego ataku, który będzie wyraźnie widoczny w ruchu.

Efekty Ataku

Po stronie atakującej

- Konsola Patatora wykaże tysiące nieudanych prób.
- Hasło nie zostanie złamane.
- Cały ruch jest zapisany w ftp_attack.pcap.

Po stronie serwera FTP

Fragment logu z /var/log/vsftpd.log:

```
WKLEIĆ TUTAJ FRAGMENT LOGÓW
```

Użycie CICFlowMeter (Python) i ekstrakcja cech

```
cd ~/CICFlowMeter
python3 CICFlowMeter.py -f ftp_attack.pcap -o output/
```

W katalogu `output/` został zapisany plik `ftp_attack_Flow.csv`.