DNS as a transport

Abstract

Develop tool to craft DNS packet and query DNS server so the message could be passed along.
These packets, carefully constructed, allow the encoding of messages. By querying a DNS server with these packets, the traditional domain resolution system is repurposed, serving as a covert medium for message transmission. However, ethical and legal considerations arise, as this unconventional use may pose security and privacy concerns.

HOW TO RUN?

1- Network

You must be in the same network (VPN vilnius school)

2- OpenNebula

The VM must be running in OpenNebula: Victim, DNS, Attacker.

☐	64961	ibmo0242	users	DNS srv	RUNNING	lxibm207	10.0.1.155
☐	64960	ibmo0242	users	Attacker	RUNNING	lxibm208	10.0.1.153
☐	64959	ibmo0242	users	Victim	RUNNING	lxibm210	10.0.1.150

3- Public Key

Make sure that your public key is allowed.

4- Connection to VMs

****Victim****
ssh -p 12468 ibmo0242@193.219.91.103
IP: 10.0.1.150

****Attacker****
ssh -p 10125 ibmo0242@193.219.91.103
IP: 10.0.1.153

****DNS server****
ssh -p 2773 ibmo0242@193.219.91.103
IP: 10.0.1.155

5- Preparing the DNS:

Restart DNS Server (to clear cache)
Command: *"sudo service bind9 restart"*

6- Execute the Attacker:

Locate the exfiltration folder:
Command: "cd exfiltration/"

Execute python script that will listen to incoming DNS requests
Command: "sudo python3 receive.py"

7- Work with the Victim:

Locate the secrets folder: "cd secrets/"

Execute python script that will exfiltrate the contents of file passwords.txt in the form of DNS queries.

Command: "sudo python3 extractor.py"

8- Final step

Look for file "ourData" in Attacker
Command: "cat ourData"

HOW TO DEVELOP THE PROJECT

Used VMs: (debian12)

-Victim
-Attack
-DNS

Victim

Update to latest debian version:
sudo apt-get update
sudo apt-get upgrade
Create a working directory
mkdir secrets
cd secrets/
Creating the data to be exfiltrated
nano passwords.txt
populate it with passwords
Install python libraries + the script
sudo apt-get install python3-dnspython
nano extractor.py
-paste script
modify the resolver to point to our server
sudo nano /etc/resolv.conf
nameserver 10.0.1.155

DNS

sudo apt-get update
sudo apt-get upgrade
Install DNS service “bind9”
“sudo apt install bind9”
Modify file **/etc/bind/named.conf.options** add:
sudo nano /etc/bind/named.conf.options

```
forwarders {
    10.0.0.1;
};
```

To forward all the data to a “real” DNS server.
Add to **/etc/bind/named.conf.options**

```
zone "evildomain.com" {
   type forward;
   forwarders {
      10.0.1.153;
   };
};
```

To simulate that we bought a domain whose DNS has the IP 10.0.0.153

//Everytime we want to run the project, we have to execute **sudo service bind9 restart** to clear the DNS cache.

Attacker

sudo apt-get update
sudo apt-get upgrade

Execute python script to start listening to port 53, and append to a file “ourData” the exfiltrated data.

Allow incoming packages to port 53(DNS)
sudo apt-get update
sudo apt-get install ufw
sudo ufw allow 53
sudo ufw allow 22
sudo ufw enable
restart machine

mkdir exfiltration
cd exfiltration/
nano receive.py
paste the script
sudo apt-get install python3-dnslib
sudo python3 receive.py
we’re now receiving exfiltrated data >:D

VULNERABILITIES AND CONCLUSIONS

DNS tunneling is slow exfiltrating data, it’s not commonly used either and that makes it dangerous since it’s not expected.

The way to detect this kind of attack is through Volumetric analysis since to exfiltrating files (by breaking them down into bytes, sending them over and later reassembling the file) a lot of queries are required