WIRESHARK

Res. Asst. Melek ŞENTÜRK

What is DNS Tunneling?

- The DNS (Domain Name System) protocol converts IP addresses of devices connected to the Internet.
- DNS tunneling is a technique used to transmit data over DNS queries and responses.
- Data is embedded in the subdomain portion of DNS queries.
- Data is broken into small pieces and encoded within DNS queries.
- The DNS server receives these queries, resolves the content, and sends it back as a response.

Areas of Use of DNS Tunneling

- Secret Data Transport: DNS tunneling can bypass firewalls and filtering systems to transmit secret data. Data is hidden within DNS queries and is generally undetected by firewalls.

- Device and System Communication: Can be used to transmit data between devices that do not have an internet connection.

- Attack and Security Testing: During penetration testing and security assessments, tunneling techniques are used for confidential data transmission.

Security Risks of DNS Tunneling

- Network Vulnerabilities: DNS tunneling may not be detected by firewalls, IDS/IPS systems. This technique can be used by malware or attackers for data theft.

- Data Leaks: Organizations can leak sensitive information using DNS tunneling.

- Impact on Network Performance: DNS tunneling can require high bandwidth and negatively impact network performance.

Methods to Detect and Prevent DNS Tunneling

- Monitor Tunneling Behavior: Anomalous DNS queries (for example, long subdomains or frequently failed queries) can be detected.
- DNS Query Restrictions: Network administrators can limit tunneling activities by monitoring the frequency and content of DNS queries.
- Firewall and IDS/IPS Systems: Filtering on DNS traffic can help detect and block suspicious queries.
- DNSSEC (DNS Security Extensions) Usage: DNSSEC is a security protocol used to ensure the accuracy of DNS responses and can help prevent tunneling attacks.

What is Iodine?

- Iodine is a tool used to tunnel data using DNS traffic on a network. This is used primarily by attackers or network administrators trying to bypass firewalls or filtering.

- DNS Tunneling: Iodine uses the DNS (Domain Name System) protocol for data transmission. Normally, DNS queries are used only for domain name resolution, but with tunneling, this protocol is transformed into a carrier that allows data to pass through firewalls and filtering systems.

- Data Transfer: Iodine can tunnel TCP/IP traffic or other protocols using the DNS protocol. This is typically used when data transmission on the network is blocked or firewalls limit traffic.

- Covert Channel: Iodine creates a covert channel within the network's normal DNS traffic. Encrypted data or commands can be sent over this channel. This can be used as a tool for data exfiltration and other malicious activities.

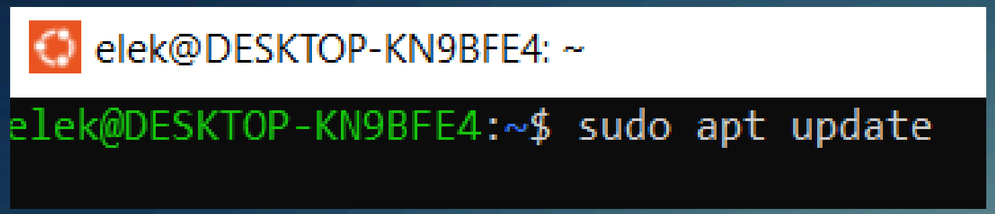
What is Iodine?

- Bypassing Firewalls and Filters: If there is a strict firewall or proxy on the network and only DNS traffic is allowed, Iodine uses this traffic to hide other internet traffic and exfiltrate data outside the network.

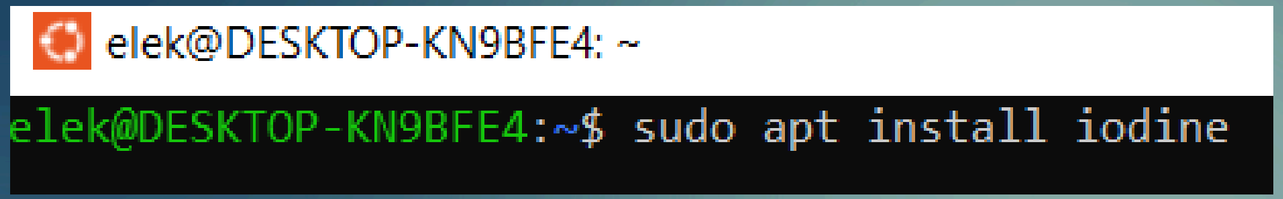
- Server and Client Structure: Iodine works between a server and a client. The server is the machine that initiates the tunnel and answers incoming DNS queries. The client receives data from the server by sending DNS queries.

Iodine Installation

- To update the package lists;



- To install the iodine tool;



Iodine Installation

- To open the etc/hosts file;

```
elek@DESKTOP-KN9BFE4:~$ sudo nano /etc/hosts
[sudo] password for elek:
```

- To use it instead of a real domain, the following line can be added to the /etc/hosts file;

```
  GNU nano 7.2                                                    /e
# This file was automatically generated by WSL. To stop aut
# [network]
# generateHosts = false
127.0.0.1       localhost
127.0.1.1       DESKTOP-KN9BFE4.        DESKTOP-KN9BFE4
127.0.0.1       tunnel.example.com

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

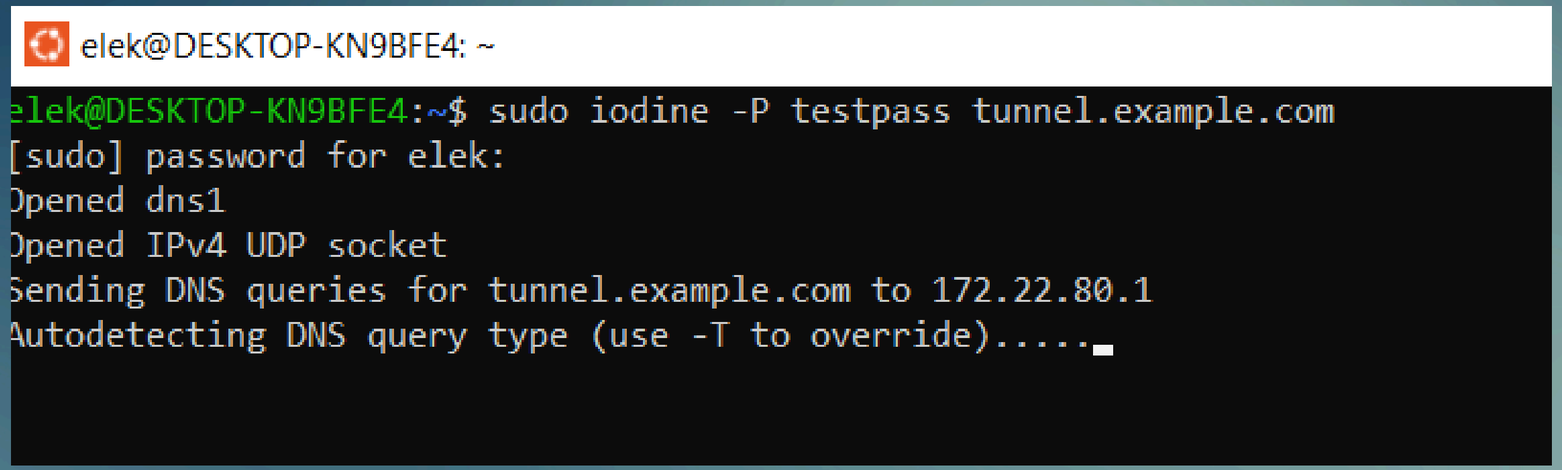
- The /etc/hosts file directs specific domain names to IP addresses that we specify.

iodined Server and Client Startup

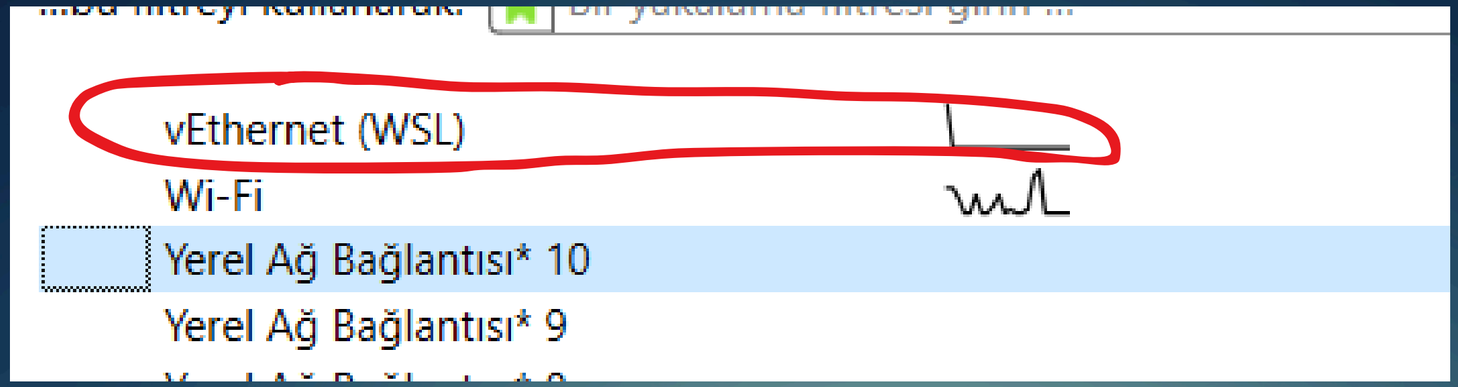
- To start iodined server;

```
elek@DESKTOP-KN9BFE4:~$ sudo iodined -f -c -P testpass 10.0.0.1 tunnel.example.com
[sudo] password for elek:
Opened dns0
Setting IP of dns0 to 10.0.0.1
Setting MTU of dns0 to 1130
Opened IPv4 UDP socket
Listening to dns for domain tunnel.example.com
```

- To start the iodined client in a different terminal;



Wireshark Side



Microsoft_b4:d6:…	Microsoft_d4:84:…	ARP	42	Who has 172.22.89.201? Tell 172.22.80.1
Microsoft_d4:84:…	Microsoft_b4:d6:…	ARP	42	172.22.89.201 is at 00:15:5d:d4:84:b5
172.22.89.201	172.22.80.1	DNS	85	Standard query 0xa0dc NULL yrbtvv.tunnel.example.com
172.22.89.201	172.22.80.1	DNS	85	Standard query 0xbf0b Unknown (65399) yrbtvw.tunnel.example.com
172.22.89.201	172.22.80.1	DNS	85	Standard query 0xdd3a TXT ytbtvx.tunnel.example.com
172.22.89.201	172.22.80.1	DNS	85	Standard query 0xfb69 SRV ytbtvy.tunnel.example.com
172.22.89.201	172.22.80.1	DNS	85	Standard query 0x1998 MX ytbtvz.tunnel.example.com