Assignment - DAT505

Building Scapy-based tools to perform ARP spoofing to become a Man-in-the-Middle and implementing selective DNS spoofing to redirect victims to attacker-controlled pages.

Author: András Tarlós

Lecturer: Ferhat Özgur Catak

November 11, 2025

Universitetet i Stavanger

Ethical Hacking (DAT505)

Norway

University
of Stavanger

Table of Contents

1 Setup summary

We will be using Oracle Virtual Box to manage our three virtual machines (VMs) needed for our experimentation. The following sections will go into detail, how the VMs have been set up, so that the experiments can be reproduced.

1.1 Attacker VM

The attacker VM has `Wireshark`, `Python3`, and `VsCode` installed.

Operating System	Kali Linux
IP Address	10.0.2.5
Hypervisor	Virtual Box
virtual CPUs	8
Base Memory	4096 Mb
Video Memory	64 Mb
Network Type	NAT Network

1.2 Victim VM

The victim only needs a web browser.

Operating System	Debian 13
IP Address	10.0.2.5
Hypervisor	Virtual Box
virtual CPUs	1
Base Memory	1028 Mb
Video Memory	16 Mb
Network Type	NAT Network

1.3 Gateway/Server VM

The server has `dnsmasq` installed and configured with DNS running on UDP port 53; the `dnsmasq.conf` can be found under the `evidence` folder. A simple Python3 webserver has been used for certain experiments during the completion of the three mandatory tasks.

Operating System	Debian 13
IP Address	10.0.2.7
Hypervisor	Virtual Box
virtual CPUs	1
Base Memory	1028 Mb
Video Memory	16 Mb
Network Type	NAT Network

2 Methodology

All of the VMs have been added to the same NAT Network, so that they can communicate with and ping each other. They can access the Internet as well, allowing to easily download tools if necessary.

The programming language Python has been used to solve the tasks. The module Scapy has been used to manipulate packets on the network.

Network traffic analysis was performed using command-line tools such as `tcpdump`, as well as graphical applications such as Wireshark, to capture and examine packets during attack demonstrations. This enabled detailed observation of how DNS spoofing attacks manipulate the domain name resolution process, and of how ARP cache poisoning redirects network traffic through the attacker's system.

3 Results

3.1 ARP Spoofer (Task 1)

Already having executed `arp_spoofer.py`, we can execute `traffic_interceptor.py`. This script captures a PCAP file with all the packets and two CSV files. The CSV files contain information about DNS domains and web URLs. To generate traffic between the target and the gateway, simple HTTP GET requests and DNS lookup requests have been generated.

```
> sudo python3 arp_spoof.py -t 10.0.2.15 -g 10.0.2.7 -i eth0


[*] Enabling IP Forwarding...


2025-11-11 09:24:22,374 - INFO - Target MAC address: 08:00:27:ff:a4:c5
2025-11-11 09:24:22,374 - INFO - Gateway MAC address: 08:00:27:57:41:35
2025-11-11 09:24:22,375 - INFO - Own MAC address: 08:00:27:d1:f8:5d
2025-11-11 09:24:22,634 - INFO - Sent spoofed ARP responses to
10.0.2.15 and 10.0.2.7
```

3.2 Interceptor (Task 2)

The `arp_spoofer.py` has to be executed on the Attacker VM. Multiple arguments have to be provided in order for the script to work. This includes the target (-t), the gateway/server (-g), and the interface (-i). The Python script ensures that both the target and the gateway exchange packets through the attacker. This allows us to sniff the packet exchange between them. This script was the cornerstone for the next tasks and was heavily used later on.

```
> sudo python3 traffic_interceptor.py
Sniffing 300 packets on eth0...
Saved 3 URLs to urls_093635.csv
Saved 6 DNS queries to dns_093635.csv

Summary: 18 HTTP requests, 125 DNS queries
Top 5 DNS: [('help_hacker.com', 36), ('example.com', 30),
('im_hacker.com', 24), ('hacker.hu', 18), ...]
```

3.3 DNS Spoofer (Task 3)

The `dns_spoofer.py` also depended on ARP spoofing. Without it, we could have only spoofed our own DNS requests, which is utmost unproductive. This Python script has to be run before the target tries to reach resolve a domain name. In the example down below, a web server is running on **10.0.2.5**; the script modifies DNS replies and in our case it points **facebook.com** to our malicious IP address **10.0.2.5** (the Attacker VM)

```
> sudo python3 dns_spoofer.py
DNS Spoofer Started - Listening for DNS queries
make sure ARP spoofing is active
Please Ctrl+C to stop
SPOOFING: facebook.com. -> 10.0.2.5
Sent spoofed response to 10.0.2.15
...
```

4 Mitigation Ideas

4.1 Mitigating ARP Poisoning Attacks

Since the ARP protocol has not been designed with security in mind, we need to find ways to mitigate the attack surface of ARP. Wanting to harden our security posture, we can try different approaches to prevent ARP Poisoning attacks:

- **Static ARP Tables**: It is possible to statically map all the MAC addresses in a network to their IP addresses. This is a highly effective technique; however, it comes with tremendous administrative burden.
- **Switch Security**: Switches have a feature called Dynamic ARP Inspection (DAI), which can evaluate the validity of each ARP message, and drop packets that appear suspicious or malicious, if enabled. Enabling Port Security on a switch can also help mitigate ARP cache poisoning attacks by only allowing a single MAC address on a switch port.
- **Network Isolation**: Creating multiple subnets – segmenting the network – may make it harder for a threat actor to achieve ARP cache poisoning. An attack on one subnet cannot impact devices in another.
- **Encryption**: Using encryption can mitigate the potential damage caused by ARP Poisoning. Even when a threat actor can achieve MITM, using encryption like SSL/TLS can render this type of attack useless. The threat actor can still intercept network traffic, but cannot do anything with it in its encrypted form.

Source: Varonis Blog

4.2 Mitigating DNS Spoofing Attacks

The keep a company safe from DNS Spoofing Attacks, one can implement the following security measures:

- **Set up DNSSEC**: Through cryptography, digital signatures, and additional methods, the Domain Name System Security Extensions (DNSSEC) system validates responses to domain name quires, ensuring that duplicate redirections do not occur at any point during the process.
- **Regularly apply patches to DNS servers**: Ensuring that the DNS server you are using has been patched to the latest version can avoid breaches.
- **Perform thorough DNS traffic-filtering**: This method has proven to be the best method to identify and combat DNS-delivered attacks.

Source: Heimdal Security Blog

5 Ethics

All of the above-mentioned experimentation with hacking tools have been conducted in a safe and isolated virtual environment. None of the showcased techniques presented have been and should be used against university, corporate, or public networks without the written permission of the network owner. Unauthorized attacks are considered **illegal** and can lead to disciplinary action.