Assignment: Practical Exploration of DNSSEC and Attack Simulation

Individual Assignment **Total Marks : 100**

Due Date : Apr 24, 2025 11:58 PM GMT+5:30 **Late Submission Penalty: 5% Per Day**

Note: It is **Highly** recommended to use **linux (Ubuntu 20/22) machine (NO VM)** for this assignment.

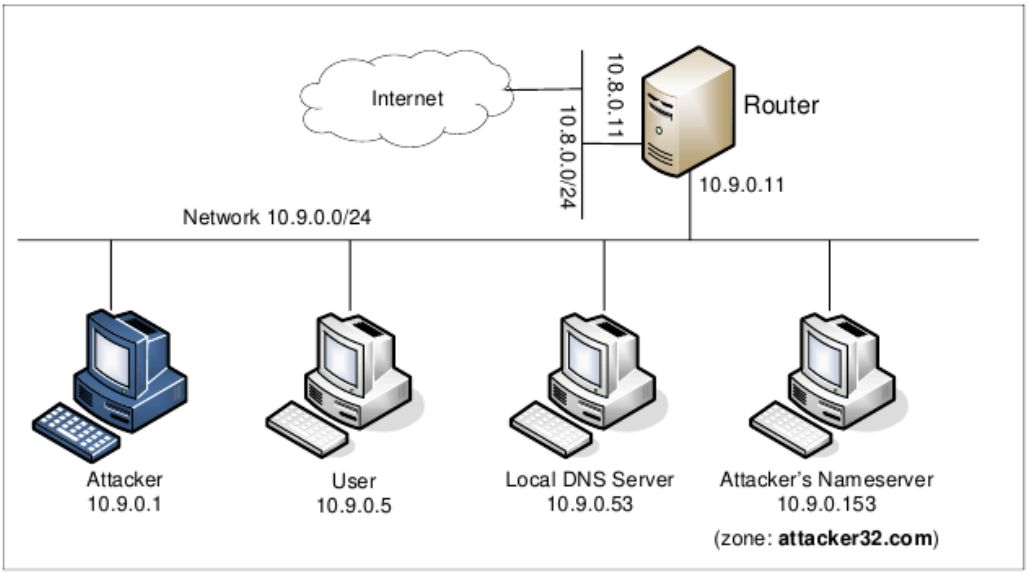
Task 0: Install Docker (You can skip this step if docker is already present)

1. Download this script: install_docker.sh
2. Give Execute permissions: $ *sudo chmod +x install_docker.sh*
3. Execute the script (It will take some time): $ *./install_docker.sh*
4. Restart your computer to apply changes and it will enable Docker to run without requiring `sudo` for each command.
5. Alternatively you can also use the below reference for installing docker.

References:

1. https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04

Task 1: DNS amplification attack simulation and mitigation (30 Marks)



In this assignment, you will demonstrate a DNS amplification attack using a spoofed IP address and then apply mitigation using rate limiting on the DNS server.
We will use Attacker's Nameserver as our DNS server

You have to perform the attack from the terminal of the Attacker container.

Command for listing all docker containers
docker ps -a

Command for getting into the terminal of a container:
docker exec -it <container_name_or_id> bash

Command for pulling the files from the docker containers into the host machine
docker cp <container_id_or_name>:<path to the file in the container> <path on the host machine>

Step 0: Setup Lab Environment

a. Download the Labsetup.zip file from this AMD_64
    i. MacOS/ ARM users download this zip -> MacOS
b. Unzip the contents
c. Navigate into the Labsetup folder.
d. Build the setup: *docker compose build*
e. Start the setup: *docker compose up*

Step 1: Analyze "**ANY**" query in DNS

a. Use the dig tool to perform an "ANY" type query on the local DNS server:
    i. Run: $ **dig @10.9.0.53 example.com ANY**
b. Observe and identify the different types of DNS records returned by the query.

Step 2: Now perform the Attack (From the Attacker's container)

a. **Send Spoofed DNS Query**:
    i. Create a Python script using Scapy to spoof the source IP (source IP set to the user's IP address [10.9.0.5]) and send a DNS "ANY" query to the local DNS server.
    ii. This causes the response to be sent to the user's machine, not the attacker.
b. **Capture DNS Request and Responses**:
    i. On the attacker machine, use tcpdump to capture the DNS request packet.
    ii. On the user's machine, use tcpdump to capture the DNS response packet.
    iii. Calculate the amplification factor:
        1. *Size of the response packet/ Size of the request packet*.
c. **Send a Burst of packets**:
    i. Modify the script to send DNS queries continuously for 5 seconds using the spoofed source IP.
    ii. On the user machine, capture the incoming burst of DNS response packets using tcpdump.

Step 3: Mitigating the attack

a. Edit the *named.conf.options* file of the local DNS server to enable rate limiting.
   i. Choose a suitable threshold.
   ii. Look at the provided reference for more information on response rate limiting.
b. Rebuild the docker setup: **redo points c, d & e from Step 0**.
c. Now use the script created in step 2 to send bursts of DNS queries to the local DNS server.
d. Now capture the burst of the DNS response packets on the user's machine using tcpdump.
e. Now compare the number of responses in Step 2 with the number of responses in step 3.

References

1. https://serverfault.com/questions/490245/bind-dns-rate-limit-and-values-for-responses-per-second-and-window

Note: After successfully completing Task 1, execute below command to clean docker images

- `sudo docker system prune -af`

Task 2: Setting Up DNSSEC Infrastructure (10 Marks)

Note : Please run the commands from outside the docker container not inside the container.

Step 1: Clone SEED Lab repository.

Step 2: Execute the following commands to setup the docker environment

a. `cd seed-labs/`
b. `cd lab-setup/docker-images/seed-dns/image_local_dns_server`
   ARM users use the below command :
   `cd lab-setup/docker-images/seed-dns-arm/image_local_dns_server`
c. `docker build -t seed-base-image-bind .`
d. `cd ../../../../category-network/DNSSEC/Labsetup/`
   ARM users use the below command :
   `cd ../../../../category-network/DNSSEC/Labsetup-arm/`
e. `cd local_dns_server`
f. `docker compose up -d --build`
g. `sudo apt-get install bind9utils`

Step 3: Follow sections 3 to 6 in the documentation to configure DNSSEC.

Task 3: DNSSEC Signature Replay Attack simulation (30 Marks)

Step 1: Perform the Signature Replay Attack

a. Capture DNSSEC Traffic: Use a packet capture tool like Wireshark or tcpdump to capture DNS responses from your DNS server.

b. Analyze the Capture: Examine the capture in Wireshark to find a DNSSEC response that contains the DNSSEC signature (RRSIG) and associated DNS records

c. Replay the Captured Signature

Step 2: Observe the Attack

a. Monitor the Resolver's Behavior:
   i. If the DNS resolver doesn't validate the timestamp or freshness of the DNSSEC signature (via the Expire field or TimeToLive in the DNSSEC records), it may accept the replayed signature, which could lead to a successful attack.
   ii. Observe this with Wireshark or by checking the logs on the resolver.

b. Explain the Risks:
   i. Highlight how the DNSSEC signature replay attack could allow the attacker to inject false DNS records into the resolver's cache.
   ii. Discuss the importance of expiration times and nonce-based signatures in DNSSEC

Step 3: Mitigation: Prevent Signature Replay

a. Implement Time-Based Validity
   i. Ensure the RRSIG records have a valid Expire timestamp to prevent signatures from being used indefinitely.
   ii. This should be configured when signing the DNS zone

b. Configure DNS Resolvers for Replay Detection:
   i. Ensure that DNS resolvers are configured to validate the freshness of signatures using expiration times or nonces. Many resolvers automatically discard signatures older than the Expire time specified in the RRSIG record.

NOTE: To Check Freshness: The script can use libraries such as dnspython to interact with DNS records and check the expiration time of DNSSEC signatures

References

1. Blog:
   a. https://blog.bluella.in/protect-against-dnssec-replay-attacks
   b. https://www.quillaudits.com/blog/web3-security/replay-attack
2. Research paper:

a. https://cs.gmu.edu/~eoster/doc/npsec_08-revocation.pdf
b. https://repository.mdx.ac.uk/download/159e25b0fe8217615f92a872ebf075289d00453180ddfc66f170782d3eea1a51/357512/paper.pdf

Task 4: DNSSec Keytrap attack simulation (30 Marks)

Step 1: Setup spare-edu name server

a. Generate multiple DNS keys with large sizes using the command in section 3.1 of documentation for a edu.smith2022 name server in the respective folder present in the nameserver folder.
b. Follow Sections 3.2 to 5 to configure the spare-edu server, the EDU TLD server, and the Root server for proper DNS resolution.

Step 2: Execute the following commands separately and log CPU utilisation of user container and local-dns-server container using below given

.

script:

`dig @10.9.0.53 www.example.edu +dnssec`
`dig @10.9.0.53 www.smith2022.edu +dnssec`

```
#!/bin/bash
LOG_FILE="docker_cpu_log.csv"
echo "Timestamp,Container,CPU%" > "$LOG_FILE"
while true; do
    TIMESTAMP=$(date +"%Y-%m-%d %H:%M:%S")
    docker stats --no-stream --format "{{.Name}},{{.CPUPerc}}" |
while read line; do
        echo "$TIMESTAMP,$line" >> "$LOG_FILE"
    done
    1
done
```

Step 3: Perform the same experiment by varying the number of keys in smith and plot the CPU Utilization of smith local dns server to get a single DNS record vs No of keys.

References

1. Keytrap attack

Note: Capture the pcap files using tcpdump wherever required and submit them along with the report.

TA Support:

Task 0, 1: Ashutosh Rajput

Task 2,4: Patel Heetkumar Dilipbhai

Task 3: Sakshi Srivastava

Submission in Classroom as Tarball/Zip

- Make a detailed report (including screenshots wherever necessary) for all the tasks.
- A single zip file named <roll_number>_DNSSEC.zip, containing the report and pcap/log/script files (named according to the task).

Evaluation Scheme

- 60% - Submission
- 40% - Viva

ANTI-PLAGIARISM STATEMENT <Include it in your report>

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarised the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honour violations by other students if I become aware of it.

Name:

Date:

Signature: <your initial>