Lab 3 - Network Intrusion Detection

Objective

The objective of this lab is to introduce you to network intrusion detection systems (NIDS). These systems are invaluable for monitoring how attackers attack servers and infrastructures, and implement mitigation schemes for such attacks. You will be setting up `Snort` in this lab, which is one of the most widely used network intrusion detection systems.

Background

`Snort` is an open-source signature based network intrusion detection system. It has a large database of built-in rules that can be used to detect many known attacks, such as command injection, privilege escalation, SQL injection, cross-site scripting, etc. The detection is done in real-time, enabling administrators and cybersecurity professionals to pro-actively mitigate attacks on their servers and infrastructures. Additionally, the system can also be used to setup custom rules using a simple and intuitive syntax. In 2009, `Snort` was named by InfoWorld's Open Source Hall of Fame as "one of the greatest open-source software of all time". You can read more about `Snort` directly on their website.

Setup

Login to the Linux VM that you used in the previous labs.

1. You will be setting up `Snort` on the Docker container `comp3611/login` that you also used for Task 2. Start this container by typing the following command in a terminal on your Linux VM.

   ```
   docker run --detach --name lab3 comp3611/login
   ```

2. Enter a `root` shell on the Docker container using the following command. You will use this shell to install `Snort`, configure it, and setup the detection rules.

   ```
   docker exec -it lab3 /bin/bash
   ```

3. Update the container's package repository, perform upgrades and install `Snort` using the following command. When prompted for the address range of local network, enter `172.17.0.0/16`. *Note that the command also installs vim to enable you to update files. You can install another editor (e.g., nano) if you prefer.*

   ```
   apt update && apt upgrade && apt install snort vim
   ```

4. The `Snort` configuration file will be installed at `/etc/snort/snort.conf`. This file contains all the configuration settings that `Snort` requires to function. Read this file once to understand how the different settings can be used by an administrator. To keep this lab simple, you need to update just the `ipvar HOME_NET` setting to `172.17.0.0/16`.

Part 1 - Creating Custom `Snort` Rules

In this part, you will create a custom rule to instruct `Snort` to detect all ICMP packets sent from / received by your container.

1. Read about `Snort`'s signature syntax in the User's Manual (Chapter 3). Ensure that you understand how to create custom rules before proceeding to the next step. In particular, review the meta-data options called `msg` and `sid`.

2. All custom rules in `Snort` can be specified in the `/etc/snort/rules/local.rules` file. On a new installation, this file should be empty. Create a new rule in this file that generates an alert whenever any ICMP packet is received by / sent from this container. The `msg` meta-data must be "Snort ICMP Test" and the `sid` must be "10000001".

3. Once you are certain the above rule is correct, start `Snort` by using the following command. The `-A console` flag instructs `Snort` to print alerts directly to the console. The `-k none` flag instructs `Snort` to analyze all packets regardless of errors. *Ensure that `Snort` starts up correctly and there are no errors.* If there are errors, re-check the configuration file and the local rules file, fix any problems and re-run the command.

   ```
   snort -c /etc/snort/snort.conf -A console -k none
   ```

4. In another terminal window on your Linux VM, run the command `ping -c 3 172.17.0.2`. Observe the console window of `Snort`. You should see six alerts generated with the message "Snort ICMP Test" and sid "10000001". If you don't, re-perform Steps 2 and 3 and try this step again. Take a screenshot of the alerts for your lab report.

Exit `Snort` (Ctrl + C) once this part has been completed before proceeding further.

Part 2 - Using Existing `Snort` Rules

In this part, you will test `Snort` using rules that were automatically downloaded during the `Snort` install. The focus will be on the `telnet` protocol, however, similar rules exist for many different network protocols.

1. Open the `/etc/snort/rules/telnet.rules` file. This file contains rules for detecting intrusions on the `telnet` protocol. Uncomment all the rules in this file. Pay special attention to rules pertaining to `4Dgifts` accesses. You can read more about the `4Dgifts` vulnerability at this link.
2. Open the `/etc/snort/rules/info.rules` file. Pay special attention to rules pertaining to bad `telnet` logins. Uncomment all those rules.
3. Start `Snort` again by using the following command. *Ensure that `Snort` starts up correctly and there are no errors.* If there are errors, re-check the `telnet.rules` and `info.rules` files, fix any problems and re-run the command.

   ```
   snort -c /etc/snort/snort.conf -A console -k none
   ```

4. In another terminal window on your Linux VM, attempt to `telnet` into the container using the command `telnet 172.17.0.2`. When prompted for a password, enter an incorrect password first and the correct password next. Once logged in, switch to user `4Dgifts` using the command `su 4Dgifts`. You will get an error that the user does not exist, however, that does not matter for our purpose.
5. Observe the console window of `Snort`. You should see alerts for both the bad login and the `4Dgifts` access. If you don't, re-perform Steps 1 to 4 and try this step again. Take a screenshot of the alerts for your lab report.

Exit `Snort` (Ctrl + C) and then exit out of the container.

Cleanup

Stop and remove the container by typing the following command in a terminal on your Linux VM.

```
docker stop lab3 && docker rm lab3
```

Lab Report

For this lab, each student must submit a report with the following information:

1. Submit the `Snort` configuration file `/etc/snort/snort.conf`.
2. Submit the `/etc/snort/rules/local.rules` file containing the ICMP detection rule.
3. Submit the screenshots showing the alerts generated during parts 1 and 2.

Grading

- 20 points - An error-free `Snort` configuration file
- 40 points - Successfully completed part 1 of the lab
- 40 points - Successfully completed part 2 of the lab

Optional Extra Credit - Host-based Intrusion Detection

Grading - 2.5% added to the final grade

Advanced Intrusion Detection Environment (`AIDE`) is an open-source host based intrusion detection system. The system can detect changes to the file-system that can be indicative of compromises on a host. Typically, a cybersecurity administrator will create a clean `AIDE` database (baseline) and use this database to check for any deviations or modifications to the host file system. This database should contain all the information regarding sensitive files, system binaries, external libraries, etc. Learn more about `AIDE` here.

1. `AIDE` can be installed on your Linux VM by typing the following command in a terminal.

   ```
   sudo apt install aide
   ```

2. To setup `AIDE`, edit the configuration file located at `/etc/aide/aide.conf`. Your configuration must address the following:

   - This config file should be owned by `root`, and readable and writable only by `root`.
   - Configure the file to detect changes in all directories and files on the file-system unless you have a strong reason not to. If you have excluded certain directories or files, explain the reason for exclusion in your lab report. Examples of exclusions can be directories that change regularly (e.g., `/tmp` and `/var/log`).
   - Setup a cronjob to run `AIDE` every 4 hours. Any alerts that are generated by `AIDE` should be emailed to you on your local Linux account (Hint: use `sendmail`).

3. Once you are satisfied with your configuration, initialize `AIDE` to create the baseline database. Note that initialization may take a long time to calculate the hashes of the files.

4. Next, change the modification time of an executable using `touch` (e.g., `touch /bin/ping`). Now when the cronjob executes, you should receive an email with an alert about the change. Take a screenshot of the email alert for your lab report.

5. Your lab report must contain the following:

   - The working `AIDE` configuration file
   - Screenshot of the cronjob setup to execute `AIDE` every 4 hours
   - Screenshot of the alert email (this should show the alert corresponding to the change you made using touch command).