CSCI 493-81
Lab 5
Security Lab
Topic: Intrusion Detection Systems

Prof. Dietrich

March 12, 2025

Group size: 4 — Setup needed: Windows/macOS, Unix hosts, SPHERE[1,2]

1 General description

In this lab we are getting familiar within intrusion detection systems (IDSes), both at the host and network levels. You are also setting up an environment similar to the one for future labs. This lab is based on work by Jelena Mirkovic and Jeff Mates.

2 File with samples and results

The files are in lab-ids.zip. The files contain sample configurations (basic) and examples for setting up simple IDSes. In some cases, you still have to develop some scripts, write your own code to improve the results, or develop signatures (direct or behavioral).

3 Setting up an IDS

In this part of the lab, we will be setting up signature for hosts and networks.

3.1 Requirement

- You use the instructions provided with this lab. You should use SPHERE as you have before, with the lab name of `snort`.
- Your output should demonstrate that you have done the experiments, such as screenshots, Unix typescripts, or logs.

[1]This document is Copyright ©2025 Sven Dietrich
[2]This document is for internal CUNY use only

4 Tasks

You are on the security team at FrobozzCo. Recent requests from management after the restructuring fiasco aim at upgrading the security of hosts and networks at FrobozzCo to deflect any attempts by outsiders, including those from the Zembor Corporation. You are to upgrade the security stance and explore host and network intrusion detection systems, using rule-based, signature, and possibly behavioral approaches. First you must do this in a test environment before deploying it in a production setting.

At FrobozzCo the computers are not the latest, so it is not always possible to ensure the security of legacy applications. At FrobozzCo they use systems that are known to be highly insecure and will continue to do so for the foreseeable future as it would not be cost effective to replace or modify these programs. As such the best we can do to improve the security of these systems is to place them behind filters that can be controlled. In this exercise you will be using Snort to help secure a highly insecure application without modifying the application itself. This application functions as a simple password protected file server for plain text documents, and it is up to you to use Snort to help protect this information.

This network is broken into three chunks. The first chunk contains the work environment which houses the legacy server along with a single client machine. This network is protected by the Snort router, which is set not to permit any traffic through at at the start of this exercise. The second network contains a single outsider machine. Finally, the third contains two client computers. All of these networks are connected to each other using a single router.

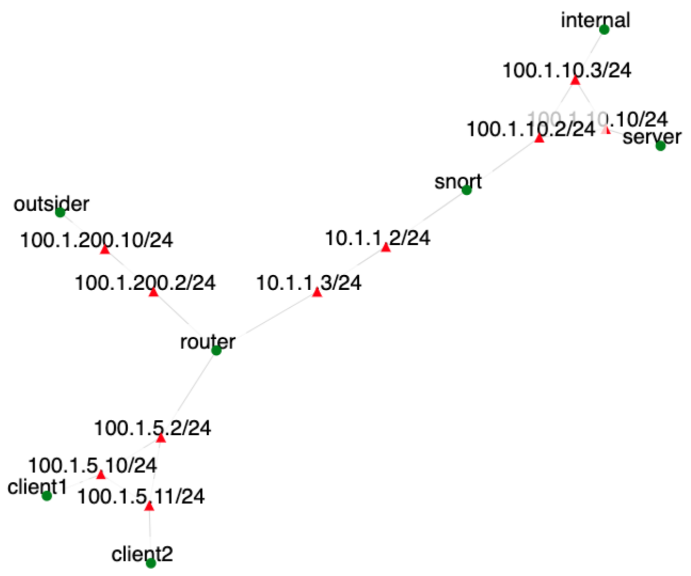

Figure 1: FrobozzCo network setup

4.1 Start Snort without rules

1. Connect to the snort node and modify `/etc/snort/snort.debian.conf` to set the `DEBIAN_SNORT_HOME_NET` to `100.1.10.0/24`. Then type:

   ```
   sudo snort --daq nfq -Q -v
   ```

2. On `client1` node type:

   ```
         ftp bob
         ...enter "anonymous" as username
         ...enter any string as password
         get ducky.xml
   ```

   You should see a large number of packets being reported by Snort. On `router` node run:

   ```
   tcpdump -i ethX -nn -s 0 -X
   ```

   replacing ethX with the interface with the IP address `10.1.1.3`.

Questions:

1. What happens to the traffic to `client1` when Snort is not running?
2. Is this a good thing?
3. Can you reconstruct contents of ducky.xml on router node?
4. Based on Snort's output what can you say about the application? What port does it connect to?
5. What does the "-Q" option do in Snort? Use man page to find out.
6. What does the "–daq nfq" option do in Snort? Use man page to find out.

4.2 Write Rules to Guard Against Simple Requests

1. In that terminal where `snort` is running stop it by using ctrl + c.
2. Use any text editor to edit `/etc/snort/rules/local.rules`. Add the following snort rule that prevents .xml files from being sent out,

   ```
   reject tcp 100.1.0.0/16 ANY -> 100.1.10.10 [Port from Question 3]
   (msg: "XML Read Attempt Detected"; sid:1; content:".xml";)
   ```

3. Using this rule as an example write a rule that prevents classified data from being sent to the outsider computer, but does not prevent it from being sent to any other computers. Classified data is stored in files with names that share a prefix ”classified”.

4. Start snort using your new rule with the command:

```
sudo snort  --daq nfq -Q -c /etc/snort/snort.conf  -A console
```

5. Log onto `client1, client2` and `outsider` to see if these rules worked. Try using `ftp` on these nodes to read classified files.

Questions:

1. What happens to FTP traffic from `client1` when you run snort? Show the FTP output as client1 tries to exfiltrate ”classified.txt” file.

2. What rule did you use to secure the ”classified” files?

3. Capture and compare the network traffic for the `server` when filtering these results using your configuration file and when no file is used. Can you recover the contents of classified files?

4. Can you think of any others files or extensions that should be filtered against? You can see what files the `server` node has in its `/srv/ftp` folder.

4.3 Attack - Defense

1. Make sure you have your filtering rule engaged on Snort

2. On `outsider` node install `nmap`:

```
apt update -y
apt install nmap -y
```

3. Run:

```
nmap 100.1.10.0/24
```

   to scan the server network, and check your Snort output.

4. On `outsider` node run:

```
ping -f server
```

   to generate a flood of ICMP traffic to `server` node. Create a new Snort rule that detects any ICMP packets sent from the outside into the network.

Questions:

1. Show Snort alerts for NMAP.
2. Show your new rule to detect ICMP traffic.
3. Show Snort alerts for ICMP flood (a few alerts are enough).
4. Try changing the action in the new rule to "reject" instead of "drop". What does this do to the traffic and why do you think this is? Is this a good or a bad thing?

5 Word Problems

You should submit a document with the following items (label each section):

1. An explanation of how Snort can be used to protect legacy systems.
2. Answers to all the questions in this file.

6 Deliverables

1. A report describing all your findings above.
2. A zip file containing:
   - Any signatures or scripts developed.
   - Answers to all word problems in Part 5.
3. Submit by the class on March 26, 2025.

7 Grading

Points will be subtracted if any of the pieces of the deliverables are missing or incomplete. The late submission policy applies.