CYBR 500 Project 4 Snort – Intrusion Detection (30 points)

This lab was developed for the Labtainer framework by the Naval Postgraduate School, Center for Cybersecurity and Cyber Operations under National Science Foundation Award No. 1438893. This work is in the public domain, and cannot be copyrighted.

# Overview

This exercise introduces the use of the snort system to provide intrusion detection within a Linux environ- ment. Students will configure simple snort rules and experiment with a network intrusion detection system, (IDS).

# Lab Environment

This lab runs in the Labtainer framework, available at [http://nps.edu/web/c3o/labtainers.](http://nps.edu/web/c3o/labtainers) That site includes links to a pre-built virtual machine that has Labtainers installed, however Labtainers can be run on any Linux host that supports Docker containers.

From your labtainer-student directory start the lab using:

labtainer snort

A link to this lab manual will be displayed.

# Network Configuration

This lab includes several networked computers as shown in Figure [1.](#_bookmark0) When the lab starts, you will get virtual terminals, one connected to each component. The gateway is configured with iptables to use NAT to translate sources addresses of traffic from internal IP addresses, e.g., 192.168.2.1, to our external address, i.e., 203.0.113.10. The iptables in the gateway also routes web traffic (ports 80 and 443) to the web server component by translating the externally visible destination address to the internal web server address.

The gateway is also configured to mirror traffic that enters the gateway via either the 203.0.113.10 link, or the link to the web server. This mirrored traffic is routed to the snort component. This mirroring allows the snort component to reconstruct TCP sessions between the web server and external addresses.

The snort component includes the Snort IDS utility. It also includes Wireshark to help you observe traffic being mirrored to the snort component.

The web server runs Apache and is configured to support SSL for web pages in the [www.example.com](http://www.example.com/) domain.

The remote ws component includes the Firefox browser, and a local /etc/hosts file that maps [www.example.com](http://www.example.com/) to the external address of the gateway, i.e., 203.0.113.10. The internal workstation (ws2) also includes Firefox and an entry in /etc/hosts for [www.example.com.](http://www.example.com/) Both workstations also include the nmap utility.

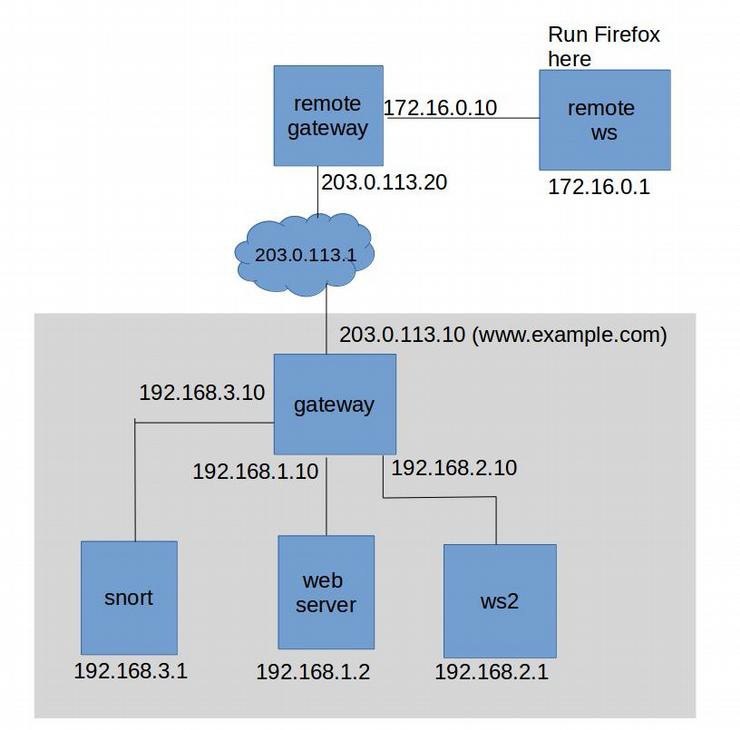


Figure 1: Network topology for the snort lab

# Lab Tasks

It is assumed that the student has received instruction or independent study on the basic operation of Snort, and the general goals and mechanics of network intrusion detection.

Review the network topology. In particular, consider the iptables settings on the gateway. These can be seen by reviewing the commands in /etc/rc.local, which are used to define the NAT translations and, critically for this lab, mirror traffic to the snort component.

## Starting and stopping snort

The Snort utility is installed on the snort component. The home directory includes a start snort.sh script that will start the utility in *Network Intrustion Dection Mode*, and display alerts to the console. For this lab, you are required to start snort with:

./start\_snort.sh

When it comes time to stop snort, e.g., to add rules, simply use CTL-C.

## Pre-configured Snort rules (5 points)

If you have used CTR-c to stop snort, start it again in Tom@Snort terminal:

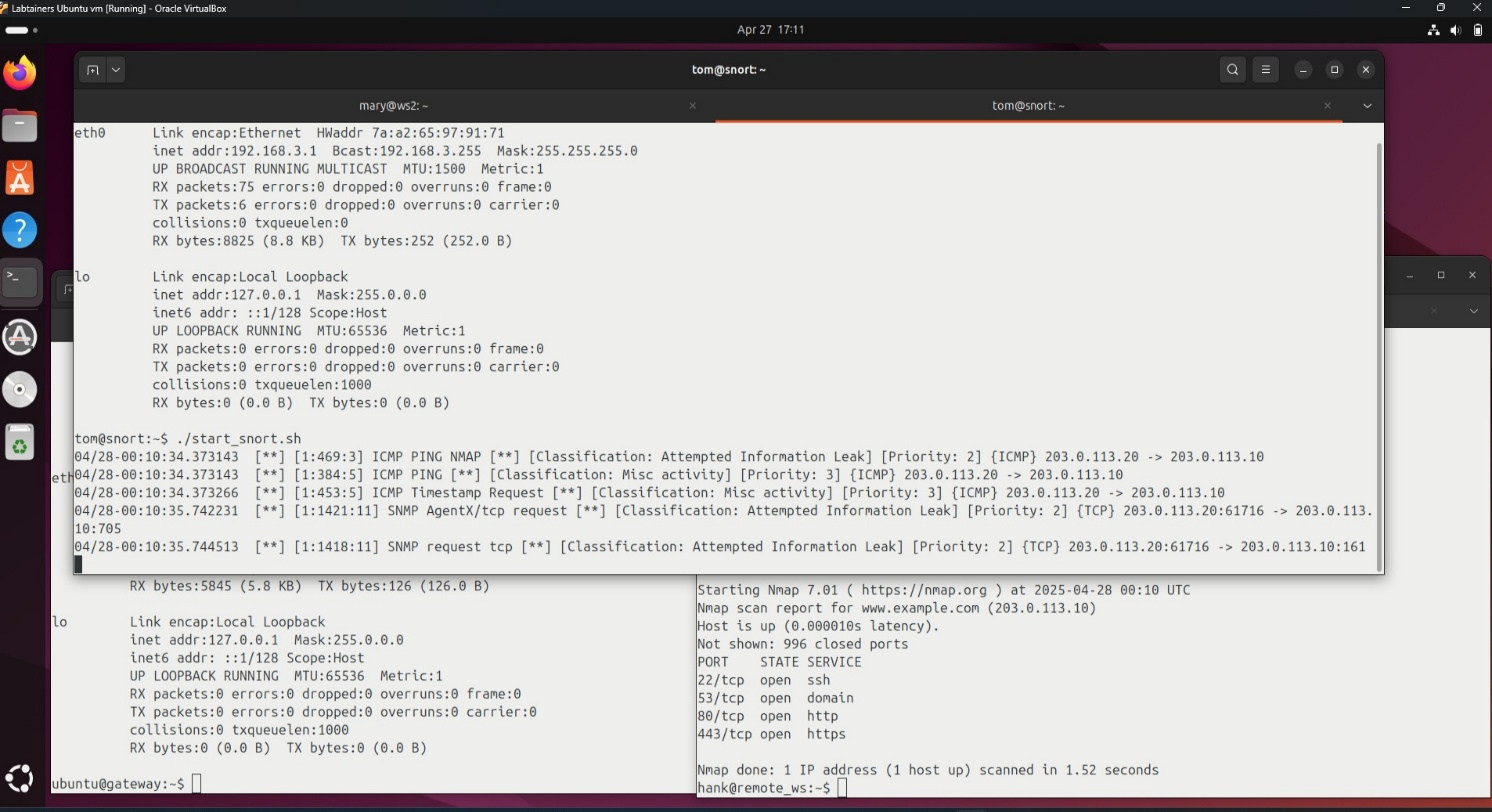
./start\_snort.sh

The Snort utility includes a set of pre-configured rules that create alerts for known suspicious network activity. The configuration on the snort component is largely as it exists after initial installation of the snort utility. To see an example of some of the preconfigured rules, perform an nmap scan of [www.example.com](http://www.example.com/) from the **remote workstation**:

sudo nmap [www.example.com](http://www.example.com/)

Note the alerts displayed at the snort console. The rules that generate these alerts can be seen, along with all rules, in /etc/snort/rules/

**Insert screen shot 1 of Snort warning here:**



## Write a simple (bad) rule (5 points)

Custom rules are typically added to the file at /etc/snort/rules/local.rules Stop snort and add a rule that generates an alert for each packet within a TCP stream. For example:

alert tcp any any -> any any (msg:"TCP detected"; sid:00002;)

That rule can be read as: “Generate an alert whenever a TCP packet from any address on any port is sent to any address on any port, and include the message tagged as msg:, and give the rule an identifier of 00002.” Then restart snort. Test this rule by starting Firefox on the remote workstation:

firefox [www.example.com](http://www.example.com/)

As you can see, the rule you wrote will overwhelm you with useless information. So, stop snort and delete the rule.

**Insert screen shot 2 of Snort warning here:**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

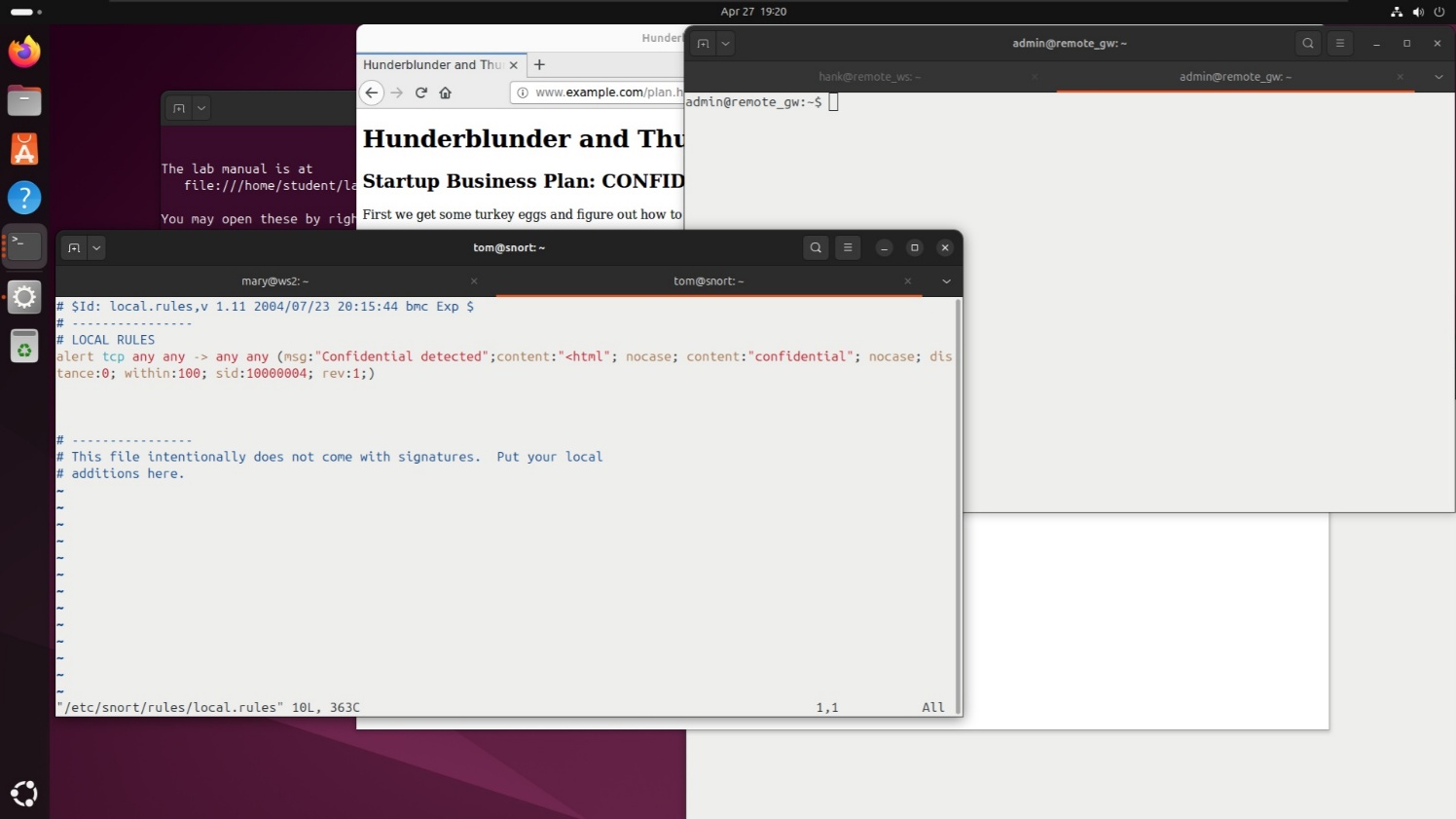
## Custom rule for CONFIDENTIAL traffic (5 points)

At the Firefox browser, which should be displaying the webpage from [www.example.com,](http://www.example.com/) we will display an unpublished web page that we know exists on the website. In particular, we have heard that the keen minds at the startup company have placed their confidential business plan at [www.example.com/plan.html.](http://www.example.com/plan.html) Take a look at it.

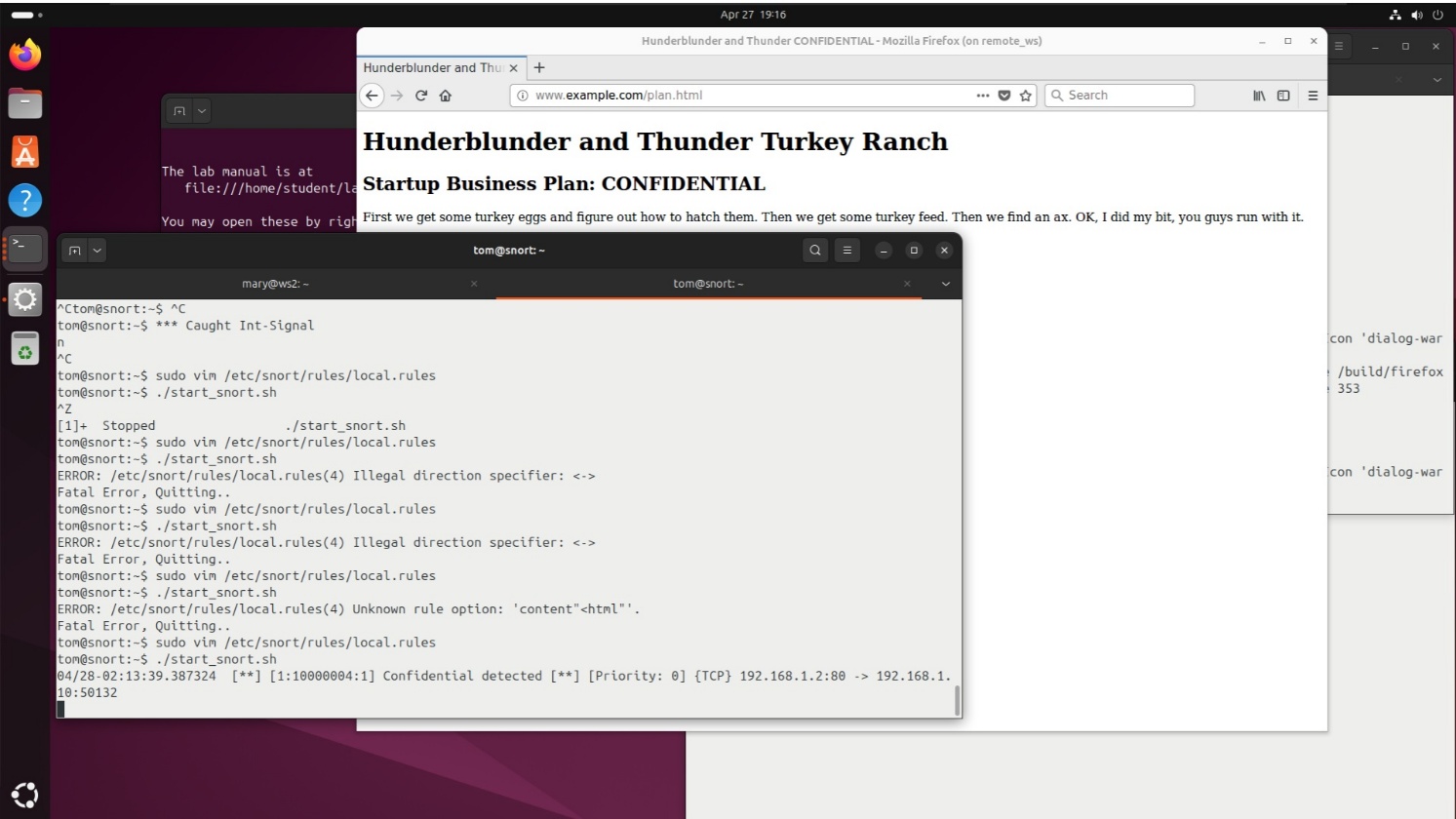
Now add a rule to your local.rules file on snort that will generate an alert whenever the text ”CON- FIDENTIAL” is sent out to the internet. Reference the snort manual [https://www.snort.org/](https://www.snort.org/#documents) [#documents](https://www.snort.org/#documents) or existing rules to understand how to qualify alerts based on *content*. Be sure to include the word ”CONFIDENTIAL” in the alert message, and give the rule its own unique sid. After adding the rule, restart snort.

On the browser at the remote workstation, clear your history (Menu / Preferences Security & Privacy), and then refresh the plan.html page. You should see an alert at the snort console.

4.4.1 Insert Your snort rule to alert “CONFIDENTIAL” in content



4.4.2 Insert screen shot 3 of snort displaying alert “CONFIDENTIAL” in content

:

## Effects of encryption

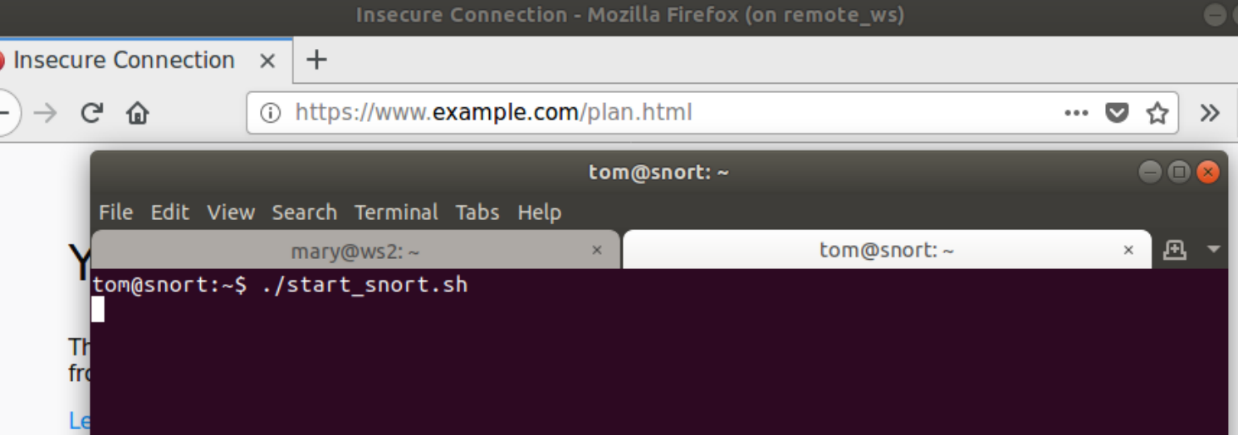
Back at the Firefox browser, again clear the browser history. Now alter the URL to make use of the web server SSL function. Change the url to [https://www.example.com/plan.html.](http://www.example.com/plan.html) Do you see a new snort alert? Why?

No, the Snort rule is looking for plain text. With https it is not able to read the packet data.

One solution to this problem is to use a reverse proxy in front of the web server. This reverse proxy would handle the incoming web traffic and manage the SSL connections. The web server would then receive only clear-text HTTP traffic, and outgoing traffic from the web server could then be mirrored to the IDS. We will not pursue that solution in this lab.

This would be what we would need to do if we were conducting a MiTM attack. 😊

Insert screen shot 4 of snort



## Watching internal traffic (5 points)

Go to the ws2 (mary) component and run nmap:

sudo nmap [www.example.com](http://www.example.com/)

What do you see on the snort component? Does it include the ICMP PING NMAP alert that you saw when the remote workstation ran nmap? Why not?

No I did not seem the ICMP PING NMAP. I believe it is because it is in the local network and not going through the router to the snort component.

Go to the gateway component and edit the /etc/rc.local script so that traffic from Mary’s work- station is mirrored to the snort component. You can do this by adding this line to the section of that file that defines the packet mirroring:

iptables -t mangle -A PREROUTING -i $lan2 -j TEE --gateway 192.168.3.1

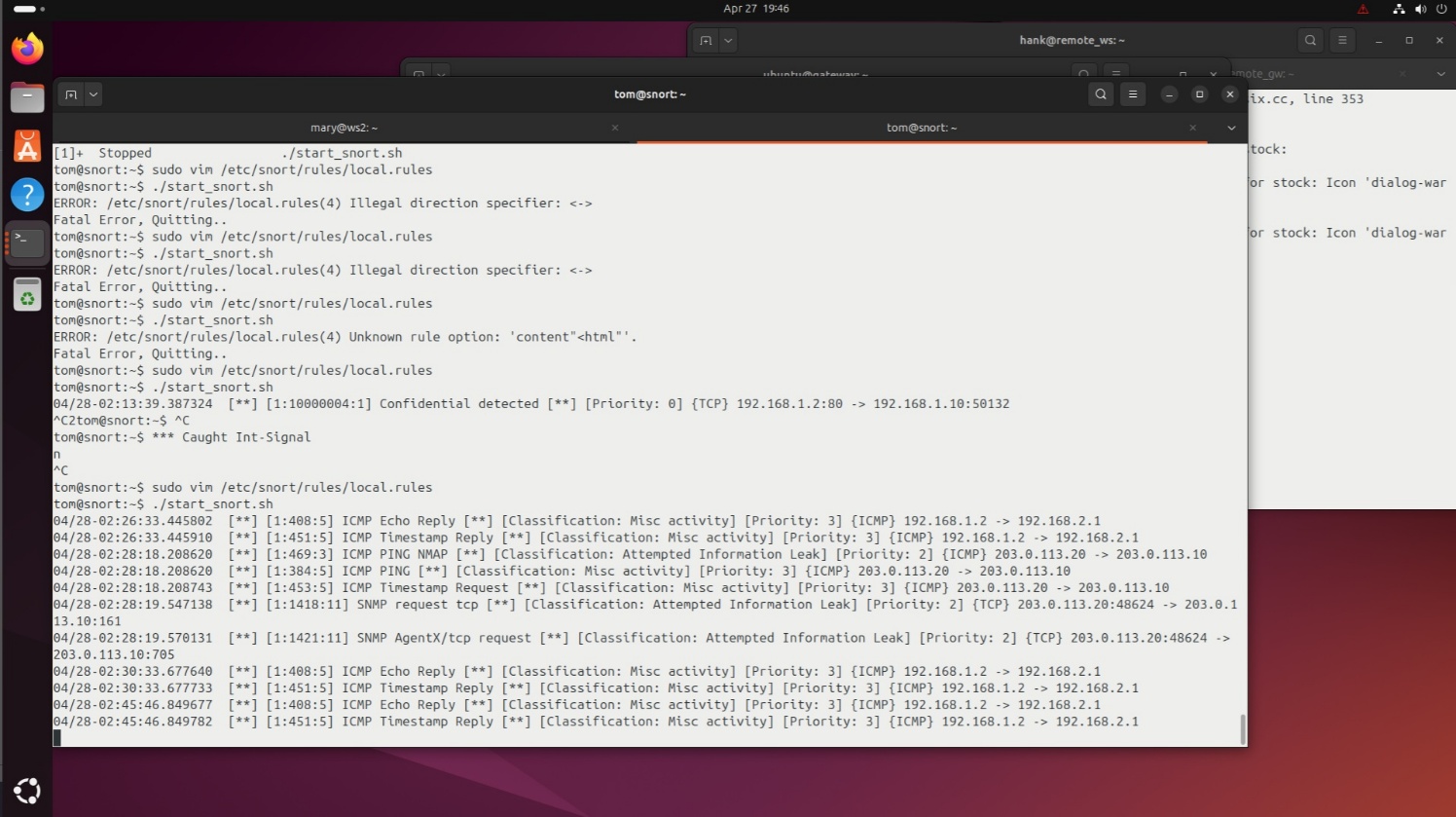
Then run the script to replace the iptables rules with your new rules :

sudo /etc/rc.local

Now restart snort and again run sudo nmap [www.example.com](http://www.example.com/)

from mary’s ws2 computer.

Insert screen shot 5 of snort



## Distinguishing traffic by address (10 points)

Start Firefox on mary’s ws2 computer to view the confidential business plan:

firefox [www.example.com/plan.html](http://www.example.com/plan.html)

Then observe the snort console. This will not do! The keen minds at the startup need to view their con- fidential business plan without IDS alerts firing off. But they do want to monitor internal computers for suspicious traffic, e.g., nmap scans. In this task, you will adjust your snort rule so that the CONFIDENTIAL alert only fires when the plan is accessed by addresses outside of the site.

If you review rules found in the /etc/snort/rules directory, you will see that rules have the general form of:

alert <protocol> <source\_addr> <src\_port> -> \

<dest\_addr> <dest\_port> <rule options in parens>

The snort rules include two address fields: source addr and dest addr. These addresses are used to check the source from which the packet originated and the destination of the packet. The address may be a single IP address or a network address. You likely have used the *any* keyword to apply a rule on all addresses. For network addresses, the address is followed by a slash character and number of bits in the netmask. For example, a network address of 192.168.2.0/24 represents C class network 192.168.2.0 with 24 bits in the network mask.

Note that as a result of our use of NAT, all traffic from the web server destined for an external address will have a destination address of the gateway, (i.e., 192.168.1.10), while web traffic destined for internal users will have destination addresses that match the internal user.

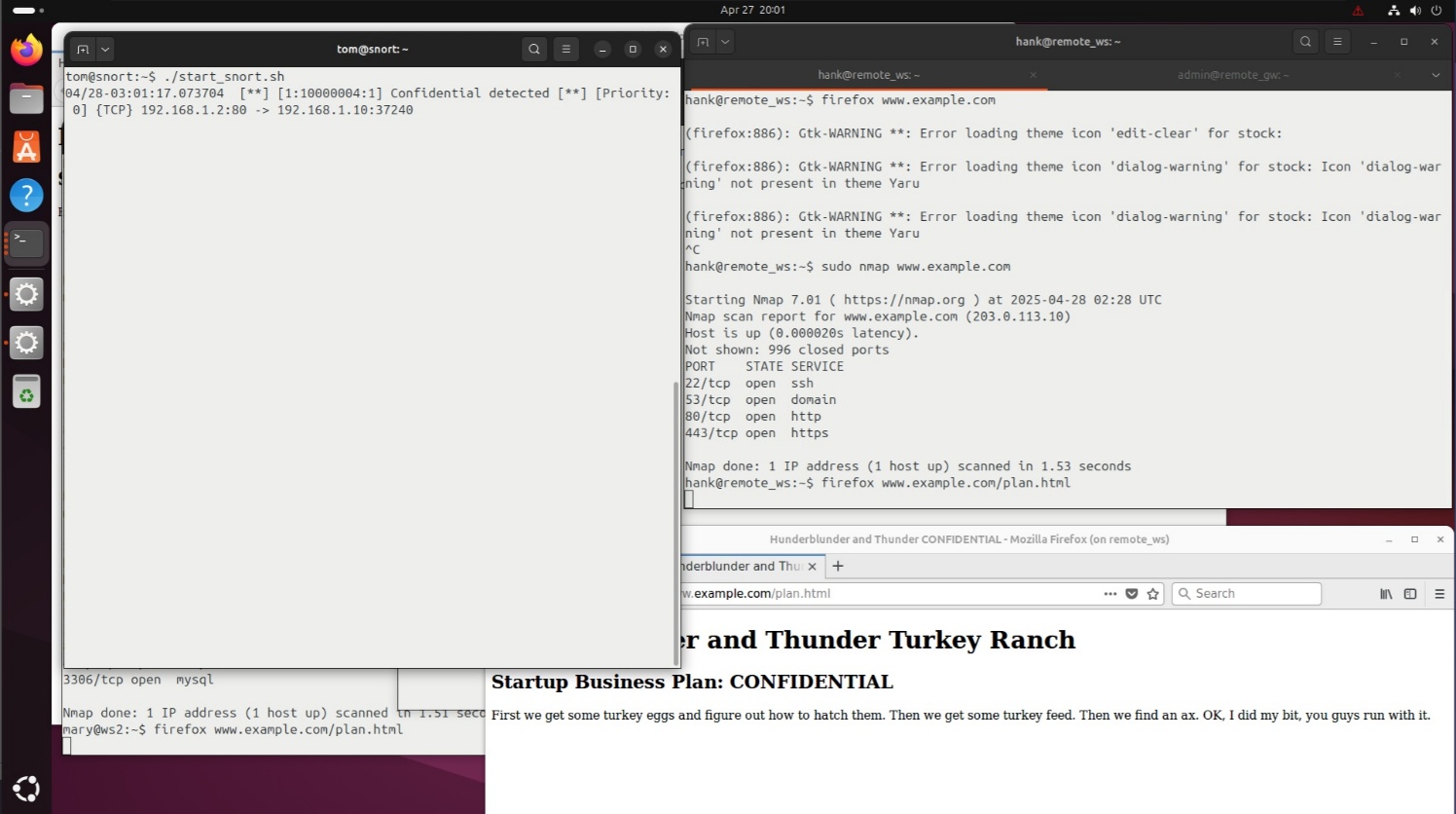
For this task, you must set your snort rules and traffic mirroring such that:

* + 1. External access to the business plan generates an alert;

**Insert your snort rule here:**

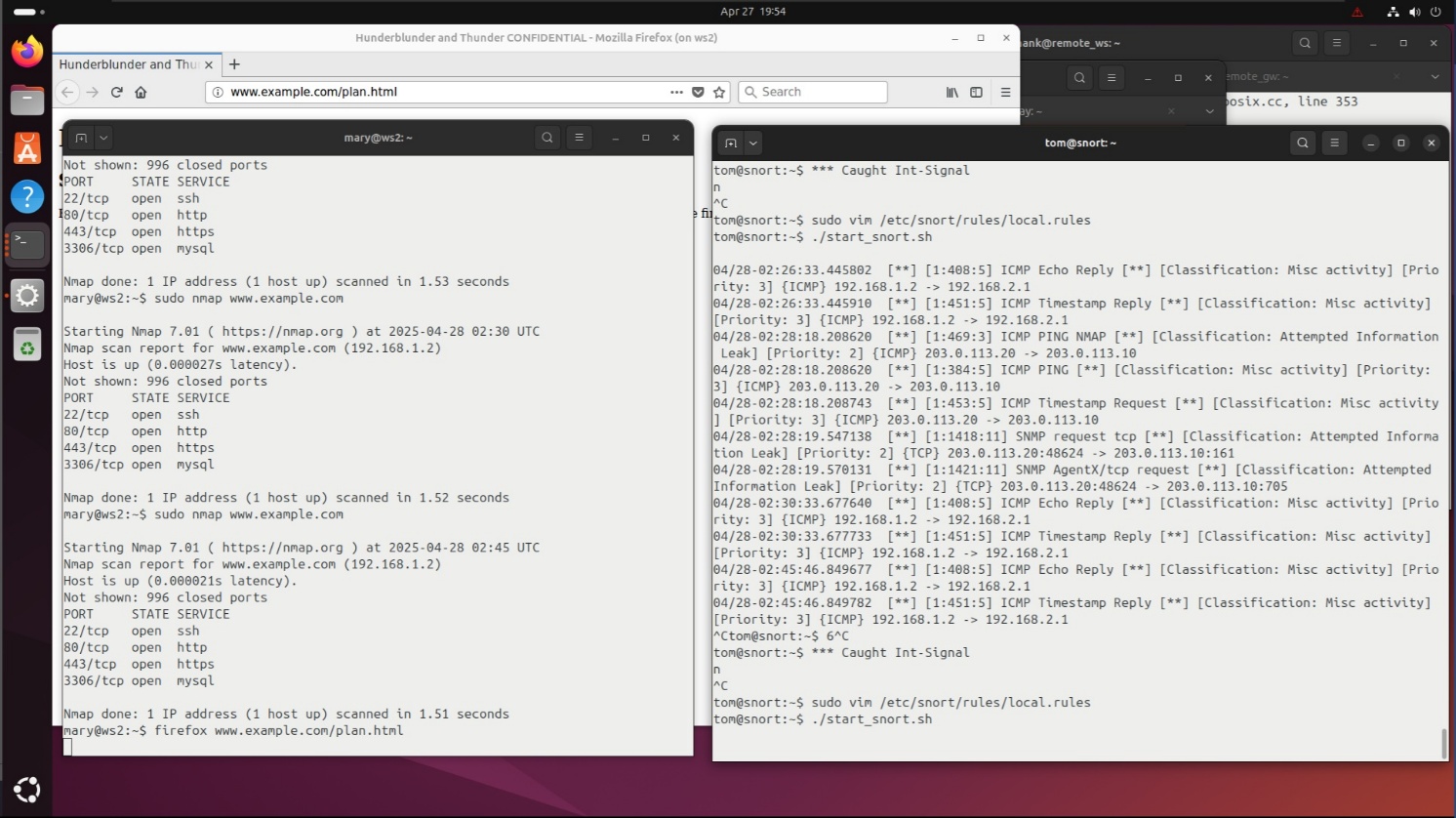
**Alert tcp any any -> 192.168.1.10 any (msg:”confidential detected” ; content:”<html”; nocase; content:”confidential”; nocase: distance:0; within:100; sid:10000004; rev:1;)**

Insert screen shot 6 of snort

****

* + 1. Internal access to the business plan does not generate an alert;

**Insert your snort rule here:**

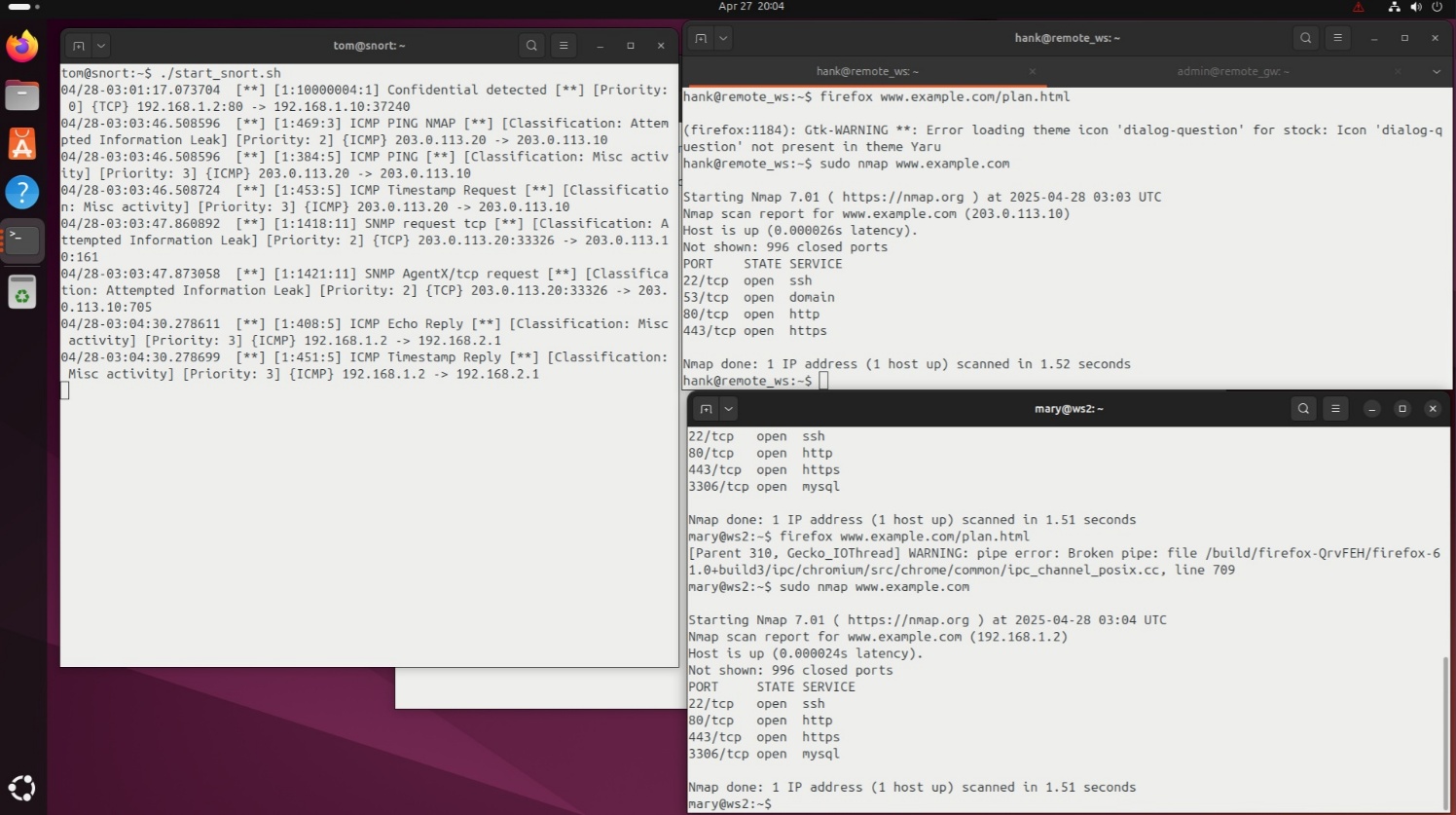
****

Insert screen shot 7 of snort

* + 1. External or internal use of nmap will generate an ICMP NMAP PING alert.

**Insert your snort rule here:**

Insert screen shot 8 of snort



Your must test each of these criteria during a single snort session, i.e., if you change a snort rule, or port mirroring, you must restart your tests.

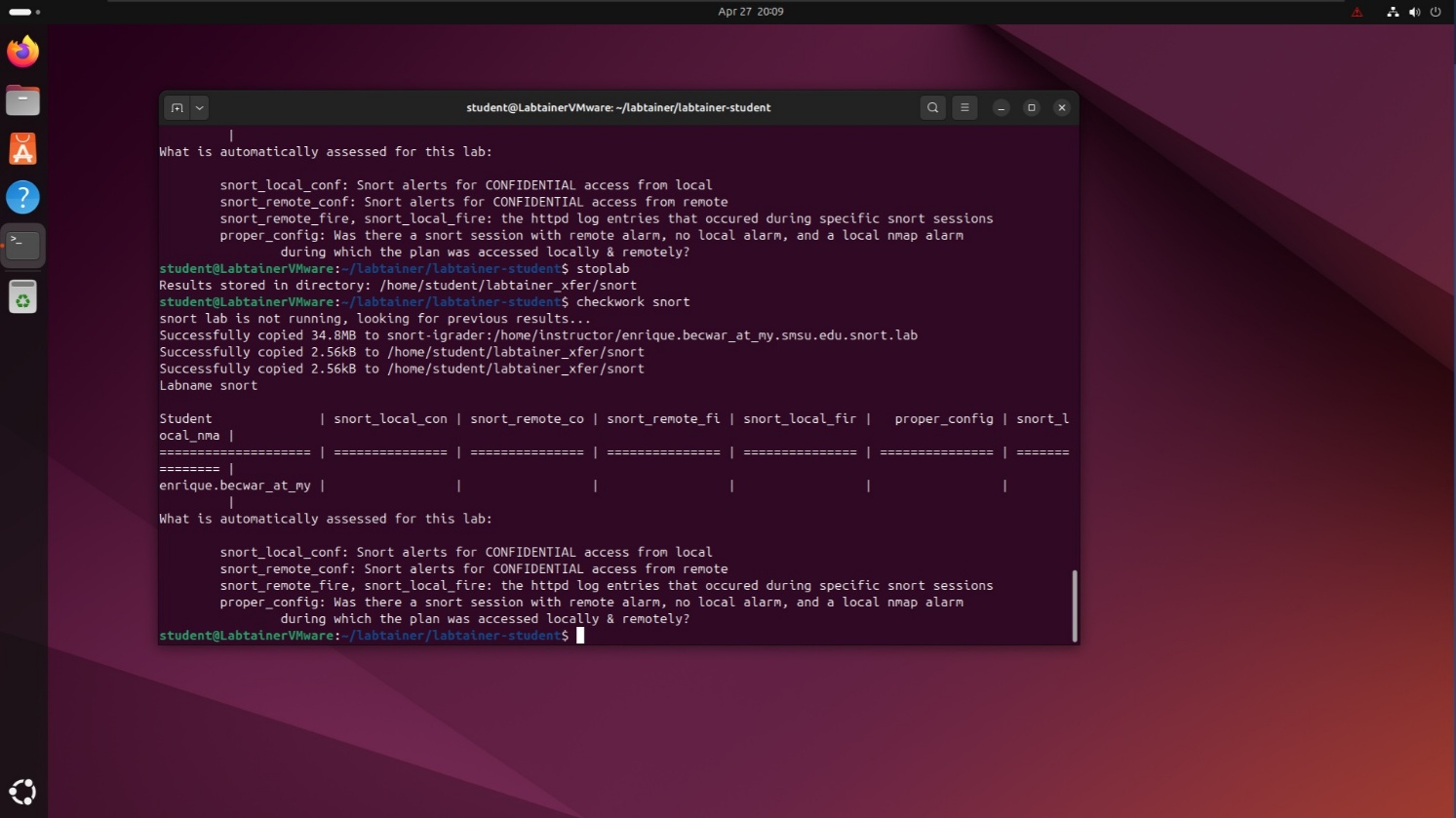
# Submission

After finishing the lab, go to the terminal on your Linux system that was used to start the lab and type:

checkwork

Insert screen shot 8 of grading report (You should see five Ys). If you already stopped the lab,

**The command is** checkwork snort



Then go to the terminal on your Linux system that was used to start the lab and type:

stoplab snort

When you stop the lab, the system will display a path to the .lab results on your Linux system. Upload that file and this report to D2L Assignment > Project 4 before 11:59 PM, 4/27/2025.

This project can be done in pairs for students enrolled in CYBR500.

**For COMP486, students have** **one of the two options**, please indicate your option in your report.

* Option I: Use project 4 to substitute only one of previous projects: Part 2 of project1 (up to 30 points), project 2 (up to 20 points), project 3 (up to 30 points).
* Option II: Get 10 bonus points for project category.