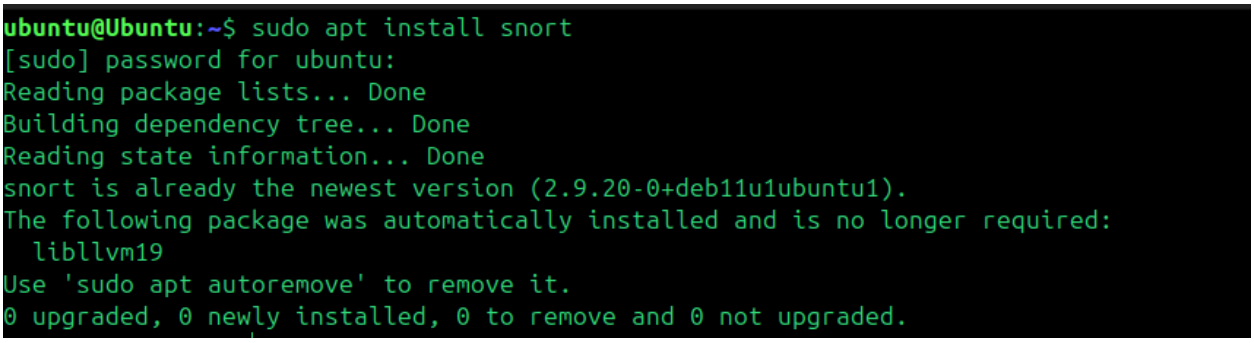
Snort IDS

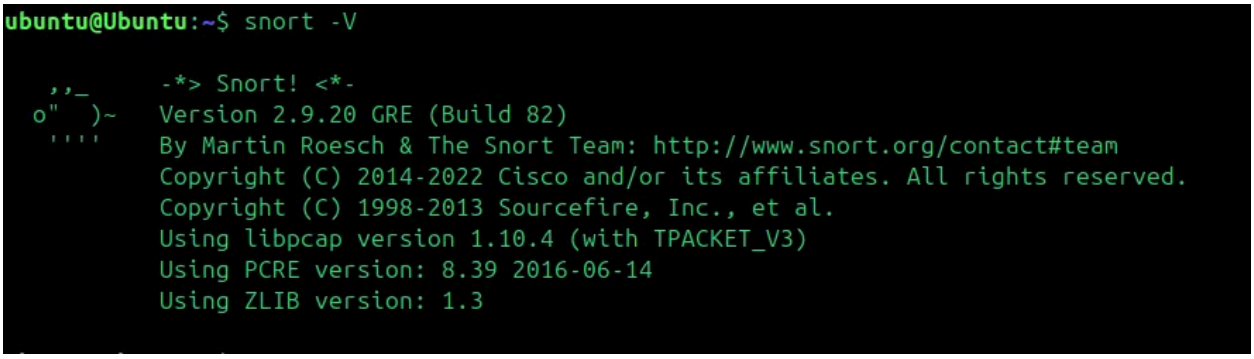
Implementation of Snort Intrusion Detection System on Ubuntu

1️⃣ Screenshots of installation & `snort -V` output

📸 Screenshot A: Snort Installation



📸 Screenshot B: Snort Version



2️⃣ snort.conf snippet showing RULE_PATH & HOME_NET

📸 Screenshot C: snort.conf

```
# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an absolute path,
# such as:  c:\snort\rules
var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules

# If you are using reputation preprocessor set these
# Currently there is a bug with relative paths, they are relative to where snort is
# not relative to snort.conf like the above variables
# This is completely inconsistent with how other vars work, BUG 89986
# Set the absolute path appropriately
var WHITE_LIST_PATH /etc/snort/rules
var BLACK_LIST_PATH /etc/snort/rules

##################################################
# Step #2: Configure the decoder.  For more information, see README.decode
##################################################

# Stop generic decode events:
config disable_decode_alerts

# Stop Alerts on experimental TCP options
config disable_tcpopt_experimental_alerts
```

```
# site specific rules
include $RULE_PATH/local.rules

# The include files commented below have been disabled
# because they are not available in the stock Debian
# rules. If you install the Sourcefire VRT please make
# sure you re-enable them again:

#include $RULE_PATH/app-detect.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
#include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/browser-chrome.rules
#include $RULE_PATH/browser-firefox.rules
#include $RULE_PATH/browser-ie.rules
#include $RULE_PATH/browser-other.rules
#include $RULE_PATH/browser-plugins.rules
#include $RULE_PATH/browser-webkit.rules
include $RULE_PATH/chat.rules
#include $RULE_PATH/content-replace.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/experimental.rules
```

```
#
# Note to Debian users: this value is overriden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET 192.168.18.0/24

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS $HOME_NET

# List of web servers on your network
ipvar HTTP_SERVERS $HOME_NET

# List of sql servers on your network
ipvar SQL_SERVERS $HOME_NET
```

3️⃣ Your 5–6 custom rules in `local.rules`

📸 Screenshot D: `local.rules`

```
ubuntu@Ubuntu:~$ sudo cat /etc/snort/rules/local.rules
[sudo] password for ubuntu:

# 1. ICMP Ping Detection
alert icmp any any -> $HOME_NET any (msg:"ICMP Ping Detected"; itype:8; sid:1000001; rev:1;)

# 2. SSH Connection Attempt
alert tcp any any -> $HOME_NET 22 (msg:"SSH Connection Attempt"; flags:S; sid:1000002; rev:1;)

# 3. HTTP Traffic Detection
alert tcp any any -> $HOME_NET 80 (msg:"HTTP Traffic Detected"; sid:1000003; rev:1;)

# 4. FTP Attempt Detection
alert tcp any any -> $HOME_NET 21 (msg:"FTP Attempt Detected"; flags:S; sid:1000004; rev:1;)

# 5. Telnet Attempt Detection
alert tcp any any -> $HOME_NET 23 (msg:"TELNET Attempt Detected"; flags:S; sid:1000005; rev:1;)

# 6. Port Scan Detection
alert tcp any any -> $HOME_NET any (flags:S; threshold:type both, track by_src, count 10, seconds 5; msg:"Possible Port Scan Detected"; sid:1000006; rev:1;)
ubuntu@Ubuntu:~$
```

4️⃣ 3+ Screenshots of generated alerts (ICMP, SSH, HTTP)

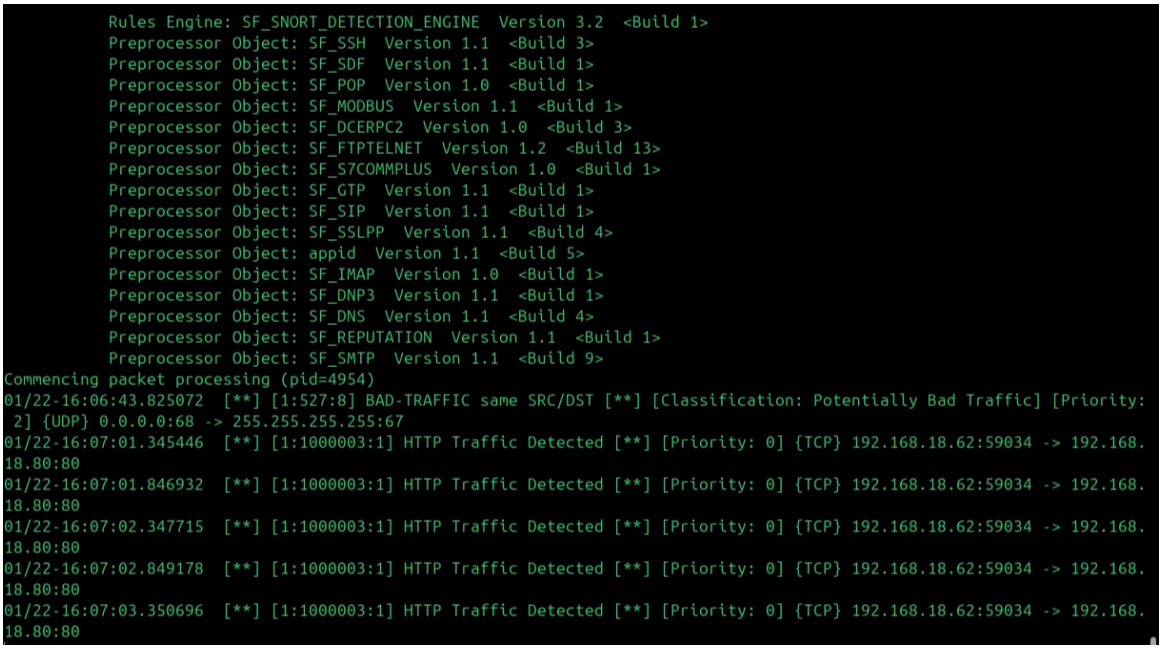
📸 Screenshot E: ICMP Alert

```
  ,,_     -*> Snort! <*-
 o"  )~   Version 2.9.20 GRE (Build 82)
  ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
          Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
          Copyright (C) 1998-2013 Sourcefire, Inc., et al.
          Using libpcap version 1.10.4 (with TPACKET_V3)
          Using PCRE version: 8.39 2016-06-14
          Using ZLIB version: 1.3

          Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.2  <Build 1>
          Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
          Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
          Preprocessor Object: SF_POP  Version 1.0  <Build 1>
          Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
          Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
          Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
          Preprocessor Object: SF_S7COMMPLUS  Version 1.0  <Build 1>
          Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
          Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
          Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
          Preprocessor Object: appid  Version 1.1  <Build 5>
          Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
          Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
          Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
          Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
          Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
Commencing packet processing (pid=4647)
01/22-15:58:12.442088  [**] [1:1000001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.18.62 -> 192.168.18.80
01/22-15:58:13.444636  [**] [1:1000001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.18.62 -> 192.168.18.80
01/22-15:58:14.448169  [**] [1:1000001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.18.62 -> 192.168.18.80
01/22-15:58:15.452836  [**] [1:1000001:1] ICMP Ping Detected [**] [Priority: 0] {ICMP} 192.168.18.62 -> 192.168.18.80
```

📸 Screenshot F: SSH Alert

```
          Using PCRE version: 8.39 2016-06-14
          Using ZLIB version: 1.3

          Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.2  <Build 1>
          Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
          Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
          Preprocessor Object: SF_POP  Version 1.0  <Build 1>
          Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
          Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
          Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
          Preprocessor Object: SF_S7COMMPLUS  Version 1.0  <Build 1>
          Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
          Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
          Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
          Preprocessor Object: appid  Version 1.1  <Build 5>
          Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
          Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
          Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
          Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
          Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
Commencing packet processing (pid=4909)
01/22-16:05:06.555046  [**] [1:1000002:1] SSH Connection Attempt [**] [Priority: 0] {TCP} 192.168.18.62:58960 -> 192.168
.18.80:22
01/22-16:05:07.055399  [**] [1:1000002:1] SSH Connection Attempt [**] [Priority: 0] {TCP} 192.168.18.62:58960 -> 192.168
.18.80:22
01/22-16:05:07.555873  [**] [1:1000002:1] SSH Connection Attempt [**] [Priority: 0] {TCP} 192.168.18.62:58960 -> 192.168
.18.80:22
01/22-16:05:08.056963  [**] [1:1000002:1] SSH Connection Attempt [**] [Priority: 0] {TCP} 192.168.18.62:58960 -> 192.168
.18.80:22
01/22-16:05:08.558401  [**] [1:1000002:1] SSH Connection Attempt [**] [Priority: 0] {TCP} 192.168.18.62:58960 -> 192.168
.18.80:22
```

📸 Screenshot G: HTTP Alert



5️⃣ Brief Report (200–300 words): Challenges faced & what you learned

Implementation of Snort Intrusion Detection System on Ubuntu

In this experiment, Snort was installed and configured on an Ubuntu virtual machine to function as a Network Intrusion Detection System (NIDS). The objective of the experiment was to monitor network traffic and detect suspicious activities using custom-defined rules. Snort was installed using the APT package manager, and the snort.conf file was configured by defining the HOME_NET variable and rule paths.

Custom rules were created in the local.rules file to detect ICMP ping requests, SSH connection attempts, HTTP traffic, FTP and Telnet attempts, and potential port scanning behavior. Network traffic was generated from the host system using tools such as ping, ssh, and curl to test the effectiveness of these rules. Snort successfully detected the traffic and generated real-time alerts, confirming that the configuration and rules were functioning correctly.

Several challenges were faced during the experiment, including identifying the correct network interface, handling IP address changes after switching to bridged networking, and initially missing custom rules in the local.rules file. These issues were resolved by verifying network settings, updating the HOME_NET configuration, and validating Snort using test mode.

Through this experiment, I gained practical experience in intrusion detection systems, custom rule creation, and real-time network traffic analysis. This lab improved my understanding of how Snort monitors and detects network-based attacks in a real-world environment.