Campbell University
UNDERGRADUATE SCHOOL

INTEGRATED STUDIES
ITS 465: Senior Project

NIDS INSTALLATION FOR CLINICAL RESEARCH LAB

SUBMITTED TO

Dr. Umesh Varma

A SYSTEMS DEVELOPMENT PROJECT PRESENTED TO CAMPBELL UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE BACHELORS IN INFORMATION TECHNOLOGY SECURITY PROGRAM

BY

Michaela Pierce

April, 2022

Abstract

Barker Labs is a clinical research lab that has created a cure-all vaccine for COVID-19, a one-time vaccine that will prevent COVID-19 and all relative variants. The vaccine is undergoing human testing but due to other competitors has become prominent, because of other vaccines requiring more than one dose, failure to prevent relative variants, major side effects, and unidentified future complications. Barker Labs was started two years ago and have grown over time along with their clinical research. The Information Technology (IT) Department have maintained the same Network Intrusion Detection System (NIDS) for the past two years. Threats were not a major concern due to the business being small and other competitors publishing their findings and making vaccines available to the public. A new NIDS will be installed to help provide security for Barker Lab's Network.

Table of Contents

Introduction

Barker Labs is a clinical research lab that has created a cure-all vaccine for COVID-19, a one-time vaccine that will prevent COVID-19 and all relative variants. The vaccine is undergoing human testing but due to other competitors has become prominent, because of other vaccines requiring more than one dose, failure to prevent relative variants, major side effects, and unidentified future complications. Barker Labs was started two years ago and have grown over time along with their clinical research. The Information Technology (IT) Department have maintained the same Network Intrusion Detection System (NIDS) for the past two years. Threats were not a major concern due to the business being small and other competitors publishing their findings and making vaccines available to the public. The current NIDS is outdated and unable to identify new threats and provide information that could help improve the other components of cybersecurity for the business. A new NIDS system will be implemented and configured to help the IT Department make decisions for other established prevention countermeasures that can increase the network security for the company.

Preliminary Investigation

Project Initiation
Initiation Team:

Role	Responsibility	Assigned To
Project Manager	The Project Manager is responsible for implementing the Project Risk Management Plan and reporting to the Project Sponsor.	Michaela Pierce
Project Owner	The Project Owner is responsible for providing managerial assistance to the Project Manager and team.	David Smith – IT Director
Project Sponsor	The Project Sponsor is responsible for discussing business needs and to sign off on all major project deliverables.	John Barker – Business Owner

Problems

- The system will experience downtime during implementation
- Possible threat risk for business until new installation
- Current system is no longer serving necessary purpose
- System will need to be configured to fit business needs accordingly

Opportunities, and Directives

-Alerts will be sent to authorized officials that can block the flagged traffic

-Reports can educate the IT Department to better understand the threats received

-Report information can be used to improve other network security components in the business

-Business production will increase due to prioritized network security

-System can be configured to fit business needs and tweaked if needs change

Project Charter

Project Name: NIDS Installation for Clinical Research Lab
Company: Barker Labs
Division: Computer Network
Department: Information Technology

Prepared by: Michaela Pierce
Project Manager: Michaela Pierce
Project Owner, IT Director: David Smith
Project Sponsor, Business Owner: John Barker

Version Control

0.1 Charter created by Michaela Pierce – 02/10/2022
0.2 Charter edited by David Smith – 02/12/2022, Scope affected/ reduced

Form 1

System Service Request (SSR)

BARKER LABS

SYSTEM SERVICE REQUEST

Requested by: David Smith Date: January 25, 2022

Department: Information Technology

Location: 120 Thornton LN, NC, 28575

Phone Number: 585-235-2389 Email: dsmith@barkerlabs.gmail.com

Request Type (If other, please specify below):
Other: _____

[ ] System Upgrade / Update [ ] System Repair

[ ] System Troubleshooting/ Diagnosis [X] New System

Problem Statement:

The Network Intrusion Detection System (NIDS) for Barker Labs is outdated and does not fulfill business needs. A new system should be considered, one that includes customer support unlike the current system.

Service Request:

I request that the Network Security be considered and prioritized to allow for continuous business growth. The NIDS system is no longer serving the duties required to protect the assets and clinical research the lab stores and maintains. Network threats will be more prevalent due to the business growing and receiving media attention.

Sponsor: David Smith

Owner: John Barker

Project Manager: Michaela Pierce

Estimated Start Date: February 3, 2022

Contact: Phone: [808-444-5566] email: [michaelap@gmail.com]

This document is used to request services needed for the business and to provide a written explanation and statement of the issue/ request. This document must be signed by the Owner for it to be of official use.

Project Charter Purpose

This project charter defines the scope, objectives, deliverables, budget, and overall approach for the work to be completed. This is the master document for the project and the single point of reference for scope, goals and objectives, organization, estimates, deliverables, and budget.

There are items within this Charter that will be supported by additional project control documents. These documents will be listed along with the purpose they intend to serve. This will keep changes to the master document to a minimum while providing a way to track and control the ongoing changes throughout the project life cycle.

This Project Charter serves as a contract between the Project team and the Project Sponsors, stating at a minimum:

-Purpose of Project and goal

-Assumptions and Constraints

-Deadline Date

-Deliverables

-Tasks that are out of the Scope

-Deliverables deadline and process

-Actors involved

-Necessary Resources and Budget

-Risks

-Process Progression

Executive Overview

Business growth has prioritized security, while there is physical security within the business, the network security needs improvement. The current network detection security system is outdated and no longer captures and alerts users for detected problem traffic.

Project Scope

This section provides the goals and objectives that are projected for the new system after implementation.

Goals & Objectives for the new system

Goals	Objectives
Customization for business needs	Configuration for alerts, rules, data logs, network policies, and for authorized users
Fulfill necessary job duties	Provide real-time traffic threat detection and alerts
Log reports	Log traffic
System will be user-friendly	System will not require a major learning curve

Project Scope Statement:

The installation of a new Network Intrusion Detection System (NIDS) will detect threats for the organization when conducting research, transferring and communicating data, and storing data. The overall goal is to install a modern, lightweight NIDS that will replace the old system and can be configured to use the community or built-in rules. Customization can be added but will not be a part of this project.

Project Deliverables:

The new system needs to detect modern attacks and alert the IT Department when there are threats detected in real time. The system should be user friendly and not require steep learning for use. A command-line interface will be used and can be modified in the future.

There will be weekly meetings to discuss the project plan and changes made

This charter and any further documentation should be stored in this folder:

H:\NIDS\process\Feb-May22 Charter\...

Deliverables out of Scope:

- There will be no new networking hardware installed or purchased for the new system.
- Internet access will not be available during the system implementation.
- Customizable Configurations will not be a part of the initial system implementation.
- Types of threats and prevention methods
- Alterations to increase protection in other areas than Network Detection

Product Selection

Research for Products that will be Installed. The NIDS that will be selected will aid in functionality for the business and can be improved at a later time.

Project Estimated Budget

The projected estimated cost of the project should not exceed $2,500

Project Conditions and Controls

Director as needed to meet security needs. Business Owner will be required for all changes and approval throughout the project.

Project Constraints:

- Cyberthreat possibility during downtime, system implementation
- Software may require additional components
- New system will require a larger budget than current system
- Project Deadline must be met

Project Assumptions:

- The new software will use the Snort software
- The system will be within the budget
- Detection of threats will provide the opportunity to increase network security
- System will be customized by IT

Project Interdependencies:

There are currently no additional projects that are in process or planned that have a relation with this project.

Organizational Impact:

Organization	Impact to and Participation of Organization
Accounting	Conduct Budget plan for project
IT	Learn how to analyze system reports Learn configuration requirements
All departments	Learn Cyberthreat prevention regarding email and internet etiquette

Project Risk Management:

This section of the charter specifies the Risk Management Plan, comprising activities, roles and responsibilities.

Activity	Description	Ownership
Documentation	Meeting will occur to discuss and monitor Risks	Project Manager Project Owner Project Sponsor
Commentary	Weekly meetings will allow for updates on the project progression and allow for customization if needed	Project Manager

Role	Responsibility	Assigned To
Project Manager	The Project Manager is responsible for implementing the Project Risk Management Plan and reporting to the Project Sponsor.	Michaela Pierce
Project Owner	The Project Owner is responsible for providing managerial assistance to the Project Manager and team.	David Smith – IT Director
Project Sponsor	The Project Sponsor is responsible for discussing business needs and to sign off on all major project deliverables.	John Barker – Business Owner

Non-People Resources Required:

Item	Description	Quantity	Comments
Office	Private Office	1	Need office for Project Manager and Project Team
Meeting Room	Room big enough to hold 10 people, and will be used to discuss Project Plan and Execution	1	Will be used for weekly meetings
LAN Layout	The Layout of the Local Area Network showing where each component resides to make up the network	N/A	Need for project use to help understand the network better

Project Scope Change:

This charter's purpose is to define the scope and the process plan. While the scope and out of scope deliverables are defined there is always a risk of scope creep that can affect deliverables and result in project failure.

This section is necessary for helping to prevent project failure due to scope creep. The project needs to fully understand the scope and disregard opportunities that are outside of the scope and will not be applicable to this project.

There will be a special form that will serve as a request for scope change if a change is needed. The form will be maintained by the Project Manager and the Project Sponsor will be required to approve changes requested. This will help track changes requested and changes made to ensure the scope is still applicable to this project.

Project Wrap-Up:

This section is for the Project Manager to fill out after the project is completed. A list of major project events is provided below if needed for assistance post project.

This section will include material that can benefit future projects and implementation. It will provide tasks that can be improved, tasks that went well and did not go well, and recommended tasks for future projects regarding this system.

April 26, 2022- The system has been implemented. Snort was installed and configured using built-in community rules. The Snort Configuration file was edited to provide a working system. Future project recommendations are:

- Customization for rule set to limit false positives and false negatives
- Graphical User Interface (GUI) can be implemented to provide an easy to interact interface for users and allow for reports to be organized and detailed
- Database Management System to store and finalize data for reports

The project required several libraries that would be needed for Snort and the other software to run smoothly. The library downloads took account for 60 % of the project, while the rest was configuring Snort to work properly.

Figure A1

Gantt Chart

ID	Task Name	Start	Finish	Duration	2022
					Jan Feb Mar Apr May Jun Jul
1	Project Management	2/1/2022	4/26/2022	61d	
2	Preliminary Investigation	2/1/2022	2/7/2022	5d	
3	Problem Analysis	2/8/2022	2/21/2022	10d	
4	Requirements Analysis	2/22/2022	3/7/2022	10d	
5	Decision Analysis	3/8/2022	3/14/2022	5d	
6	Design	3/15/2022	3/28/2022	10d	
7	Construction	3/29/2022	4/11/2022	10d	
8	Implementation	4/12/2022	4/18/2022	5d	
9	Operations and Support	4/12/2022	4/26/2022	11d	

Problem Analysis

Project Name: NIDS Installation for Clinical Research Lab
Company: Barker Labs
Division: Computer Network
Department: Information Technology

Prepared by: Michaela Pierce
Project Manager: Michaela Pierce
Project Owner, IT Director: David Smith
Project Sponsor, Business Owner: John Barker

Version Control

0.1 Charter created by Michaela Pierce – 02/10/2022
0.2 Charter edited by David Smith – 02/12/2022, Scope affected/ reduced
0.3 Problem Analysis created by Michaela Pierce – 02/20/2022
0.4 Problem Analysis edited by David Smith – 02/21/2022
0.5 Feasibility Analysis added to Problem Analysis, by Michaela Pierce – 02/26/2022

System Analysis

This section provides an analysis of the current system Barker Labs contains. Analyzing the system and business process is ideal for understanding and implementing a system that will benefit Barker Labs as well as not prevent growth by limiting daily business processes.

Current System : Wicker

Version: 1.2.0

Wicker is the current NIDS system installed. This system is outdated and lacks features of modern NID Systems. Wicker uses a command line interface and includes rule lists; however, they are rarely updated.

Flaws:

- Detects false positives / false negatives
- Outdated Signature Library
- Alert Configuration failure
- Not user-friendly
- Undetailed Reports
- Rules are not updated, no support

Cost:

Currently Barker Labs spends $80/month for Wicker's system.

Wicker does not offer remote customer support which can make diagnosing an issue inconvenient and difficult to achieve when scheduling appointments for techs to come and fix the system.

Requirements Determination

The Project Owner, Project Sponsor, and the IT Manager are in charge of requirements.

Process

The old system will be replaced with Snort, the new system. The new system will be configured using the community rules and snort configuration for setting up the default usage for the computer network. There will be some downtime during the installation in which some programs may not be completely accessible.

Data will be stored and implemented as usual following business practices. The data collected from Snort can be stored within additional software that can be installed to better manage the traffic logs. There will be no additional software or configuration rules implemented in this project.

Fact Finding and Data Gathering

Snort is the best recommendation due to the software being open source and providing additional snort rules that can be installed and configured further to provide security to the network.

Requirements Analysis

Project Name: NIDS Installation for Clinical Research Lab
Company: Barker Labs
Division: Computer Network
Department: Information Technology

Prepared by: Michaela Pierce
Project Manager: Michaela Pierce
Project Owner, IT Director: David Smith
Project Sponsor, Business Owner: John Barker

Version Control

0.1 Charter created by Michaela Pierce – 02/10/2022
0.2 Charter edited by David Smith – 02/12/2022, Scope affected/ reduced
0.3 Problem Analysis created by Michaela Pierce – 02/20/2022
0.4 Problem Analysis edited by David Smith – 02/21/2022
0.5 Feasibility Analysis added to Problem Analysis, by Michaela Pierce – 02/26/2022
0.6 Requirements Analysis added by Michaela Pierce – 03/01/2022

Business Requirements Purpose

This business requirements document will contain the requirements that need to be met for the project to be completed successfully.

This document will contain the following:

- Business Rules
- Business Requirements
- Problems and Opportunities
- Problem Statements
- Business Processes
- System Improvement Objectives
- Project Plan Update
- Findings and Recommendations

Business Rules

This section contains the business rules.

- The IT director and the business owner have the authority and permissions to update these rules as needed.
- The IT Administrators will be responsible for the following tasks:
    - -Monitor logs daily
    - -Analyze traffic patterns
    - -Identify false positives
    - -Examine reports
    - -Properly respond to threats/alerts if IT Manager and IT Director are unavailable
    - -Notify IT Director or IT Manager of changes, threats, etc.

- The IT Manager is responsible for the following tasks:

  -Perform maintenance checks

  -Analyze reports

  -Respond to high-level threats if IT Director is unavailable

  -Acknowledge all alerts received detecting threats

  -Report findings to IT Director

-The IT Director is responsible for the following tasks:

  -Customize configuration rules

  -Perform updates

  -Perform maintenance checks

  -Analyze reports

  -Use threat findings to make changes to security system to eliminate further attacks

  -Respond to high-level threats

  -Acknowledge all alerts received detecting threats

  -Communicate findings with business owner

Business Requirements

This section defines the business requirements:

- Detected threats should send an alert to the IT Director, if unavailable, send an alert to IT Manager, if unavailable, send alert to IT Administrators

- Perform well with moderate volumes of traffic

- Faster response time

- Detailed Reports

- User-friendly interface

Problem Domain

Network traffic can be harmless and harmful and should be monitored to understand what kind of traffic is present within the network. A NIDS system not only monitors traffic but also will alert authorized users when there is a positive threat detected. The current system is outdated and lacking key components, therefore, a new system will be installed.

Problems and Opportunities

This section defines the problems and opportunities.

Problems with the current system are the following:

- False positives and false negatives are detected
- Only tolerates low volumes of traffic
- Fails to detect newer attacks

Problems with implementation of new system:

- The system will face downtime during the installation of the new system

Opportunities are the following:

- Data from reports can be used to increase security by modifying the security system installed
- Understand the risks of the threats detected to improve other security measures
- Increased business production

Problem Statements Chart

Brief Statements of Problem, Opportunity, or Directive	Urgency	Visibility	Annual Benefits	Priority or Rank	Proposed Solution
False positives or false negatives are occurring within the system.	ASAP	High	NA	1	New system
Late detection alerts	1 month	High	NA	1	New system
Negligence of network and business	1 month	High	NA	1	New system
Reports are not detailed or accurate	1 month	High	NA	2	Configuration rules need to be changed
Attacks are a disruption to the business	1 month	High	NA	1	New system
Risk of data loss	1 month	High	NA	1	New system
Risk of System failure	1 month	High	NA	1	New system

Business Processes

The installation of the NIDS will require the network to be down during a period of time. This will be planned downtime and all preventative steps will be taken to secure the network. The old system will be uninstalled and the new system will be installed.

System Improvement Objectives

- The system can be modified and additional software can be added.

Findings and Recommendations

A new system is the best option for having a successful NIDS. The new system can be modified according to business needs and user preference. The reports will be detailed and false positives and negatives can be decreased by changing the configuration rules.

Decision Analysis

Project Name: NIDS Installation for Clinical Research Lab
Company: Barker Labs
Division: Computer Network
Department: Information Technology

Prepared by: Michaela Pierce
Project Manager: Michaela Pierce
Project Owner, IT Director: David Smith
Project Sponsor, Business Owner: John Barker

Version Control

- 0.1 Charter created by Michaela Pierce – 02/10/2022
- 0.2 Charter edited by David Smith – 02/12/2022, Scope affected/ reduced
- 0.3 Problem Analysis created by Michaela Pierce – 02/20/2022
- 0.4 Problem Analysis edited by David Smith – 02/21/2022
- 0.5 Feasibility Analysis added to Problem Analysis, by Michaela Pierce – 02/26/2022
- 0.6 Requirements Analysis added by Michaela Pierce – 03/01/2022
- 0.7 Decision Analysis created by Michaela Pierce – 03/08/2022
- 0.8 Decision Analysis edited by David Smith – 03/10/2022

Identify Candidate Solution

Install a new NIDS – Snort

Analyze Candidate Solution

Snort can be configured further to benefit the network

Requirements Analysis

Project Name: NIDS Installation for Clinical Research Lab
Company: Barker Labs
Division: Computer Network
Department: Information Technology

Prepared by: Michaela Pierce
Project Manager: Michaela Pierce
Project Owner, IT Director: David Smith
Project Sponsor, Business Owner: John Barker

Version Control

0.1 Charter created by Michaela Pierce – 02/10/2022
0.2 Charter edited by David Smith – 02/12/2022, Scope affected/ reduced
0.3 Problem Analysis created by Michaela Pierce – 02/20/2022
0.4 Problem Analysis edited by David Smith – 02/21/2022
0.5 Feasibility Analysis added to Problem Analysis, by Michaela Pierce – 02/26/2022
0.6 Requirements Analysis added by Michaela Pierce – 03/01/2022
0.7 Decision Analysis created by Michaela Pierce – 03/08/2022
0.8 Decision Analysis edited by David Smith – 03/10/2022
0.9 Requirements Analysis created by Michaela Pierce – 03/12/2022

Design

Project Name: NIDS Installation for Clinical Research Lab
Company: Barker Labs
Division: Computer Network
Department: Information Technology

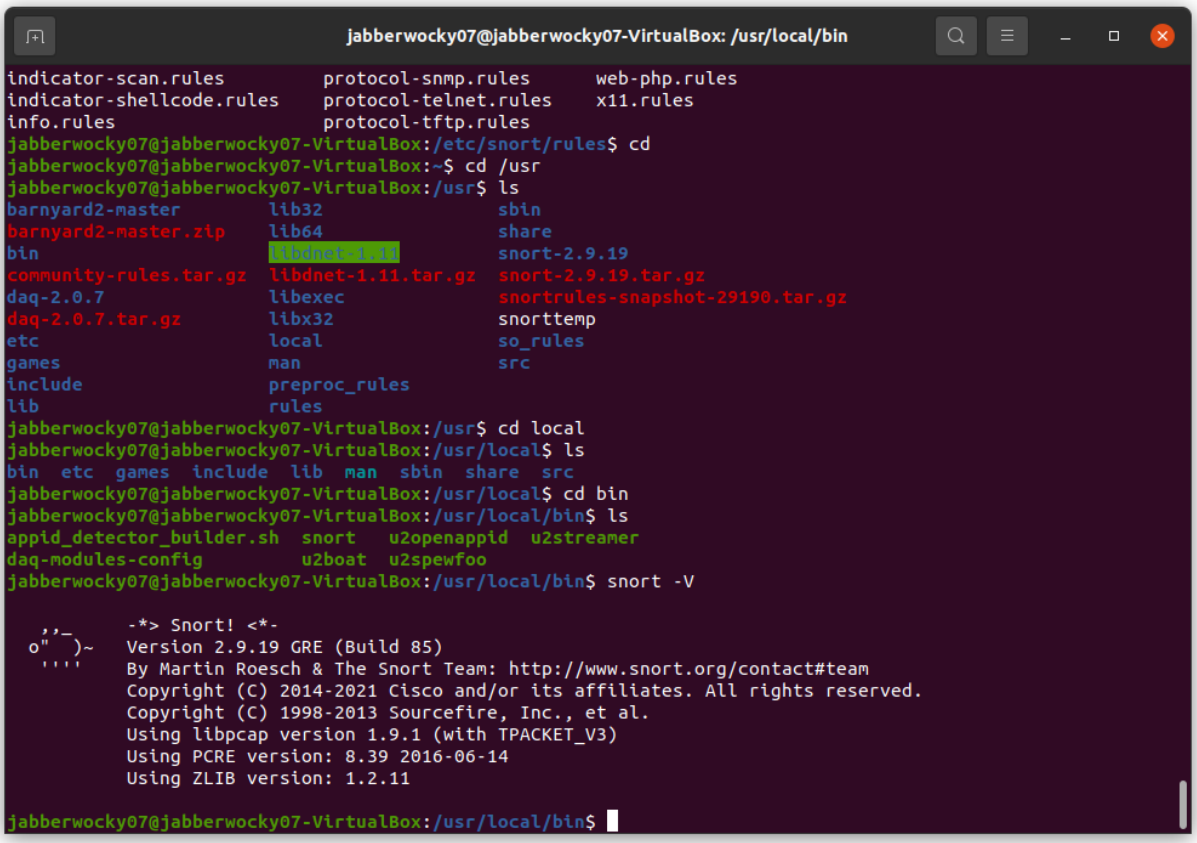
Prepared by: Michaela Pierce
Project Manager: Michaela Pierce
Project Owner, IT Director: David Smith
Project Sponsor, Business Owner: John Barker

Version Control

0.1 Charter created by Michaela Pierce – 02/10/2022
0.2 Charter edited by David Smith – 02/12/2022, Scope affected/ reduced
0.3 Problem Analysis created by Michaela Pierce – 02/20/2022
0.4 Problem Analysis edited by David Smith – 02/21/2022
0.5 Feasibility Analysis added to Problem Analysis, by Michaela Pierce – 02/26/2022
0.6 Requirements Analysis added by Michaela Pierce – 03/01/2022
0.7 Decision Analysis created by Michaela Pierce – 03/08/2022
0.8 Decision Analysis edited by David Smith – 03/10/2022
0.9 Design was added by Michaela Pierce – 03/15/2022

User Interface Design- The new system will use the command line prompt for all commands.

Additional software can be installed at a later time to provide a more interactive user interface.



Construction

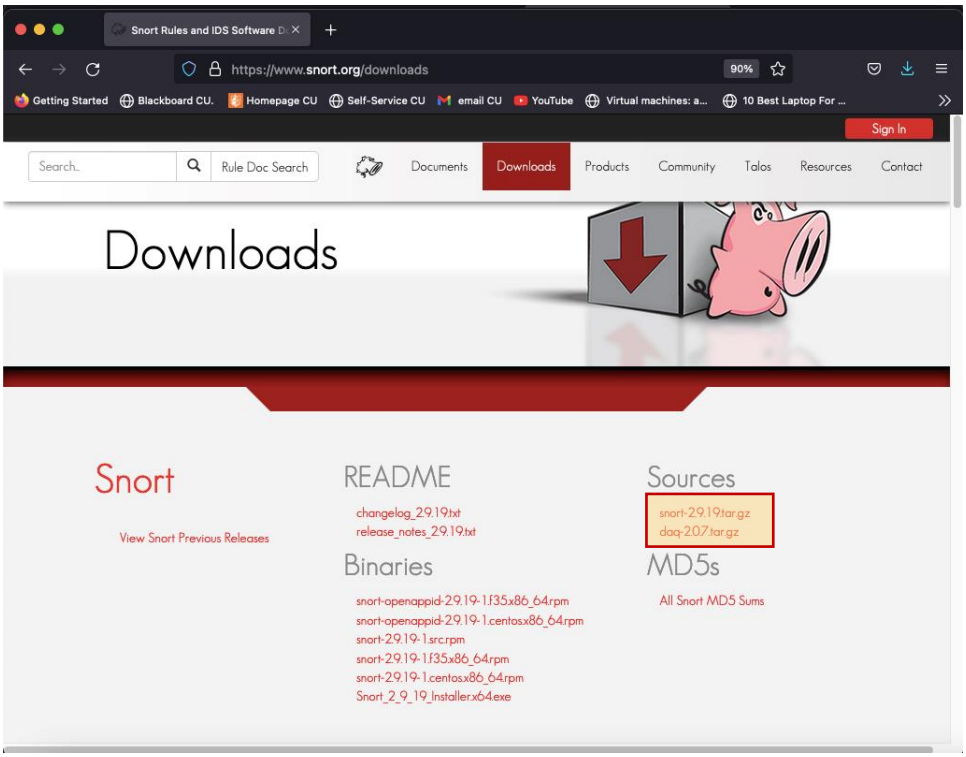
Project Name: NIDS Installation for Clinical Research Lab
Company: Barker Labs
Division: Computer Network
Department: Information Technology

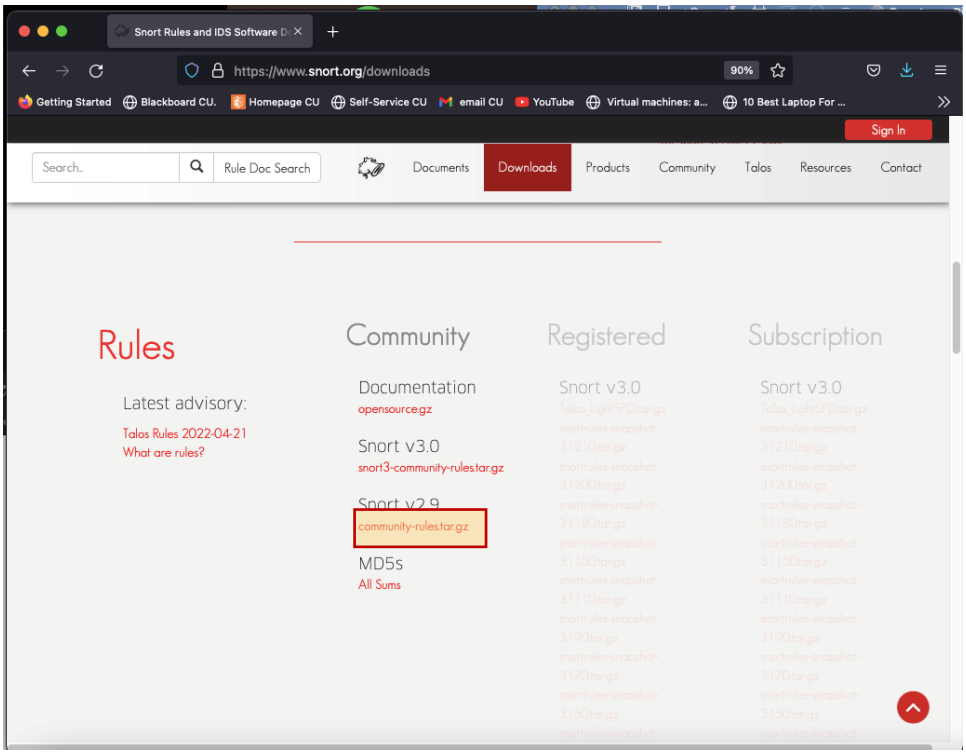
Prepared by: Michaela Pierce
Project Manager: Michaela Pierce
Project Owner, IT Director: David Smith
Project Sponsor, Business Owner: John Barker

Version Control

0.1 Charter created by Michaela Pierce – 02/10/2022
0.2 Charter edited by David Smith – 02/12/2022, Scope affected/ reduced
0.3 Problem Analysis created by Michaela Pierce – 02/20/2022
0.4 Problem Analysis edited by David Smith – 02/21/2022
0.5 Feasibility Analysis added to Problem Analysis, by Michaela Pierce – 02/26/2022
0.6 Requirements Analysis added by Michaela Pierce – 03/01/2022
0.7 Decision Analysis created by Michaela Pierce – 03/08/2022
0.8 Decision Analysis edited by David Smith – 03/10/2022
0.9 Design was added by Michaela Pierce – 03/15/2022
0.10 Construction was added by Michaela Pierce – 03/25/2022

These packages were downloaded to the system to use later.
Go to https://www.snort.org/downloads and download the libdaq, snort, and community-rules tar files





```
sudo apt install build-essential libpcap-dev libpcre3-dev libnet1-dev
zlib1g-dev luajit hwloc libdnet-dev libdumbnet-dev bison flex liblzma-
dev openssl libssl-dev pkg-config libhwloc-dev cmake cpputest
libsqlite3-dev uuid-dev libcmocka-dev libnetfilter-queue-dev libmnl-dev
autotools-dev libluajit-5.1-dev libunwind-dev
```

Create a directory for snort

```
mkdir snort-source-files
cd snort-source-files
```

Install the libdaq package

Locate the dar-2.0.7.tar.gz file under Downloads

```
cd downloads
sudo mv daq-2.0.7.tar.gz  ~/snort-source-files
cd ~/snort-source-files
```

```
tar -xvzf daq-2.0.7.tar.gz
cd daq-2.0.7
autoreconf -f -i
```

```
./configure && make && sudo make install
```

```
sudo mkdir -p /etc/snort/rules
sudo mkdir /var/log/snort
sudo mkdir /usr/local/lib/snort_dynamicrules
```

```
cd downloads
sudo mv snort-2.9.19.tar.gz  ~/snort-source-files
cd ~/snort-source-files
```

```
sudo chmod -r 5775 /etc/snort
sudo chmod -r 5775 /var/log/snort
sudo chmod -r 5775 /usr/local/lib/snort_dynamicrules
sudo chown -r snort:snort /etc/snort
sudo chown -r snort:snort /var/log/snort
sudo chown -r snort:snort /usr/local/lib/snort_dynamicrules
```

```
tar -xvzf snort-2.9.19.tar.gz
cd snort-2.9.19
./configure_cmake.sh
sudo make && sudo make install
```

```
sudo touch /etc/snort/rules/white_list.rules
sudo touch /etc/snort/rules/black_list.rules
sudo touch /etc/snort/local.rules
```

```
sudo ldconfig
```

```
sudo cp ~/snort-source-files/snort-2.9.19/etc/snort.conf
/etc/snort/rules

sudo cp ~/snort-source-files/snort-2.9.19/map /etc/snort/rules
```

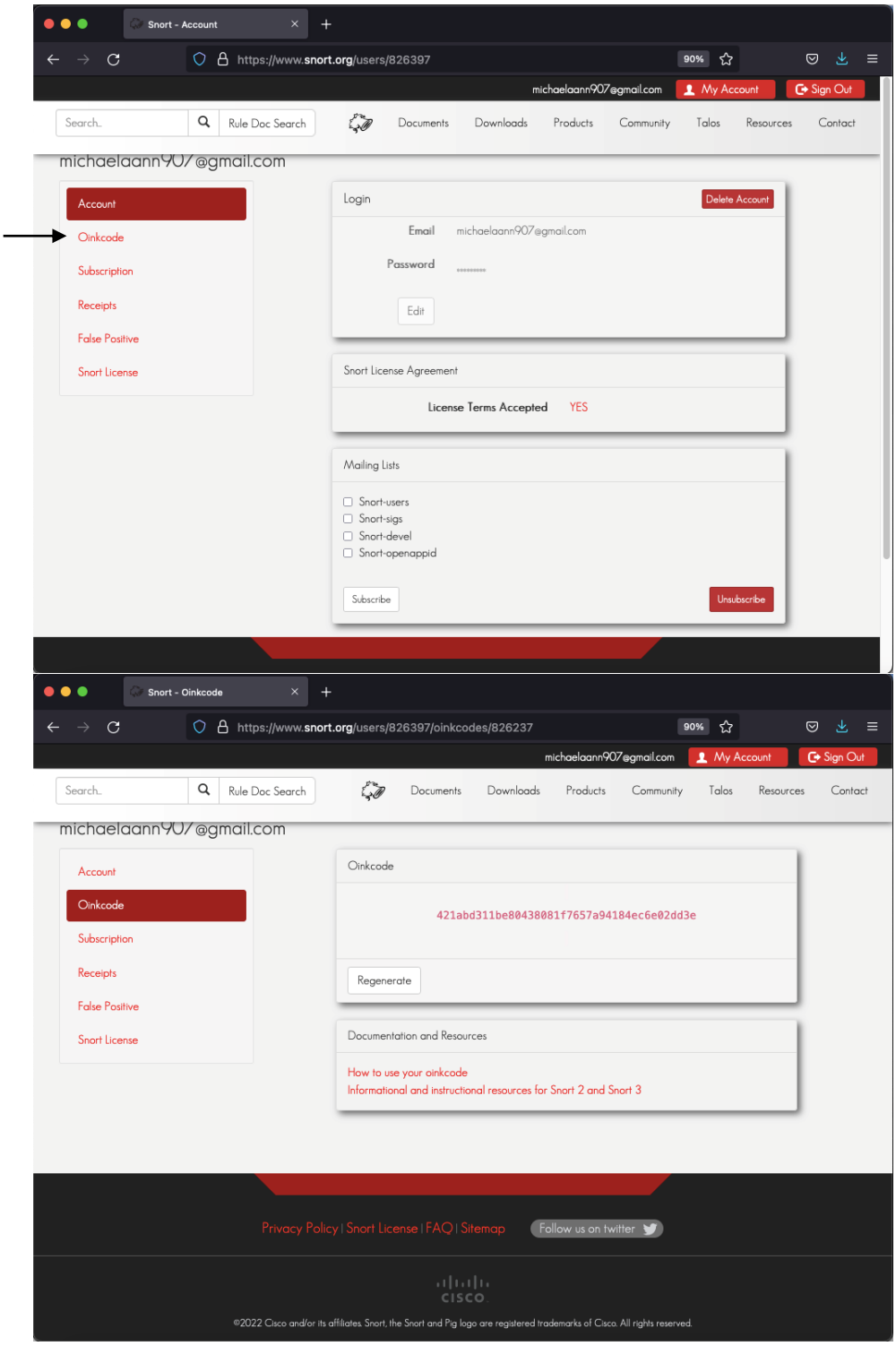
```
sudo ln -s /usr/local/bin/snort usr/bin/snort
```

```
sudo groupadd snort
sudo useradd snort -r -s /sbin/login -c SNORT_IDS -g snort
```

```
cd downloads
sudo mv community_rules.tar.gz
cd ~/snort-source-files
```

```
sudo tar -xvf community-rules.tar.gz -C
cd community-rules
sudo cp ~/community-rules/* /etc/snort/rules
```

Sign up and login in to https://www.snort.org



Download the registered user rules and use oink code from account in command line prompt:

wget https://www.snort.org/rules/snortrules-snapshot-29160.tar.gz?oinkcode=OINKCODE-O ~/registered.tar.gz

```
wget https://www.snort.org/rules/snortrules-snapshot-29160.tar.gz?oinkcode=421abd311be80438081f7657a94184ec6e02dd3e -O ~/registered.tar.gz
```

```
sudo tar -xvf  ~/registered.tar.gz -C /etc/snort
```

```
sudo nano /etc/snort/rules/snort.conf
```

```
sudo nano /etc/snort/rules/snort.conf
```

Snort.Conf file (below)

```
#--------------------------------------------------
#   VRT Rule Packages Snort.conf
#
#   For more information visit us at:
#     http://www.snort.org                Snort Website
#     http://vrt-blog.snort.org/    Sourcefire VRT Blog
#
#     Mailing list Contact:      snort-sigs@lists.sourceforge.net
#     False Positive reports:    fp@sourcefire.com
#     Snort bugs:                bugs@snort.org
#
#     Compatible with Snort Versions:
#     VERSIONS : 2.9.19.0
#
#     Snort build options:
#     OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofiling --enable-zlib --enable-active-response --enable-normalizer --enable-reload --enable-react --enable-flexresp3
#
#     Additional information:
#     This configuration file enables active response, to run snort in
#     test mode -T you are required to supply an interface -i <interface>
#     or test mode will fail to fully validate the configuration and
```

```
#     exit with a FATAL error
#--------------------------------------------------

##################################################
# This file contains a sample snort configuration.
# You should take the following steps to create your own custom configuration:
#
#  1) Set the network variables.
#  2) Configure the decoder
#  3) Configure the base detection engine
#  4) Configure dynamic loaded libraries
#  5) Configure preprocessors
#  6) Configure output plugins
#  7) Customize your rule set
#  8) Customize preprocessor and decoder rule set
#  9) Customize shared object rule set
##################################################

##################################################
# Step #1: Set the network variables.  For more information, see README.variables
##################################################

# Setup the network addresses you are protecting
#ipvar HOME_NET 192.168.1.1/24
ipvar HOME_NET 192.168.1.0/24

# Set up the external network addresses. Leave as "any" in most situations
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

# List of telnet servers on your network
ipvar TELNET_SERVERS $HOME_NET

# List of ssh servers on your network
ipvar SSH_SERVERS $HOME_NET
```

```
# List of ftp servers on your network
ipvar FTP_SERVERS $HOME_NET

# List of sip servers on your network
ipvar SIP_SERVERS $HOME_NET

# List of ports you run web servers on
portvar HTTP_PORTS
[36,80,81,82,83,84,85,86,87,88,89,90,311,323,383,443,555,591,593,623,631,664,801,808,818,9
01,972,1158,1220,1270,1414,1533,1581,1719,1720,1741,1801,1812,1830,1942,2231,2301,2375,
2381,2578,2809,2869,2980,3000,3029,3037,3057,3128,3323,3443,3702,4000,4343,4444,4592,4
848,5000,5054,5060,5061,5117,5222,5250,5416,5443,5450,5480,5555,5600,5814,5894,5984,59
85,5986,6060,6080,6173,6988,7000,7001,7005,7070,7071,7080,7144,7145,7180,7181,7510,777
0,7777,7778,7779,8000,8001,8008,8014,8015,8020,8028,8040,8080,8081,8082,8085,8088,8090,
8095,8118,8123,8161,8180,8181,8182,8222,8243,8280,8300,8333,8344,8393,8400,8484,8500,8
509,8511,8694,8787,8800,8848,8852,8880,8888,8899,8983,9000,9001,9002,9050,9060,9080,90
90,9091,9111,9200,9201,9290,9443,9447,9700,9710,9788,9830,9850,9999,10000,10080,10100,
10250,10255,10297,10443,11371,12601,13014,14592,15489,16000,16992,16993,16994,16995,1
7000,18081,19980,20000,29991,30007,30018,30888,33300,34412,34443,34444,36099,37215,40
007,41080,44449,49152,49153,50000,50002,50452,51423,53331,54444,55252,55555,56712]

# List of ports you want to look for SHELLCODE on.
portvar SHELLCODE_PORTS !80

# List of ports you might see oracle attacks on
portvar ORACLE_PORTS 1024:

# List of ports you want to look for SSH connections on:
portvar SSH_PORTS 22

# List of ports you run ftp servers on
portvar FTP_PORTS [21,2100,3535]

# List of ports you run SIP servers on
portvar SIP_PORTS [5060,5061,5600]

# List of file data ports for file inspection
portvar FILE_DATA_PORTS [$HTTP_PORTS,110,143]

# List of GTP ports for GTP preprocessor
portvar GTP_PORTS [2123,2152,3386]

# other variables, these should not be modified
```

```
ipvar AIM_SERVERS
[64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.
188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.179.0/24,205.188.248.0/24]

# 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
#var WHITE_LIST_PATH ../rules
#var BLACK_LIST_PATH ../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

# Stop Alerts on obsolete TCP options
config disable_tcpopt_obsolete_alerts

# Stop Alerts on T/TCP alerts
config disable_tcpopt_ttcp_alerts

# Stop Alerts on all other TCPOption type events:
config disable_tcpopt_alerts

# Stop Alerts on invalid ip options
config disable_ipopt_alerts

# Alert if value in length field (IP, TCP, UDP) is greater th elength of the packet
# config enable_decode_oversized_alerts

# Same as above, but drop packet if in Inline mode (requires enable_decode_oversized_alerts)
# config enable_decode_oversized_drops

# Configure IP / TCP checksum mode
config checksum_mode: all
```

```
# Configure maximum number of flowbit references.  For more information, see
README.flowbits
# config flowbits_size: 64

# Configure ports to ignore
# config ignore_ports: tcp 21 6667:6671 1356
# config ignore_ports: udp 1:17 53

# Configure active response for non inline operation. For more information, see REAMDE.active
# config response: eth0 attempts 2

# Configure DAQ related options for inline operation. For more information, see README.daq
#
# config daq: <type>
# config daq_dir: <dir>
# config daq_mode: <mode>
# config daq_var: <var>
#
# <type> ::= pcap | afpacket | dump | nfq | ipq | ipfw
# <mode> ::= read-file | passive | inline
# <var> ::= arbitrary <name>=<value passed to DAQ
# <dir> ::= path as to where to look for DAQ module so's

# Configure specific UID and GID to run snort as after dropping privs. For more information see
snort -h command line options
#
# config set_gid:
# config set_uid:

# Configure default snaplen. Snort defaults to MTU of in use interface. For more information see
README
#
# config snaplen:
#

# Configure default bpf_file to use for filtering what traffic reaches snort. For more information
see snort -h command line options (-F)
#
# config bpf_file:
#

# Configure default log directory for snort to log to.  For more information see snort -h command
line options (-l)
#
# config logdir:
```

```
#/var/log/snort

##################################################
# Step #3: Configure the base detection engine.  For more information, see  README.decode
##################################################

# Configure PCRE match limitations
config pcre_match_limit: 3500
config pcre_match_limit_recursion: 1500

# Configure the detection engine  See the Snort Manual, Configuring Snort - Includes - Config
config detection: search-method ac-split search-optimize max-pattern-len 20

# Configure the event queue.  For more information, see README.event_queue
config event_queue: max_queue 15 log 15 order_events content_length

##################################################
## Configure GTP if it is to be used.
## For more information, see README.GTP
##################################################

# config enable_gtp

##################################################
# Per packet and rule latency enforcement
# For more information see README.ppm
##################################################

# Per Packet latency configuration
#config ppm: max-pkt-time 250, \
#   fastpath-expensive-packets, \
#   pkt-log

# Per Rule latency configuration
#config ppm: max-rule-time 200, \
#   threshold 3, \
#   suspend-expensive-rules, \
#   suspend-timeout 20, \
#   rule-log alert

##################################################
# Configure Perf Profiling for debugging
# For more information see README.PerfProfiling
##################################################

#config profile_rules: print all, sort avg_ticks
```

```
#config profile_preprocs: print all, sort avg_ticks

##################################################
# Configure protocol aware flushing
# For more information see README.stream5
##################################################
config paf_max: 16000

##################################################
# Step #4: Configure dynamic loaded libraries.
# For more information, see Snort Manual, Configuring Snort - Dynamic Modules
##################################################

# path to dynamic preprocessor libraries
dynamicpreprocessor directory /usr/local/lib/snort_dynamicpreprocessor/

# path to base preprocessor engine
dynamicengine /usr/local/lib/snort_dynamicengine/libsf_engine.so

# path to dynamic rules libraries (Shared Object (SO) Rules)
# Set this path to where the compiled *.so binaries are installed
#dynamicdetection directory /usr/local/lib/snort_dynamicrules
dynamicdetection directory /usr/local/lib/snort_dynamicrules
##################################################
# Step #5: Configure preprocessors
# For more information, see the Snort Manual, Configuring Snort - Preprocessors
##################################################

# GTP Control Channle Preprocessor. For more information, see README.GTP
# preprocessor gtp: ports { 2123 3386 2152 }

# Inline packet normalization. For more information, see README.normalize
# Does nothing in IDS mode
preprocessor normalize_ip4
preprocessor normalize_tcp: block, rsv, pad, urp, req_urg, req_pay, req_urp, ips, ecn stream
#preprocessor normalize_icmp4
preprocessor normalize_ip6
#preprocessor normalize_icmp6

# Target-based IP defragmentation.  For more inforation, see README.frag3
preprocessor frag3_global: max_frags 65536
preprocessor frag3_engine: policy windows detect_anomalies overlap_limit 10
min_fragment_length 100 timeout 180

# Target-Based stateful inspection/stream reassembly.  For more inforation, see
README.stream5
```

```
preprocessor stream5_global: track_tcp yes, \
   track_udp yes, \
   track_icmp no, \
   max_tcp 262144, \
   max_udp 131072, \
   max_active_responses 2, \
   min_response_seconds 5
preprocessor stream5_tcp: policy windows, detect_anomalies, require_3whs 180, \
   overlap_limit 10, small_segments 3 bytes 150, timeout 180, \
    ports client 21 22 23 25 42 53 70 79 109 110 111 113 119 135 136 137 139 143 161 445 513
514 587 593 691 1433 1521 1741 2100 3306 6070 6665 6666 6667 6668 6669 7000 8181 32770
32771 32772 32773 32774 32775 32776 32777 32778 32779, \
    ports both 36 80 81 82 83 84 85 86 87 88 89 90 110 311 323 383 443 465 555 563 591 593
623 631 636 664 801 808 818 901 972 989 992 993 994 995 1158 1220 1270 1414 1533 1581
1719 1720 1741 1801 1812 1830 1942 2231 2301 2375 2381 2578 2809 2869 2980 3000 3001
3029 3037 3057 3128 3300 3323 3443 3702 3901 4000 4343 4444 4592 4848 5000 5054 5060
5061 5117 5222 5250 5416 5443 5450 5480 5555 5600 5814 5894 5984 5985 5986 6060 6080
6173 6988 7000 7001 7005 7070 7071 7080 7144 7145 7180 7181 7510 7770 7777 7778 7779
7801 7802 7900 7901 7902 7903 7904 7905 7906 7907 7908 7909 7910 7911 7912 7913 7914
7915 7916 7917 7918 7919 7920 8000 8001 8008 8014 8015 8020 8028 8040 8080 8081 8082
8085 8088 8090 8095 8118 8123 8161 8180 8181 8182 8222 8243 8280 8300 8333 8344 8393
8400 8484 8500 8509 8511 8694 8787 8800 8848 8852 8880 8888 8899 8983 9000 9001 9002
9050 9060 9080 9090 9091 9111 9200 9201 9290 9443 9447 9700 9710 9788 9830 9850 9999
10000 10080 10100 10250 10255 10297 10443 11371 12601 13014 14592 15489 15672 16000
16992 16993 16994 16995 17000 18081 19980 20000 29991 30007 30018 30888 33300 34412
34443 34444 36099 37215 40007 41080 44449 49152 49153 50000 50002 50452 51423 53331
54444 55252 55555 56712
preprocessor stream5_udp: timeout 180

# performance statistics.  For more information, see the Snort Manual, Configuring Snort -
Preprocessors - Performance Monitor
# preprocessor perfmonitor: time 300 file /var/snort/snort.stats pktcnt 10000

# HTTP normalization and anomaly detection.  For more information, see
README.http_inspect
preprocessor http_inspect: global iis_unicode_map unicode.map 1252 compress_depth 65535
decompress_depth 65535
preprocessor http_inspect_server: server default \
   http_methods { GET POST PUT SEARCH MKCOL COPY MOVE LOCK UNLOCK
NOTIFY POLL BCOPY BDELETE BMOVE LINK UNLINK OPTIONS HEAD DELETE
TRACE TRACK CONNECT SOURCE SUBSCRIBE UNSUBSCRIBE PROPFIND
PROPPATCH BPROPFIND BPROPPATCH RPC_CONNECT PROXY_SUCCESS
BITS_POST CCM_POST SMS_POST RPC_IN_DATA RPC_OUT_DATA
RPC_ECHO_DATA } \
   chunk_length 500000 \
   server_flow_depth 0 \
```

```
    client_flow_depth 0 \
    post_depth 65495 \
    oversize_dir_length 500 \
    max_header_length 750 \
    max_headers 100 \
    max_spaces 200 \
    small_chunk_length { 10 5 } \
    ports { 36 80 81 82 83 84 85 86 87 88 89 90 311 323 383 443 555 591 593 623 631 664 801
808 818 901 972 1158 1220 1270 1414 1533 1581 1719 1720 1741 1801 1812 1830 1942 2231
2301 2375 2381 2578 2809 2869 2980 3000 3029 3037 3057 3128 3323 3443 3702 4000 4343
4444 4592 4848 5000 5054 5060 5061 5117 5222 5250 5416 5443 5450 5480 5555 5600 5814
5894 5984 5985 5986 6060 6080 6173 6988 7000 7001 7005 7070 7071 7080 7144 7145 7180
7181 7510 7770 7777 7778 7779 8000 8001 8008 8014 8015 8020 8028 8040 8080 8081 8082
8085 8088 8090 8095 8118 8123 8161 8180 8181 8182 8222 8243 8280 8300 8333 8344 8393
8400 8484 8500 8509 8511 8694 8787 8800 8848 8852 8880 8888 8899 8983 9000 9001 9002
9050 9060 9080 9090 9091 9111 9200 9201 9290 9443 9447 9700 9710 9788 9830 9850 9999
10000 10080 10100 10250 10255 10297 10443 11371 12601 13014 14592 15489 15672 16000
16992 16993 16994 16995 17000 18081 19980 20000 29991 30007 30018 30888 33300 34412
34443 34444 36099 37215 40007 41080 44449 49152 49153 50000 50002 50452 51423 53331
54444 55252 55555 56712 } \
    non_rfc_char { 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 } \
    enable_cookie \
    extended_response_inspection \
    inspect_gzip \
    normalize_utf \
    unlimited_decompress \
    normalize_javascript \
    apache_whitespace no \
    ascii no \
    bare_byte no \
    directory no \
    double_decode no \
    iis_backslash no \
    iis_delimiter no \
    iis_unicode no \
    multi_slash no \
    utf_8 no \
    u_encode yes \
    webroot no \
#    decompress_swf { deflate lzma } \
    decompress_swf { deflate }  \
    decompress_pdf { deflate }


# ONC-RPC normalization and anomaly detection.  For more information, see the Snort Manual,
Configuring Snort - Preprocessors - RPC Decode
```

```
preprocessor rpc_decode: 111 32770 32771 32772 32773 32774 32775 32776 32777 32778
32779 no_alert_multiple_requests no_alert_large_fragments no_alert_incomplete

# Back Orifice detection.
preprocessor bo

# FTP / Telnet normalization and anomaly detection.  For more information, see
README.ftptelnet
preprocessor ftp_telnet: global inspection_type stateful encrypted_traffic no check_encrypted
preprocessor ftp_telnet_protocol: telnet \
   ayt_attack_thresh 20 \
   normalize ports { 23 } \
   detect_anomalies
preprocessor ftp_telnet_protocol: ftp server default \
   def_max_param_len 100 \
   ports { 21 2100 3535 } \
   telnet_cmds yes \
   ignore_telnet_erase_cmds yes \
   ftp_cmds { ABOR ACCT ADAT ALLO APPE AUTH CCC CDUP } \
   ftp_cmds { CEL CLNT CMD CONF CWD DELE ENC EPRT } \
   ftp_cmds { EPSV ESTA ESTP FEAT HELP LANG LIST LPRT } \
   ftp_cmds { LPSV MACB MAIL MDTM MIC MKD MLSD MLST } \
   ftp_cmds { MODE NLST NOOP OPTS PASS PASV PBSZ PORT } \
   ftp_cmds { PROT PWD QUIT REIN REST RETR RMD RNFR } \
   ftp_cmds { RNTO SDUP SITE SIZE SMNT STAT STOR STOU } \
   ftp_cmds { STRU SYST TEST TYPE USER XCUP XCRC XCWD } \
   ftp_cmds { XMAS XMD5 XMKD XPWD XRCP XRMD XRSQ XSEM } \
   ftp_cmds { XSEN XSHA1 XSHA256 } \
   alt_max_param_len 0 { ABOR CCC CDUP ESTA FEAT LPSV NOOP PASV PWD QUIT
REIN STOU SYST XCUP XPWD } \
   alt_max_param_len 200 { ALLO APPE CMD HELP NLST RETR RNFR STOR STOU
XMKD } \
   alt_max_param_len 256 { CWD RNTO } \
   alt_max_param_len 400 { PORT } \
   alt_max_param_len 512 { SIZE } \
   chk_str_fmt { ACCT ADAT ALLO APPE AUTH CEL CLNT CMD } \
   chk_str_fmt { CONF CWD DELE ENC EPRT EPSV ESTP HELP } \
   chk_str_fmt { LANG LIST LPRT MACB MAIL MDTM MIC MKD } \
   chk_str_fmt { MLSD MLST MODE NLST OPTS PASS PBSZ PORT } \
   chk_str_fmt { PROT REST RETR RMD RNFR RNTO SDUP SITE } \
   chk_str_fmt { SIZE SMNT STAT STOR STRU TEST TYPE USER } \
   chk_str_fmt { XCRC XCWD XMAS XMD5 XMKD XRCP XRMD XRSQ } \
   chk_str_fmt { XSEM XSEN XSHA1 XSHA256 } \
   cmd_validity ALLO < int [ char R int ] > \
   cmd_validity EPSV < [ { char 12 | char A char L char L } ] > \
   cmd_validity MACB < string > \
```

```
    cmd_validity MDTM < [ date nnnnnnnnnnnnnn[.n[n[n]]] ] string > \
    cmd_validity MODE < char ASBCZ > \
    cmd_validity PORT < host_port > \
    cmd_validity PROT < char CSEP > \
    cmd_validity STRU < char FRPO [ string ] > \
    cmd_validity TYPE < { char AE [ char NTC ] | char I | char L [ number ] } >
preprocessor ftp_telnet_protocol: ftp client default \
    max_resp_len 256 \
    bounce yes \
    ignore_telnet_erase_cmds yes \
    telnet_cmds yes


# SMTP normalization and anomaly detection.  For more information, see README.SMTP
preprocessor smtp: ports { 25 465 587 691 } \
    inspection_type stateful \
    b64_decode_depth 0 \
    qp_decode_depth 0 \
    bitenc_decode_depth 0 \
    uu_decode_depth 0 \
    log_mailfrom \
    log_rcptto \
    log_filename \
    log_email_hdrs \
    normalize cmds \
    normalize_cmds { ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO EMAL ESAM
ESND ESOM ETRN EVFY } \
    normalize_cmds { EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT
RSET SAML SEND SOML } \
    normalize_cmds { STARTTLS TICK TIME TURN TURNME VERB VRFY X-ADAT X-
DRCP X-ERCP X-EXCH50 } \
    normalize_cmds { X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN
XLICENSE XQUE XSTA XTRN XUSR } \
    max_command_line_len 512 \
    max_header_line_len 1000 \
    max_response_line_len 512 \
    alt_max_command_line_len 260 { MAIL } \
    alt_max_command_line_len 300 { RCPT } \
    alt_max_command_line_len 500 { HELP HELO ETRN EHLO } \
    alt_max_command_line_len 255 { EXPN VRFY ATRN SIZE BDAT DEBUG EMAL ESAM
ESND ESOM EVFY IDENT NOOP RSET } \
    alt_max_command_line_len 246 { SEND SAML SOML AUTH TURN ETRN DATA RSET
QUIT ONEX QUEU STARTTLS TICK TIME TURNME VERB X-EXPS X-LINK2STATE
XADR XAUTH XCIR XEXCH50 XGEN XLICENSE XQUE XSTA XTRN XUSR } \
    valid_cmds { ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO EMAL ESAM
ESND ESOM ETRN EVFY } \
```

```
   valid_cmds { EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT RSET
SAML SEND SOML } \
   valid_cmds { STARTTLS TICK TIME TURN TURNME VERB VRFY X-ADAT X-DRCP
X-ERCP X-EXCH50 } \
   valid_cmds { X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN
XLICENSE XQUE XSTA XTRN XUSR } \
   xlink2state { enabled }

# Portscan detection.  For more information, see README.sfportscan
# preprocessor sfportscan: proto  { all } memcap { 10000000 } sense_level { low }

# ARP spoof detection.  For more information, see the Snort Manual - Configuring Snort -
Preprocessors - ARP Spoof Preprocessor
# preprocessor arpspoof
# preprocessor arpspoof_detect_host: 192.168.40.1 f0:0f:00:f0:0f:00

# SSH anomaly detection.  For more information, see README.ssh
preprocessor ssh: server_ports { 22 } \
                autodetect \
                max_client_bytes 19600 \
                max_encrypted_packets 20 \
                max_server_version_len 100 \
                enable_respoverflow enable_ssh1crc32 \
                enable_srvoverflow enable_protomismatch

# SMB / DCE-RPC normalization and anomaly detection.  For more information, see
README.dcerpc2
preprocessor dcerpc2: memcap 102400, events [co ]
preprocessor dcerpc2_server: default, policy WinXP, \
   detect [smb [139,445], tcp 135, udp 135, rpc-over-http-server 593], \
   autodetect [tcp 1025:, udp 1025:, rpc-over-http-server 1025:], \
   smb_max_chain 3, smb_invalid_shares ["C$", "D$", "ADMIN$"]

# DNS anomaly detection.  For more information, see README.dns
preprocessor dns: ports { 53 } enable_rdata_overflow

# SSL anomaly detection and traffic bypass.  For more information, see README.ssl
preprocessor ssl: ports { 443 465 563 636 989 992 993 994 995 5061 7801 7802 7900 7901 7902
7903 7904 7905 7906 7907 7908 7909 7910 7911 7912 7913 7914 7915 7916 7917 7918 7919
7920 }, trustservers, noinspect_encrypted

# SDF sensitive data preprocessor.  For more information see README.sensitive_data
preprocessor sensitive_data: alert_threshold 25

# SIP Session Initiation Protocol preprocessor.  For more information see README.sip
preprocessor sip: max_sessions 40000, \
```

```
  ports { 5060 5061 5600 }, \
  methods { invite \
        cancel \
        ack \
        bye \
        register \
        options \
        refer \
        subscribe \
        update \
        join \
        info \
        message \
        notify \
        benotify \
        do \
        qauth \
        sprack \
        publish \
        service \
        unsubscribe \
        prack }, \
  max_uri_len 512, \
  max_call_id_len 80, \
  max_requestName_len 20, \
  max_from_len 256, \
  max_to_len 256, \
  max_via_len 1024, \
  max_contact_len 512, \
  max_content_len 2048

# IMAP preprocessor.  For more information see README.imap
preprocessor imap: \
  ports { 143 } \
  b64_decode_depth 0 \
  qp_decode_depth 0 \
  bitenc_decode_depth 0 \
  uu_decode_depth 0

# POP preprocessor. For more information see README.pop
preprocessor pop: \
  ports { 110 } \
  b64_decode_depth 0 \
  qp_decode_depth 0 \
  bitenc_decode_depth 0 \
  uu_decode_depth 0
```

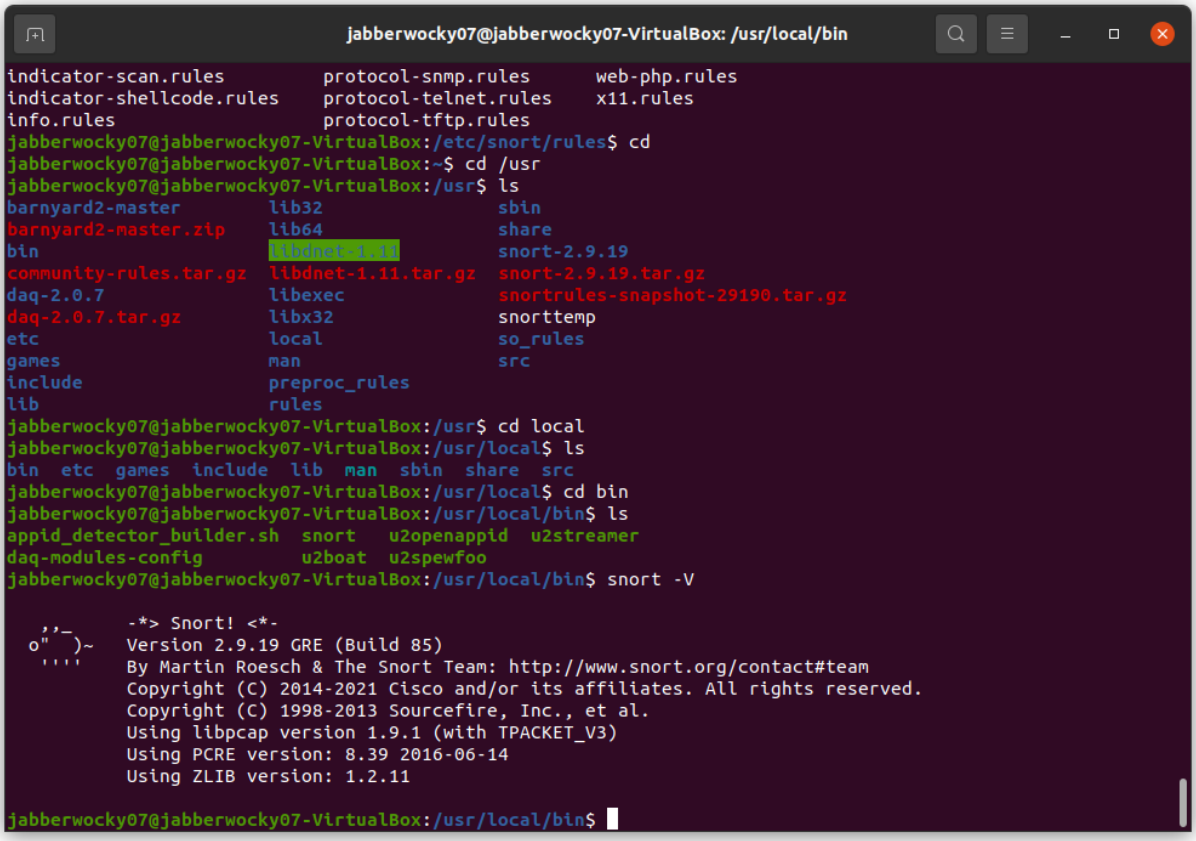
```
# Modbus preprocessor. For more information see README.modbus
preprocessor modbus: ports { 502 }

# DNP3 preprocessor. For more information see README.dnp3
preprocessor dnp3: ports { 20000 } \
  memcap 262144 \
  check_crc

# Reputation preprocessor. For more information see README.reputation
#preprocessor reputation: \
#   memcap 500, \
#   priority whitelist, \
#   nested_ip inner, \
#   whitelist $WHITE_LIST_PATH/white_list.rules, \
#   blacklist $BLACK_LIST_PATH/black_list.rules

##################################################
# Step #6: Configure output plugins
# For more information, see Snort Manual, Configuring Snort - Output Modules
##################################################

# unified2
# Recommended for most installs
# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
#####output unified2: filename merged.log, limit 128
# Additional configuration for specific types of installs
# output alert_unified2: filename snort.alert, limit 128, nostamp
# output log_unified2: filename snort.log, limit 128, nostamp

# syslog
# output alert_syslog: LOG_AUTH LOG_ALERT

# pcap
# output log_tcpdump: tcpdump.log

# metadata reference data.  do not modify these lines
include classification.config
include reference.config


##################################################
# Step #7: Customize your rule set
# For more information, see Snort Manual, Writing Snort Rules
#
# NOTE: All categories are enabled in this conf file
```

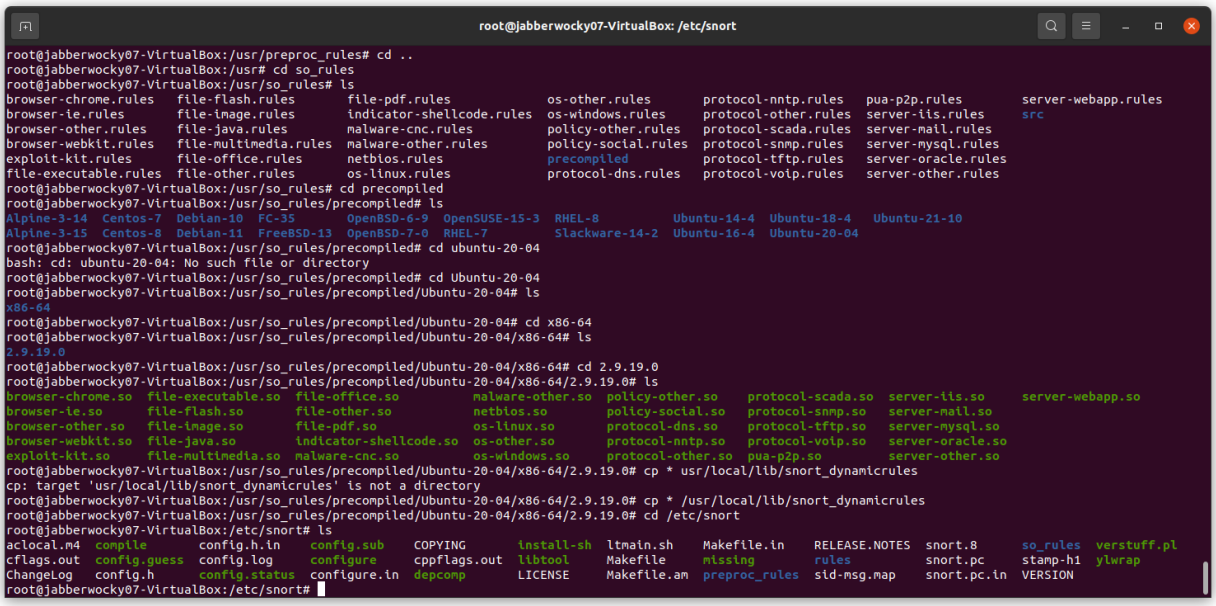
```
##################################################

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


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
include $RULE_PATH/exploit-kit.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/file-executable.rules
include $RULE_PATH/file-flash.rules
include $RULE_PATH/file-identify.rules
include $RULE_PATH/file-image.rules
include $RULE_PATH/file-java.rules
include $RULE_PATH/file-multimedia.rules
include $RULE_PATH/file-office.rules
include $RULE_PATH/file-other.rules
include $RULE_PATH/file-pdf.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/icmp-info.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/imap.rules
include $RULE_PATH/indicator-compromise.rules
include $RULE_PATH/indicator-obfuscation.rules
include $RULE_PATH/indicator-scan.rules
include $RULE_PATH/indicator-shellcode.rules
include $RULE_PATH/info.rules
```

include $RULE_PATH/malware-backdoor.rules
include $RULE_PATH/malware-cnc.rules
include $RULE_PATH/malware-other.rules
include $RULE_PATH/malware-tools.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/multimedia.rules
include $RULE_PATH/mysql.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/nntp.rules
include $RULE_PATH/oracle.rules
include $RULE_PATH/os-linux.rules
include $RULE_PATH/os-mobile.rules
include $RULE_PATH/os-other.rules
include $RULE_PATH/os-solaris.rules
include $RULE_PATH/os-windows.rules
include $RULE_PATH/other-ids.rules
include $RULE_PATH/p2p.rules
include $RULE_PATH/phishing-spam.rules
include $RULE_PATH/policy-multimedia.rules
include $RULE_PATH/policy-other.rules
include $RULE_PATH/policy.rules
include $RULE_PATH/policy-social.rules
include $RULE_PATH/policy-spam.rules
include $RULE_PATH/pop2.rules
include $RULE_PATH/pop3.rules
include $RULE_PATH/protocol-dns.rules
include $RULE_PATH/protocol-finger.rules
include $RULE_PATH/protocol-ftp.rules
include $RULE_PATH/protocol-icmp.rules
include $RULE_PATH/protocol-imap.rules
include $RULE_PATH/protocol-nntp.rules
include $RULE_PATH/protocol-other.rules
include $RULE_PATH/protocol-pop.rules
include $RULE_PATH/protocol-rpc.rules
include $RULE_PATH/protocol-scada.rules
include $RULE_PATH/protocol-services.rules
include $RULE_PATH/protocol-snmp.rules
include $RULE_PATH/protocol-telnet.rules
include $RULE_PATH/protocol-tftp.rules
include $RULE_PATH/protocol-voip.rules
include $RULE_PATH/pua-adware.rules
include $RULE_PATH/pua-other.rules
include $RULE_PATH/pua-p2p.rules
include $RULE_PATH/pua-toolbars.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules

```
include $RULE_PATH/scada.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/server-apache.rules
include $RULE_PATH/server-iis.rules
include $RULE_PATH/server-mail.rules
include $RULE_PATH/server-mssql.rules
include $RULE_PATH/server-mysql.rules
include $RULE_PATH/server-oracle.rules
include $RULE_PATH/server-other.rules
include $RULE_PATH/server-samba.rules
include $RULE_PATH/server-webapp.rules
include $RULE_PATH/shellcode.rules
include $RULE_PATH/smtp.rules
include $RULE_PATH/snmp.rules
include $RULE_PATH/specific-threats.rules
include $RULE_PATH/spyware-put.rules
include $RULE_PATH/sql.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/tftp.rules
include $RULE_PATH/virus.rules
include $RULE_PATH/voip.rules
include $RULE_PATH/web-activex.rules
include $RULE_PATH/web-attacks.rules
include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-client.rules
include $RULE_PATH/web-coldfusion.rules
include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-iis.rules
include $RULE_PATH/web-misc.rules
include $RULE_PATH/web-php.rules
include $RULE_PATH/x11.rules

##################################################
# Step #8: Customize your preprocessor and decoder alerts
# For more information, see README.decoder_preproc_rules
##################################################

# decoder and preprocessor event rules
include $PREPROC_RULE_PATH/preprocessor.rules
include $PREPROC_RULE_PATH/decoder.rules
include $PREPROC_RULE_PATH/sensitive-data.rules

##################################################
# Step #9: Customize your Shared Object Snort Rules
# For more information, see http://vrt-blog.snort.org/2009/01/using-vrt-certified-shared-object-
rules.html
```

```
##################################################

# dynamic library rules (Shared Object (SO) Rules)
# These includes point to the SO stub rules files.  For the detections to work, you
#    must also point "dynamicdetection directory" option above to point to where the
#    compiled SO Rule *.so binaries are located
# include $SO_RULE_PATH/browser-chrome.rules
# include $SO_RULE_PATH/browser-ie.rules
# include $SO_RULE_PATH/browser-other.rules
# include $SO_RULE_PATH/exploit-kit.rules
# include $SO_RULE_PATH/file-executable.rules
# include $SO_RULE_PATH/file-flash.rules
# include $SO_RULE_PATH/file-image.rules
# include $SO_RULE_PATH/file-java.rules
# include $SO_RULE_PATH/file-multimedia.rules
# include $SO_RULE_PATH/file-office.rules
# include $SO_RULE_PATH/file-other.rules
# include $SO_RULE_PATH/file-pdf.rules
# include $SO_RULE_PATH/indicator-shellcode.rules
# include $SO_RULE_PATH/malware-cnc.rules
# include $SO_RULE_PATH/malware-other.rules
# include $SO_RULE_PATH/netbios.rules
# include $SO_RULE_PATH/os-linux.rules
# include $SO_RULE_PATH/os-other.rules
# include $SO_RULE_PATH/os-windows.rules
# include $SO_RULE_PATH/policy-other.rules
# include $SO_RULE_PATH/policy-social.rules
# include $SO_RULE_PATH/protocol-dns.rules
# include $SO_RULE_PATH/protocol-nntp.rules
# include $SO_RULE_PATH/protocol-other.rules
# include $SO_RULE_PATH/protocol-scada.rules
# include $SO_RULE_PATH/protocol-snmp.rules
# include $SO_RULE_PATH/protocol-tftp.rules
# include $SO_RULE_PATH/protocol-voip.rules
# include $SO_RULE_PATH/pua-p2p.rules
# include $SO_RULE_PATH/server-apache.rules
# include $SO_RULE_PATH/server-iis.rules
# include $SO_RULE_PATH/server-mail.rules
# include $SO_RULE_PATH/server-mysql.rules
# include $SO_RULE_PATH/server-oracle.rules
# include $SO_RULE_PATH/server-other.rules
# include $SO_RULE_PATH/server-webapp.rules

# Event thresholding or suppression commands. See threshold.conf
include threshold.conf
Installed Snort Version
```


jabberwocky07@jabberwocky07-VirtualBox: /usr/local/bin
indicator-scan.rules protocol-snmp.rules web-php.rules
indicator-shellcode.rules protocol-telnet.rules x11.rules
info.rules protocol-tftp.rules
jabberwocky07@jabberwocky07-VirtualBox:/etc/snort/rules$ cd
jabberwocky07@jabberwocky07-VirtualBox:~$ cd /usr
jabberwocky07@jabberwocky07-VirtualBox:/usr$ ls
barnyard2-master lib32 sbin
barnyard2-master.zip lib64 share
bin libdnet-1.11 snort-2.9.19
community-rules.tar.gz libdnet-1.11.tar.gz snort-2.9.19.tar.gz
daq-2.0.7 libexec snortrules-snapshot-29190.tar.gz
daq-2.0.7.tar.gz libx32 snorttemp
etc local so_rules
games man src
include preproc_rules
lib rules
jabberwocky07@jabberwocky07-VirtualBox:/usr$ cd local
jabberwocky07@jabberwocky07-VirtualBox:/usr/local$ ls
bin etc games include lib man sbin share src
jabberwocky07@jabberwocky07-VirtualBox:/usr/local$ cd bin
jabberwocky07@jabberwocky07-VirtualBox:/usr/local/bin$ ls
appid_detector_builder.sh snort u2openappid u2streamer
daq-modules-config u2boat u2spewfoo
jabberwocky07@jabberwocky07-VirtualBox:/usr/local/bin$ snort -V
-*> Snort! <*-
Version 2.9.19 GRE (Build 85)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2021 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.9.1 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.11
jabberwocky07@jabberwocky07-VirtualBox:/usr/local/bin$


root@jabberwocky07-VirtualBox: /etc/snort
root@jabberwocky07-VirtualBox:/usr/preproc_rules# cd ..
root@jabberwocky07-VirtualBox:/usr# cd so_rules
root@jabberwocky07-VirtualBox:/usr/so_rules# ls
browser-chrome.rules file-flash.rules file-pdf.rules os-other.rules protocol-nntp.rules pua-p2p.rules server-webapp.rules
browser-ie.rules file-image.rules indicator-shellcode.rules os-windows.rules protocol-other.rules server-iis.rules src
browser-other.rules file-java.rules malware-cnc.rules policy-other.rules protocol-scada.rules server-mail.rules
browser-webkit.rules file-multimedia.rules malware-other.rules policy-social.rules protocol-snmp.rules server-mysql.rules
exploit-kit.rules file-office.rules netbios.rules precompiled protocol-tftp.rules server-oracle.rules
file-executable.rules file-other.rules os-linux.rules protocol-dns.rules protocol-voip.rules server-other.rules
root@jabberwocky07-VirtualBox:/usr/so_rules# cd precompiled
root@jabberwocky07-VirtualBox:/usr/so_rules/precompiled# ls
Alpine-3-14 Centos-7 Debian-10 FC-35 OpenBSD-6-9 OpenSUSE-15-3 RHEL-8 Ubuntu-14-4 Ubuntu-18-4 Ubuntu-21-10
Alpine-3-15 Centos-8 Debian-11 FreeBSD-13 OpenBSD-7-0 RHEL-7 Slackware-14-2 Ubuntu-16-4 Ubuntu-20-04
root@jabberwocky07-VirtualBox:/usr/so_rules/precompiled# cd ubuntu-20-04
bash: cd: ubuntu-20-04: No such file or directory
root@jabberwocky07-VirtualBox:/usr/so_rules/precompiled# cd Ubuntu-20-04
root@jabberwocky07-VirtualBox:/usr/so_rules/precompiled/Ubuntu-20-04# ls
x86-64
root@jabberwocky07-VirtualBox:/usr/so_rules/precompiled/Ubuntu-20-04# cd x86-64
root@jabberwocky07-VirtualBox:/usr/so_rules/precompiled/Ubuntu-20-04/x86-64# ls
2.9.19.0
root@jabberwocky07-VirtualBox:/usr/so_rules/precompiled/Ubuntu-20-04/x86-64# cd 2.9.19.0
root@jabberwocky07-VirtualBox:/usr/so_rules/precompiled/Ubuntu-20-04/x86-64/2.9.19.0# ls
browser-chrome.so file-executable.so file-office.so malware-other.so policy-other.so protocol-scada.so server-iis.so server-webapp.so
browser-ie.so file-flash.so file-other.so netbios.so policy-social.so protocol-snmp.so server-mail.so
browser-other.so file-image.so file-pdf.so os-linux.so protocol-dns.so protocol-tftp.so server-mysql.so
browser-webkit.so file-java.so indicator-shellcode.so os-other.so protocol-nntp.so protocol-voip.so server-oracle.so
exploit-kit.so file-multimedia.so malware-cnc.so os-windows.so protocol-other.so pua-p2p.so server-other.so
root@jabberwocky07-VirtualBox:/usr/so_rules/precompiled/Ubuntu-20-04/x86-64/2.9.19.0# cp * usr/local/lib/snort_dynamicrules
cp: target 'usr/local/lib/snort_dynamicrules' is not a directory
root@jabberwocky07-VirtualBox:/usr/so_rules/precompiled/Ubuntu-20-04/x86-64/2.9.19.0# cp * /usr/local/lib/snort_dynamicrules
root@jabberwocky07-VirtualBox:/usr/so_rules/precompiled/Ubuntu-20-04/x86-64/2.9.19.0# cd /etc/snort
root@jabberwocky07-VirtualBox:/etc/snort# ls
aclocal.m4 compile config.h.in config.sub COPYING install-sh ltmain.sh Makefile.in RELEASE.NOTES snort.8 so_rules verstuff.pl
cflags.out config.guess config.log configure cppflags.out libtool Makefile missing rules snort.pc stamp-h1 ylwrap
ChangeLog config.h config.status configure.in depcomp LICENSE Makefile.am preproc_rules sid-msg.map snort.pc.in VERSION
root@jabberwocky07-VirtualBox:/etc/snort#

Implementation

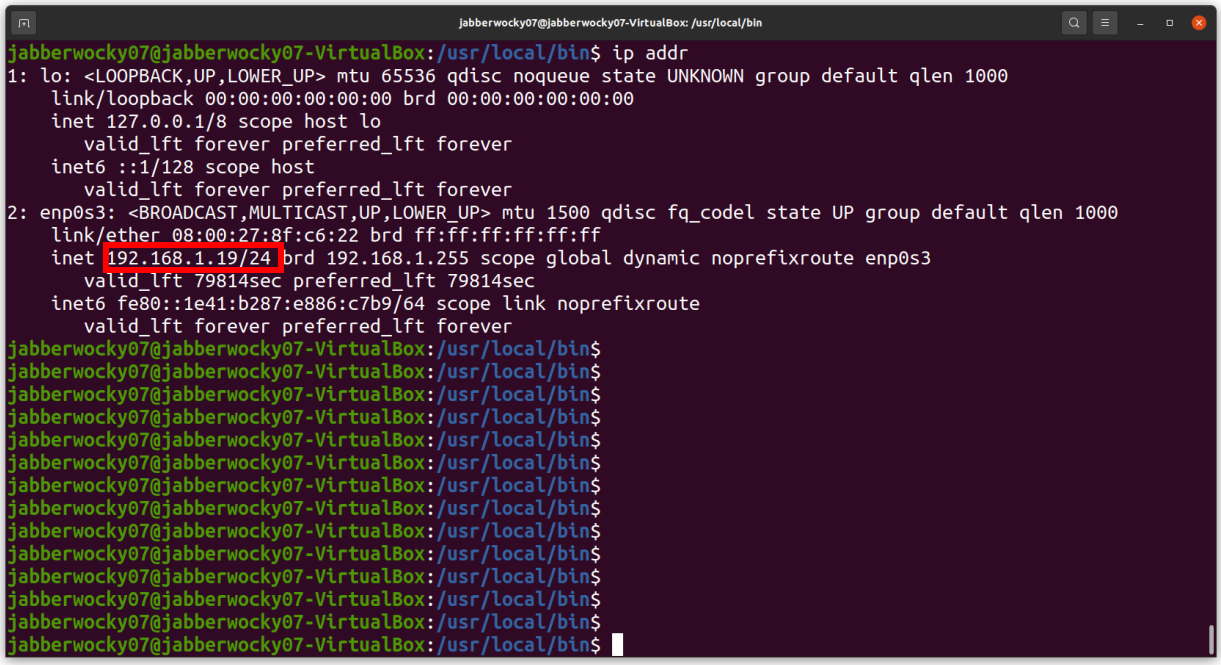
Project Name: NIDS Installation for Clinical Research Lab
Company: Barker Labs
Division: Computer Network
Department: Information Technology

Prepared by: Michaela Pierce
Project Manager: Michaela Pierce
Project Owner, IT Director: David Smith
Project Sponsor, Business Owner: John Barker

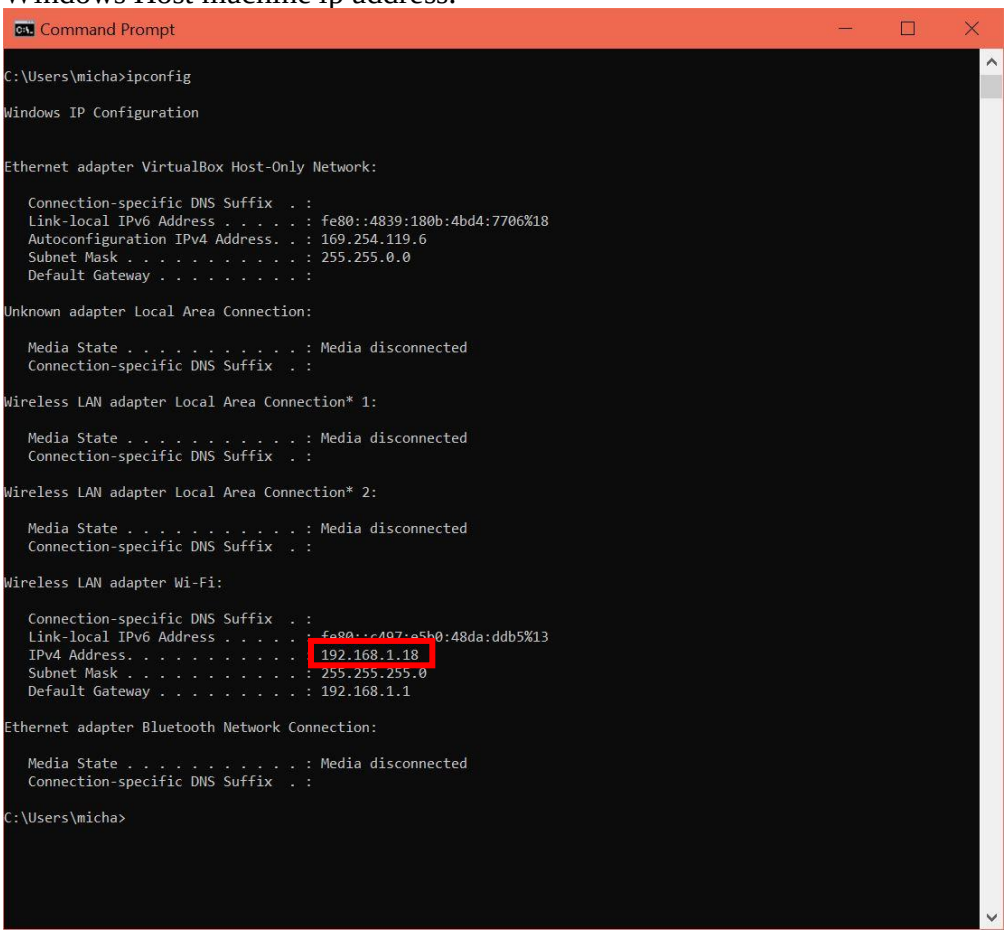
Version Control

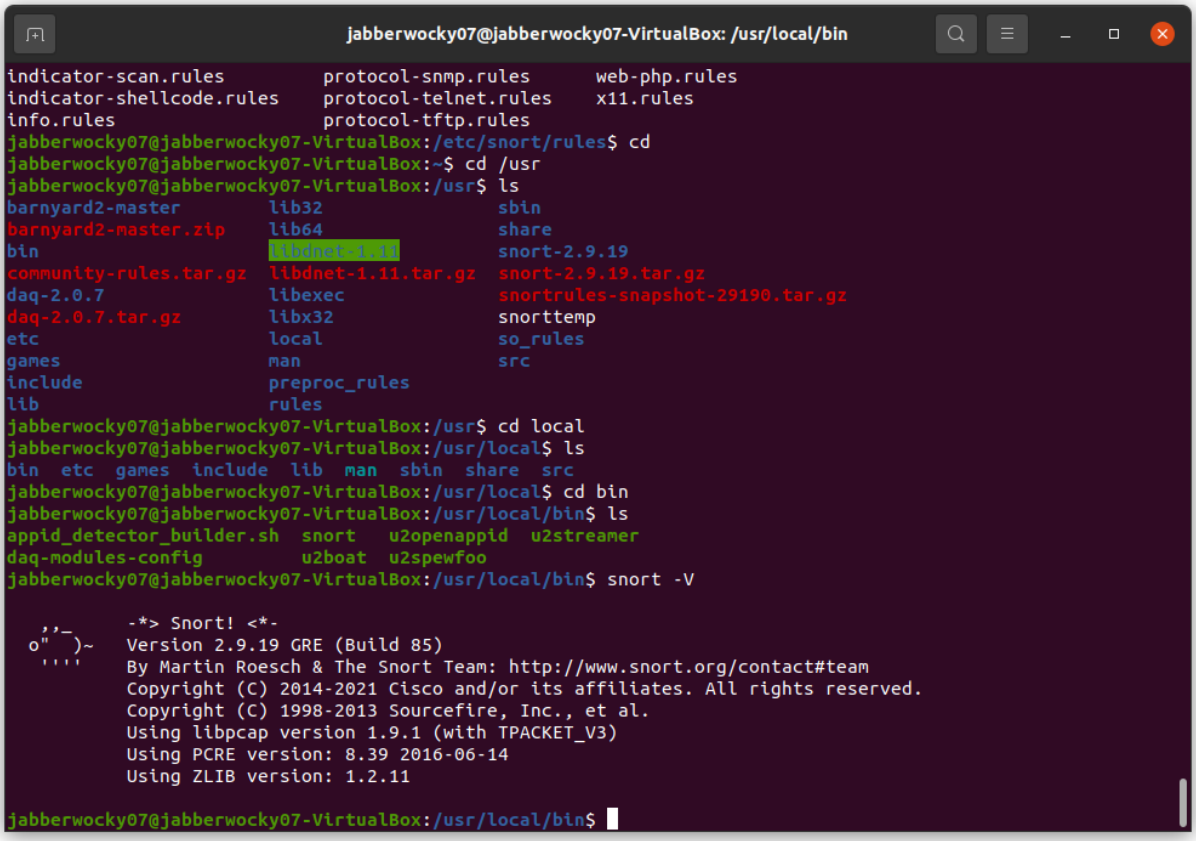
0.1 Charter created by Michaela Pierce – 02/10/2022
0.2 Charter edited by David Smith – 02/12/2022, Scope affected/ reduced
0.3 Problem Analysis created by Michaela Pierce – 02/20/2022
0.4 Problem Analysis edited by David Smith – 02/21/2022
0.5 Feasibility Analysis added to Problem Analysis, by Michaela Pierce – 02/26/2022
0.6 Requirements Analysis added by Michaela Pierce – 03/01/2022
0.7 Decision Analysis created by Michaela Pierce – 03/08/2022
0.8 Decision Analysis edited by David Smith – 03/10/2022
0.9 Design was added by Michaela Pierce – 03/15/2022
0.10 Construction was added by Michaela Pierce – 03/25/2022
0.11 Implementation was added by Michaela Pierce & David Smith – 04/10/2022

Virtual Machine – Ubuntu with the ip address:

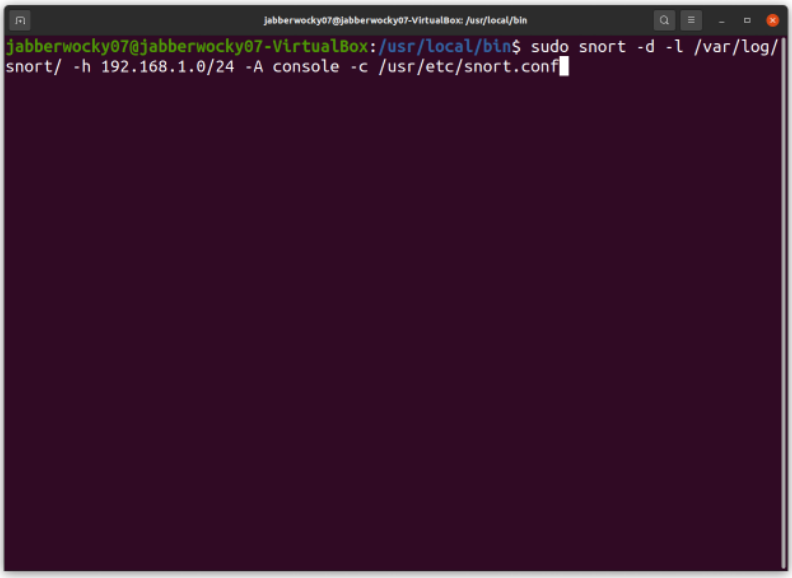


Windows Host machine ip address:

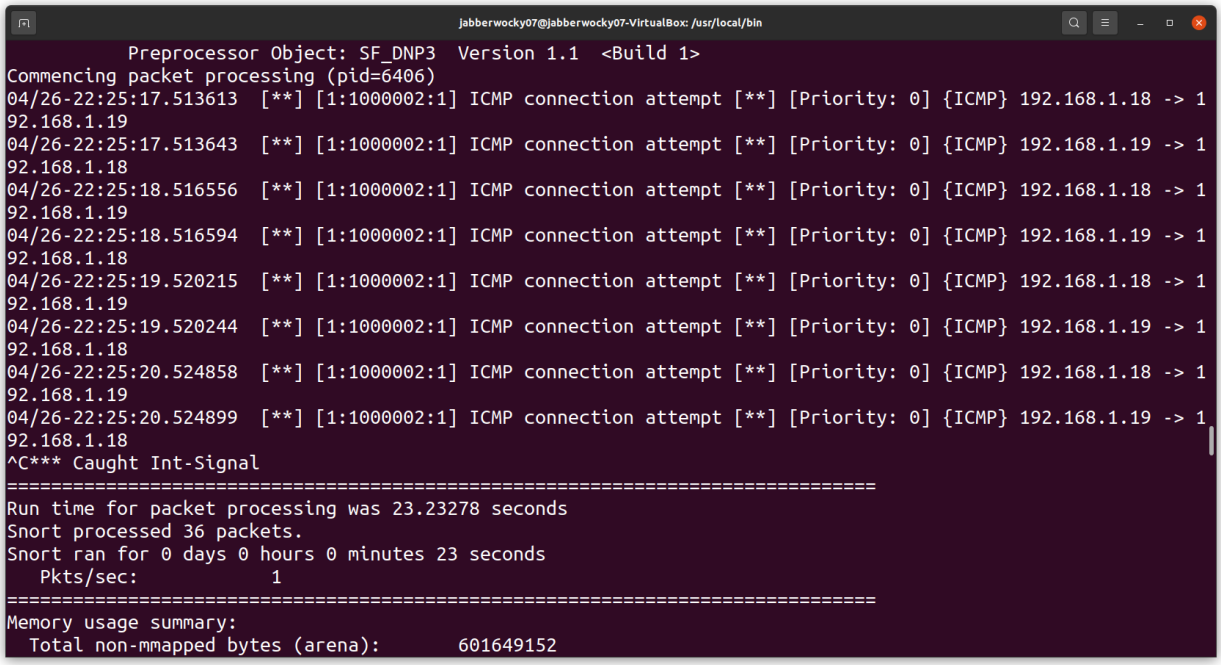


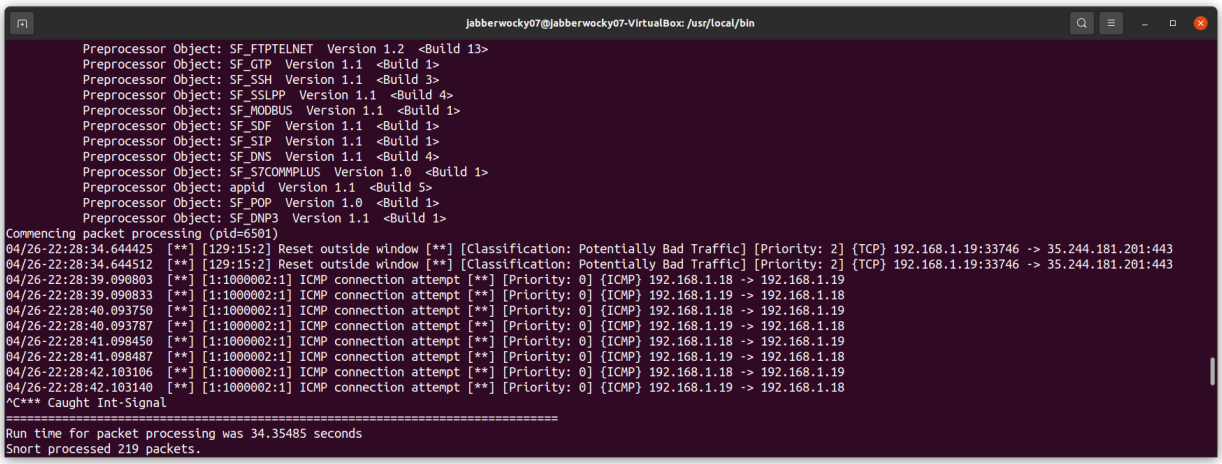


To start Snort



Ping command on host shows these results on Snort





Operations and Support

Project Name: NIDS Installation for Clinical Research Lab
Company: Barker Labs
Division: Computer Network
Department: Information Technology

Prepared by: Michaela Pierce
Project Manager: Michaela Pierce
Project Owner, IT Director: David Smith
Project Sponsor, Business Owner: John Barker

Version Control

0.1 Charter created by Michaela Pierce – 02/10/2022
0.2 Charter edited by David Smith – 02/12/2022, Scope affected/ reduced
0.3 Problem Analysis created by Michaela Pierce – 02/20/2022
0.4 Problem Analysis edited by David Smith – 02/21/2022
0.5 Feasibility Analysis added to Problem Analysis, by Michaela Pierce – 02/26/2022
0.6 Requirements Analysis added by Michaela Pierce – 03/01/2022
0.7 Decision Analysis created by Michaela Pierce – 03/08/2022
0.8 Decision Analysis edited by David Smith – 03/10/2022
0.9 Design was added by Michaela Pierce – 03/15/2022
0.10 Construction was added by Michaela Pierce – 03/25/2022
0.11 Implementation was added by Michaela Pierce & David Smith – 04/10/2022
0.12 Operations and Support was added by Michaela Pierce – 04/16/2022

Maintenance and Updates will be installed automatically. Snort rules can be configured and additional software installed.

These can be configured after the project installment.

Conclusion

Barker Labs is a clinical research lab that has created a cure-all vaccine for COVID-19, a one-time vaccine that will prevent COVID-19 and all relative variants. The vaccine is undergoing human testing but due to other competitors has become prominent, because of other vaccines requiring more than one dose, failure to prevent relative variants, major side effects, and unidentified future complications. Barker Labs was started two years ago and have grown over time along with their clinical research. The Information Technology (IT) Department have maintained the same Network Intrusion Detection System (NIDS) for the past two years. Threats were not a major concern due to the business being small and other competitors publishing their findings and making vaccines available to the public. The new Snort system will be beneficial for them in that it will provide better network security.

Appendix-A: Repository or Data Dictionary

Anomaly-based – The NIDS pays attention to network patterns and will detect a threat if there is abnormal traffic patterns

Downtime – A period of time in which the network will not be accessible

False Negative – The NIDS fails to detect and alert the user for an active attack, can be due to a failure in the NIDS

False Positive – The system detects normal traffic as a threat, can be due to weak configuration rules, outdated systems, etc.

GUI – Graphical User Interface, an interactive user interface that has buttons and other objects that can be clicked on and provide communication throughout the application

LAN – Local Area Network, a network where all devices are in one physical location, can be wired or wireless

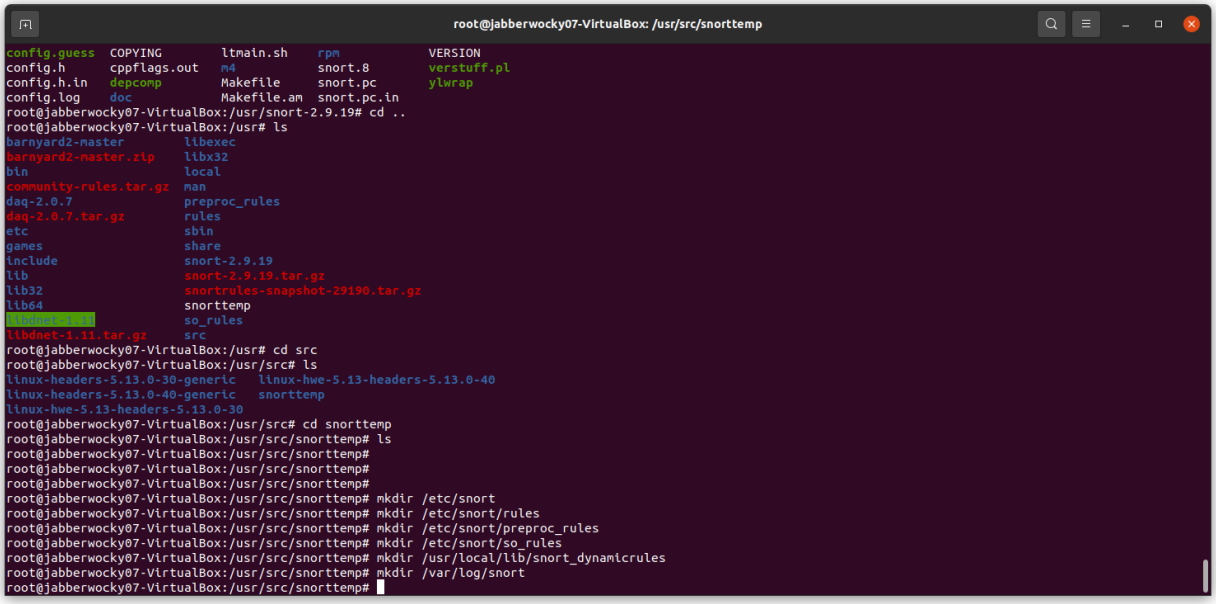
NIDS - Network Intrusion Detection System, a passive system that monitors the network traffic using anomaly-based or signature-based method and will alert the user if there is a threat

Positive – The system detects a threat in the network

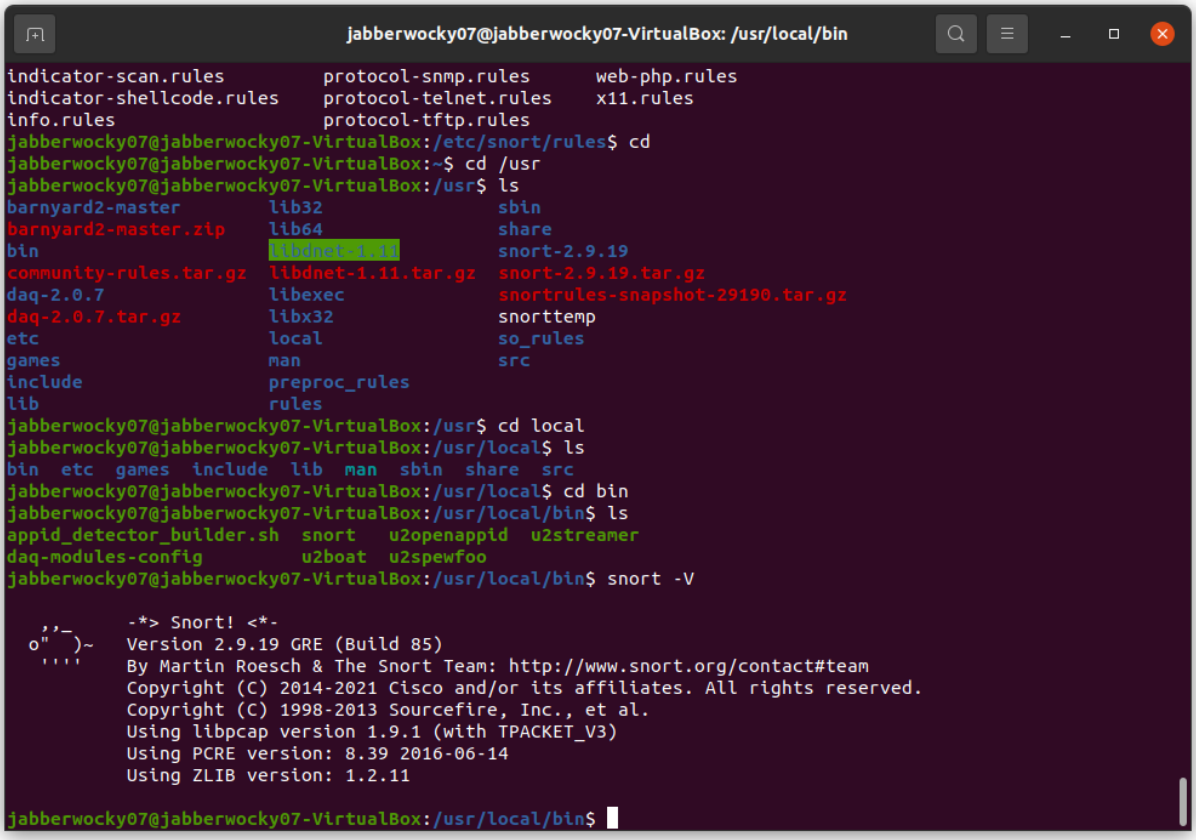
Signature-based – The NIDS compares traffic signatures to known attack signatures and will detect a threat if the signatures match

WAP – Wireless Access Point, hardware that connects wireless devices to a wired network and serves as Wi-Fi for a specific location.

Appendix-B: Sample Input



Appendix-C: Sample Output



Appendix-D: Reference/Bibliography

Cómo crear una página de manual en Linux | Systempeaker. (2021, October 17). Systempeaker. https://systempeaker.com/linux/como-crear-una-pagina-de-manual-en-linux/

Compiling Shared Object Rules | SecurityArchitecture.com. (2022). Securityarchitecture.com. https://www.securityarchitecture.com/learning/intrusion-detection-systems-learning-with-snort/compiling-shared-object-rules/

Rights, R. F. (2003). Global information assurance certification paper. GIAC.

Sample project charter. (n.d.). Retrieved March 8, 2022, from https://images.template.net/wp-content/uploads/2016/07/13105411/sample-project-charter.pdf

T I I I O O O N N N, T. (n.d.). *A*. https://oa.mo.gov/sites/default/files/CC-NetworkBasedIDS.pdf

The security risks of outdated software - Parker Software. (2019, April 25). Parker Software. https://www.parkersoftware.com/blog/the-security-risks-of-outdated-software/