USER MANUAL:

Network Anomaly Detection Suite

(First Edition)

1st Edition

Table of Contents

I. Introduction

A. Background

The Network Anomaly Detection Suite was developed in our bachelor's final design course ICOM5047 at the University of Puerto Rico, Mayaguez Campus, Electrical and Computer Engineering Department, with the purpose of applying our acquired knowledge throughout the Computer Engineering bachelor degree courses into a final project. The project was ours to design, develop, integrate and deploy; taking into consideration our client's needs and restrictions as to provide a scenario similar to that of when we graduate to prepare us for what's to come.

University Of Puerto Rico
Mayaguez Campus

B. Us

We are three students: Pedro Colon, Antoine Cotto and Marie A Nazario (Project Manager); in the Computer Engineering program of the University of Puerto, Mayaguez Campus.

C. Acknowledgments

We would like to give our thanks to our client, Luis Lugo, the system administrator of the Electrical and Computer Engineering at the University for his availability and patience throughout the process of the suite design and development as well as his constant support and feedback in the project. We would also like to thanks our two course professors, Dr. Nayda Santiago and Dr. Isidoro Couvertier, for their guidance through the development of the suite as well as their guidance throughout the project. Lastly, we would like to thank Dr. Suresh Damodaran for his help with the development of the KQL layer and overall feedback of the project.

II. About the Manual

This manual has been written with the intent to help users install NADS and verify its proper functionality once installed. It will take you step by step to ensure a proper installation as long as the system requirements and operating system are met.

III. System Overview

A. Technical Support

Technical Support will be provided upon request and evaluated case by case. Prices are still being discussed but will differ upon the scale and difficulty of the request. For any question please communicate through email to NADSsystem@gmail.com.

B. Hardware Requirements

NADS need at least the following hardware specifications to be able to function properly:

- Processor - 2GHz or faster
- Memory - 2GB of DDR3 RAM or more
- Hard Drive - 1GB of disk space or more
- Operating system - Debian or Fedora (CentOS)
- Networking hardware - STP, SNMP and SSH enabled switch
- Additional - Internet Access (required for text and email notification)

IV. Installation and Set Up

NADS installation slightly differs between Fedora and Debian OS's. The following sections will demonstrate the process for each of the operating systems as well as the necessary libraries for it be properly installed. The system should be able to work on other operating systems with a few changes made to the correct installation.

Note: of the installation instructions whenever the symbol ">>" shows up it means that the text next to it is a command.

A. Pre-Installation Notes

For the proper installation of NADS you will need to have the following previously installed in your operating system:

- The build-essential package or its equivalent
- Latest version of PIP (and include it in the system path)
- Have sudo

  Note: It's required for the scripts that install the NADS service.

After this is done you will now need to proceed to install or update the following: (These are essential software requirements for NADS)

1. Python 2.7.10 or greater (Not 3.*)
2. Php 5.2
3. Build-essential package
4. Nginx
5. Elasticsearch 2.2.1
6. Logstash
7. Kibana 4

If these are not already installed in your operating system, the following sections of the the manual will walk you through the installation process (seccion B for Fedora or seccion D for Debian).

There are **two** types of installation for both operating systems:

1. Processing Server.
   - A machine that will analyze log data to detect protocol attacks in the site and monitor switches for network loops. This will be your central processing hub (or "server").

     Note: There should always be at least 1 server per network with multiple nodes.

     Note 2: A server can be a node. If filebeat is installed and configured correctly, the server will in turn monitor itself for any attacks thus also becoming a node.
2. Node
   - A machine that will be monitored to prevent protocol attacks. This machine will be sending data to the Processing Server to process.

B. Installation on Fedora/CentOS (Automated)

Server Installation:

- Install Prerequisites

  >> sudo chmod 777 setup.sh
  >> sudo ./setup.sh

- NADS Setup

  a. Change into the deliverables folder
     >> cd nadsdeliverable/
  b. Edit the node configuration file for the system to have the correct paths
     >> vi Platform_Module/src/resources/configNode.json
     i. Edit the folder variables and any other path that might be pre written, to an actual path in the system
  c. Change into the service wrapper directory
     >> cd Platform_Module/src/resources/yajsw-beta-12.05/
  d. Edit the service configuration
     >> vi conf/wrapper.conf
     i. Change the variable wrapper.working.dir to where the Platform_Module/target folder is located
        Note: You can find this directory under Platform_Module/target
     ii. Change the wrapper.app.parameter.1 variable to point towards the node configuration(configNode.json)
        Note: You can find these under Platform_Module/src/resources
  e. Change into the bin folder
     >> cd bin/
  f. Install the service
     >> sudo ./installDaemon.sh
  g. Start the service (you can do either of the next three options)
     >> sudo ./startDaemon.sh
     >> sudo systemctl start nads
     >> sudo /etc/init.d/nads start

Node Installation:

- Install Prerequisites

  >> sudo chmod 777 setup.sh
  >> sudo ./setup.sh

- Filebeat installation

1. Copy the created certificate from the processing node over to the release directory
   >> cp /etc/pki/tls/private/logstash-forwarder.key .
2. Edit the filebeat configuration to set up processing server address
   i. Open the filebeat.yml file
   ii. Need the top of file find the line that says “paths:”
   iii. Under the path line fine the line that says “/var/log/auth.log” and change it to “/var/log/secure”
   iv. Search for “The Logstash hosts” (It will commented out)
   v. About two lines under the previous instrucctions there will be a line that looks like this: “” hosts: ["136.145.59.139:5044"] ””
   vi. Change the ip of the previous line to the one were the ELK server is located at.
   vii. Save the file
3. Run the filebeat install script
   >> sudo ./filebeatin_3_cent.sh

- NADS Setup

a. Change into the deliverables folder
   >> cd nadsdeliverable/
b. Edit the node configuration file for the system to have the correct paths
   >> vi Platform_Module/src/resources/confignode.json
   i. Edit the folder variables and any other path that might be pre written, to an actual path in the system
c. Change into the service wrapper directory
   >> cd Platform_Module/src/resources/yajsw-beta-12.05/
d. Edit the service configuration
   >> vi conf/wrapper.conf
   i. Change the variable wrapper.working.dir to where the Platform_Module/target folder is located
      Note: You can find this directory under Platform_Module/target

ii. Change the wrapper.app.parameter.1 variable to point towards the node configuration(confignode.json)

Note: You can find these under Platform_Module/src/resources

e. Change into the bin folder
   >> cd bin/
f. Install the service
   >> sudo ./installDaemon.sh
g. Start the service (you can do either of the next three options)
   >> sudo ./startDaemon.sh
   >> sudo systemctl start nads
   >> sudo /etc/init.d/nads start

C. Installation on Fedora/CentOS (manually)

Server Installation:

➢ Setup

a. Install Net-SNMP library
   >> yum -y install net-snmp net-snmp-utils
   >> yum -y install net-snmp-devel
b. Install tools for dependency compilation
   >> yum -y install gcc python-devel
c. Install epel repository
   >> yum -y install epel-release
d. Install pip
   >> yum -y install python-pip
e. Install compilation system
   >> yum groupinstall "Development Tools" "Development Libraries"
f. Install additional libraries
   >> yum -y install python-devel libxml2-devel libxml2 libxslt-devel libxslt xmlsec1 xmlsec1-openssl
g. Install dependency libraries
   >> yum -y install zlib-devel
h. Install pip pre-requisites
   >> pip install -r ./requirements.txt
   Note: Must be done within the release folder
i. If compiling from source (optional)
   >> yum -y install git
   >> git clone REPOSITORY

- NADS Setup

a. Change into the deliverables folder
   >> cd nadsdeliverable/
b. Edit the configuration file for the system to have the correct paths
   >> vi Platform_Module/src/resources/config.json
   i. Edit the folder variables and any other path that might be pre written, which does not exist
c. Change into the service wrapper directory
   >> cd Platform_Module/src/resources/yajsw-beta-12.05/
d. Edit the service configuration
   >> vi conf/wrapper.conf
   i. Change the variable wrapper.working.dir to where the Platform_Module/target folder is located
      Note: You can find this directory under Platform_Module/target
   ii. Change the wrapper.app.parameter.1 variable to point towards either the processing server configuration (config.json) or the node client (confignode.json)
      Note: You can find these under Platform_Module/src/resources
e. Change into the bin folder
   >> cd bin/
f. Install the service
   >> sudo ./installDaemon.sh
g. Start the service (you can do either of the next three options)
   >> sudo ./startDaemon.sh
   >> sudo systemctl start nads
   >> sudo /etc/init.d/nads start

Node Installation:

- Setup

a. Install Net-SNMP library
   >> yum -y install net-snmp net-snmp-utils
   >> yum -y install net-snmp-devel
b. Install tools for dependency compilation
   >> yum -y install gcc python-devel
c. Install epel repository
   >> yum -y install epel-release
d. Install pip
   >> yum -y install python-pip

e. Install compilation system
   >> yum groupinstall "Development Tools" "Development Libraries"
f. Install additional libraries
   >> yum -y install python-devel libxml2-devel libxml2 libxslt-devel libxslt xmlsec1 xmlsec1-openssl
g. Install dependency libraries
   >> yum -y install zlib-devel
h. Install pip pre-requisites
   >> pip install -r ./requirements.txt
   Note: Must be done within the release folder
i. If compiling from source (optional)
   >> yum -y install git
   >> git clone REPOSITORY

- Filebeat Setup

a. Create certificate folder
   >> sudo mkdir -p /etc/pki/tls/certs
b. Copy certificate generated from processing server to certificate folder
   >> sudo cp -f ./logstash-forwarder.crt /etc/pki/tls/certs/
c. Add the filebeat repository key
   >> sudo rpm --import http://packages.elastic.co/GPG-KEY-elasticsearch
d. Add filebeat repository
   >> sudo cp -f ./elastic-beats.repo /etc/yum.repos.d/
e. Install filebeat
   >> sudo yum -y install filebeat
f. Edit the filebeat configuration to set up processing server address
   i. Open the filebeat.yml file
   ii. Need the top of file find the line that says “paths:”
   iii. Under the path line fine the line that says “/var/log/auth.log” and change it to “/var/log/secure”
   iv. Search for “The Logstash hosts” (It will commented out)
   v. About two lines under the previous instrucctions there will be a line that looks like this: “” hosts: ["136.145.59.139:5044"] ””
   vi. Change the ip of the previous line to the one were the ELK server is located at.
   vii. Save the file
g. Copy the filebeat configuration
   >> sudo cp -f ./filebeat.yml /etc/filebeat/filebeat.yml
h. Restart filebeat service
   >> sudo systemctl start filebeat
   >> sudo systemctl enable filebeat

- NADS Setup

a. Change into the deliverables folder
   >> cd nadsdeliverable/
b. Edit the configuration file for the system to have the correct paths
   >> vi Platform_Module/src/resources/config.json
   i. Edit the folder variables and any other path that might be pre written, which does not exist
c. Change into the service wrapper directory
   >> cd Platform_Module/src/resources/yajsw-beta-12.05/
d. Edit the service configuration
   >> vi conf/wrapper.conf
   i. Change the variable wrapper.working.dir to where the Platform_Module/target folder is located
      Note: You can find this directory under Platform_Module/target
   ii. Change the wrapper.app.parameter.1 variable to point towards either the processing server configuration (config.json) or the node client (confignode.json)
      Note: You can find these under Platform_Module/src/resources
e. Change into the bin folder
   >> cd bin/
f. Install the service
   >> sudo ./installDaemon.sh
g. Start the service (you can do either of the next three options)
   >> sudo ./startDaemon.sh
   >> sudo systemctl start nads
   >> sudo /etc/init.d/nads start

D. Installation on Debian/Ubuntu (Automated)

If using Debian, the following procedure should be executed before commencing the installation on both Node and Server installations:

Debian sudo and apt-get configuration

1. Login as super user
   >> su
2. Comment out line that includes the cd in the following file
   >> /etc/apt/sources.list
3. Add the contrib and non-free repositories inside /etc/apt/sources.list
   i. Sources. List content:
      >> deb http://ftp.us.debian.org/debian/ jessie main contrib non-free

>> deb-src http://ftp.us.debian.org/debian/ jessie main contrib non-free
>> deb http://security.debian.org/ jessie/updates main contrib non-free
>> deb-src http://security.debian.org/ jessie/updates main contrib non-free
>> # jessie-updates, previously known as 'volatile'
>> deb http://ftp.us.debian.org/debian/ jessie-updates main contrib non-free
>> deb-src http://ftp.us.debian.org/debian/ jessie-updates main contrib non-free

4. Perform an update on packages
   >> apt-get update

5. Install sudo
   >> apt-get install sudo
6. Add user to sudoers file
   >> Visudo
   ii. Give permissions as needed. A simple (not very secure) alternative is: UserName ALL=(ALL:ALL) ALL
7. Exit root user
   >> exit

Server Installation:

Install Prerequisites

>> sudo chmod 777 ubuntu-requirements_1.sh
>> sudo ./ubuntu-requirements_1.sh

ELK Installation

1. Open openssl.cnf with a text editor
   >> Vi openssl.cnf
   - In the openssl.cnf find the field:
     [v3_ca]
   - Under it, add the following line
     subjectAltName =IP: 192.168.42.168
   - Be sure to replace the ip with the one the machine has
2. Run the install script
   - Edit the line that says cd ~/Downloads/install to reflect the release directory

   >> sudo ./elk_install_2.sh

- NADS Setup

a. Change into the deliverables folder
   >> cd nadsdeliverable/
b. Edit the configuration file for the system to have the correct paths
   >> vi Platform_Module/src/resources/config.json
   i. Edit the folder variables and any other path that might be pre written, which does not exist
c. Change into the service wrapper directory
   >> cd Platform_Module/src/resources/yajsw-beta-12.05/
d. Edit the service configuration
   >> vi conf/wrapper.conf
   i. Change the variable wrapper.working.dir to where the Platform_Module/target folder is located
      Note: You can find this directory under Platform_Module/target
   ii. Change the wrapper.app.parameter.1 variable to point towards either the processing server configuration (config.json) or the node client (confignode.json)
      Note: You can find these under Platform_Module/src/resources
e. Change into the bin folder
   >> cd bin/
f. Install the service
   >> sudo ./installDaemon.sh
g. Start the service (you can do either of the next three options)
   >> sudo ./startDaemon.sh
   >> sudo systemctl start nads
   >> sudo /etc/init.d/nads start

Node Installation:

- Install Prerequisites

>> sudo chmod 777 ubuntu-requirements_1.sh
>> sudo ./ubuntu-requirements_1.sh

- Filebeat installation (Node)

1. Copy the created certificate from the processing node over to the release directory
   >> cp /etc/pki/tls/private/logstash-forwarder.key .
2. Edit the filebeat configuration to send logs to the processing node
   >> vi filebeat.yml

i. Edit the line
hosts: ["192.168.42.168:5044"]
to reflect the ip of the processing server

4. Run the filebeat install script
>> sudo ./filebeatin_3.sh

➢ NADS Setup

a. Change into the deliverables folder
>> cd nadsdeliverable/
b. Edit the configuration file for the system to have the correct paths
>> vi Platform_Module/src/resources/confignode.json
i. Edit the folder variables and any other path that might be pre written, which does not exist
c. Change into the service wrapper directory
>> cd Platform_Module/src/resources/yajsw-beta-12.05/
d. Edit the service configuration
>> vi conf/wrapper.conf
i. Change the variable wrapper.working.dir to where the Platform_Module/target folder is located
Note: You can find this directory under Platform_Module/target
ii. Change the wrapper.app.parameter.1 variable to point towards the node configuration (confignode.json)
Note: You can find these under Platform_Module/src/resources
e. Change into the bin folder
>> cd bin/
f. Install the service
>> sudo ./installDaemon.sh
g. Start the service (you can do either of the next three options)
>> sudo ./startDaemon.sh
>> sudo systemctl start nads
>> sudo /etc/init.d/nads start

E. Installation on Debian/Ubuntu (Manual)

Server Installation:

➢ Debian sudo and apt-get configuration

1. Login as super user
>> su

2. Comment out line that includes the cd in the following file
   >> /etc/apt/sources.list
3. Add the contrib and non-free repositories inside /etc/apt/sources.list
   - Sources. List content:
     >> deb http://ftp.us.debian.org/debian/ jessie main contrib non-free
     >> deb-src http://ftp.us.debian.org/debian/ jessie main contrib non-free
     >> deb http://security.debian.org/ jessie/updates main contrib non-free
     >> deb-src http://security.debian.org/ jessie/updates main contrib non-free
     >> # jessie-updates, previously known as 'volatile'
     >> deb http://ftp.us.debian.org/debian/ jessie-updates main contrib non-free
     >> deb-src http://ftp.us.debian.org/debian/ jessie-updates main contrib non-free
4. Perform an update on packages
   >> apt-get update
5. Install sudo
   >> apt-get install sudo
6. Add user to sudoers file
   >> Visudo
   - Give permissions as needed. A simple (not very secure) alternative is: UserName ALL=(ALL:ALL) ALL
7. Exit root user
   >> exit

➢ General Setup

1. Install SNMP in the system
   >> apt-get install snmp
2. Install Git in the system (Optional)
   >> apt-get install git
   >> Git clone REPOSITORY
3. Instal various libraries
   >> apt-get -y install libxml2-dev libxslt1-dev python-dev
   >> apt-get install zlib1g-dev
4. Install SNMP additional libraries
   >> apt-get install libsnmp-dev snmp-mibs-downloader
5. Install PIP as well as other python libraries
   >> apt-get -y install python-pip python-dev build-essential

6. Change directory into the release directory
   >> cd /path/to/release/dir
7. Install the pip modules required by the project
   >> pip install -r requirements.txt
   - Note: must be sudo/root to run this command
   - Note 2: If there is an OpenSSL problem do the following:
     >> sudo easy_install --upgrade pip
     >> sudo pip install requests==2.6.0
8. Install Java *
   >> add-apt-repository -y ppa:webupd8team/java
   Note: if using debian do the following:
   - Earlier than Jessie
     >> sudo apt-get install python-software-properties
   - Older than Jessie
     >> sudo apt-get install software-properties-common
     >> sudo vi /etc/apt/sources.list.d/java-8-debian.list
   - Paste the following in the file:
     >> deb
     http://ppa.launchpad.net/webupd8team/java/ubuntu
     trusty main
     >> deb-src
     http://ppa.launchpad.net/webupd8team/java/ubuntu
     trusty main
   - Add the repository key
     >> sudo apt-key adv --keyserver keyserver.ubuntu.com
     --recv-keys EEA14886

   >> apt-get update
   >> apt-get -y install oracle-java8-installer

➢ ELK Stack set up

1. Import the Elasticsearch public GPG key
   >> wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch
   | sudo apt-key add -
2. Add elasticsearch to the repository list
   >> echo "deb http://packages.elastic.co/elasticsearch/2.x/debian
   stable main" | sudo tee -a /etc/apt/sources.list.d/elasticsearch-
   2.x.list
   >> sudo apt-get update
3. Install elasticsearch version 2.2.1 (**It needs to be this one**)
   >> sudo apt-get -y install elasticsearch=2.2.1

4. Copy elasticsearch configuration file to appropriate folder from current directory
   >> sudo cp -f ./elasticsearch.yml /etc/elasticsearch/
5. Start elasticsearch service and add it to run on boot list
   >> sudo service elasticsearch restart
   >> sudo update-rc.d elasticsearch defaults 95 10
6. Install KQL-SQL elasticsearch plugin
   >> sudo /usr/share/elasticsearch/bin/plugin install file:./elasticsearch-sql-2.2.1.zip
7. Add kibana to the repository list
   >> echo "deb http://packages.elastic.co/kibana/4.4/debian stable main" | sudo tee -a /etc/apt/sources.list.d/kibana-4.4.x.list
   >> sudo apt-get update
8. Install kibana
   >> sudo apt-get -y install kibana
9. Copy kibana configuration file to appropriate folder from current directory
   >> sudo cp -f ./kibana.yml /opt/kibana/config/
10. Start kibana service and add it to run on boot list
    >> sudo update-rc.d kibana defaults 96 9
    >> sudo service kibana start
11. Install Nginx
    >> sudo apt-get install nginx apache2-utils
12. Add password to be used when accessing kibana through Nginx (NOTE: If password inputted when running the script run this command alone after script completion to add correct password)
    >> sudo htpasswd -c /etc/nginx/htpasswd.users kibanaadmin
13. Copy Nginx config file to correct location from current directory
    >> sudo cp -f ./default /etc/nginx/sites-available/
14. Copy php dependencies to nginx server
    >> sudo mkdir -p /var/www/html
    >> sudo cp -r ./sqlparser /var/www/html/
15. Restart Nginx service
    >> sudo service nginx restart
16. Add logstash to the repository list
    >> echo 'deb http://packages.elastic.co/logstash/2.2/debian stable main' | sudo tee /etc/apt/sources.list.d/logstash-2.2.x.list
    >> sudo apt-get update
17. Install Logstash
    >> sudo apt-get install logstash
18. Create certificate directories
    >> sudo mkdir -p /etc/pki/tls/certs

>> sudo mkdir /etc/pki/tls/private

19. Set ip address for certificate generation
    - Open openssl.cnf with a text editor
      >> Vi openssl.cnf
    - In the openssl.cnf find the field:
      [v3_ca]
    - Under it, add the following line
      subjectAltName =IP: 192.168.42.168
    - Be sure to replace the ip with the one the machine has
20. Copy certificate generation configuration
    - Copy certificate generation configuration

    >> sudo cp -f ./openssl.cnf /etc/ssl
21. Change directory to certificate directory
    >> cd /etc/pki/tls
22. Generate certificates for nodes
    >> sudo openssl req -config /etc/ssl/openssl.cnf -x509 -days 3650 -batch -nodes -newkey rsa:2048 -keyout private/logstash-forwarder.key -out certs/logstash-forwarder.crt
23. Change to the release directory
    >> cd ~/Downloads/install
    - May vary depending on where the folder was extracted
24. Copy Logstash configuration files
    >> sudo cp -f ./02-beats-input.conf /etc/logstash/conf.d/
    >> sudo cp -f ./30-elasticsearch-output.conf /etc/logstash/conf.d/
    >> sudo cp -f ./10-syslog-filter.conf /etc/logstash/conf.d/
25. Restart logstash service and set it to startup
    >> sudo service logstash restart
    >> sudo update-rc.d logstash defaults 96 9
26. Install PHP and its dependencies
    >> sudo apt-get install php5
    >> sudo apt-get install php5-fpm
    - Ignore the errors that this may produce

- Install and start NADS service

1. Change into the deliverables folder
   >> cd nadsdeliverable/
2. Edit the configuration file for the system to have the correct paths
   >> vi Platform_Module/src/resources/config.json
   - Edit the folder variables and any other path that might be pre written, which does not exist
3. Change into the service wrapper directory
   >> cd Platform_Module/src/resources/yajsw-beta-12.05/

4. Edit the service configuration
   >> vi conf/wrapper.conf
   - Change the variable wrapper.working.dir to where the Platform_Module/target folder is located
     - You can find this directory under Platform_Module/target
   - Change the wrapper.app.parameter.1 variable to point towards either the processing server configuration (config.json) or the node client (confignode.json)
     - You can find these under Platform_Module/src/resources
5. Change into the bin folder
   >> cd bin/
6. Install the service
   >> sudo ./installDaemon.sh
7. Start the service (use either one of the following options)
   >> sudo ./startDaemon.sh
   >> sudo service nads start
   >> sudo /etc/init.d/nads start

Node Installation:

Debian sudo and apt-get configuration

1. Login as super user
   >> su
2. Comment out line that includes the cd in the following file
   >> /etc/apt/sources.list
3. Add the contrib and non-free repositories inside /etc/apt/sources.list
   - Sources. List content:
     >> deb http://ftp.us.debian.org/debian/ jessie main contrib non-free
     >> deb-src http://ftp.us.debian.org/debian/ jessie main contrib non-free
     >> deb http://security.debian.org/ jessie/updates main contrib non-free
     >> deb-src http://security.debian.org/ jessie/updates main contrib non-free
     >> # jessie-updates, previously known as 'volatile'
     >> deb http://ftp.us.debian.org/debian/ jessie-updates main contrib non-free
     >> deb-src http://ftp.us.debian.org/debian/ jessie-updates main contrib non-free
4. Perform an update on packages

>> apt-get update

5. Install sudo
   >> apt-get install sudo
6. Add user to sudoers file
   >> Visudo
   - Give permissions as needed. A simple (not very secure) alternative is: UserName ALL=(ALL:ALL) ALL
7. Exit root user
   >> exit

General Setup

1. Install SNMP in the system
   >> apt-get install snmp
2. Install Git in the system (Optional)
   >> apt-get install git
   >> Git clone REPOSITORY
3. Instal various libraries
   >> apt-get -y install libxml2-dev libxslt1-dev python-dev
   >> apt-get install zlib1g-dev
4. Install SNMP additional libraries
   >> apt-get install libsnmp-dev snmp-mibs-downloader
5. Install PIP as well as other python libraries
   >> apt-get -y install python-pip python-dev build-essential
6. Change directory into the release directory
   >> cd /path/to/release/dir
7. Install the pip modules required by the project
   >> pip install -r requirements.txt
   - Note: must be sudo/root to run this command
   - Note 2: If there is an OpenSSL problem do the following:
     >> sudo easy_install --upgrade pip
     >> sudo pip install requests==2.6.0
8. Install Java *
   >> add-apt-repository -y ppa:webupd8team/java
   - Note if using debian do the following:
     - Earlier than Jessie
       >> sudo apt-get install python-software-properties
     - Older than Jessie
       >> sudo apt-get install software-properties-common
       >> sudo vi /etc/apt/sources.list.d/java-8-debian.list

- Paste the following in the file:
  >> deb
  http://ppa.launchpad.net/webupd8team/java/ubuntu
  trusty main
  >> deb-src
  http://ppa.launchpad.net/webupd8team/java/ubuntu
  trusty main
- Add the repository key
  >> sudo apt-key adv --keyserver keyserver.ubuntu.com
  --recv-keys EEA14886

>> apt-get update
>> apt-get -y install oracle-java8-installer

Install filebeat

1. If you want the machine installed to be monitored do the following:
2. Copy the created certificate from the processing node over to the release directory. The certificate inside this directory is the one that will be used to ship logs to the processing server (Optional)
   >> cp /etc/pki/tls/private/logstash-forwarder.key
3. Create certificate folder
   >> sudo mkdir -p /etc/pki/tls/certs
4. Copy certificate generated from processing server to certificate folder
   >> sudo cp -f ./logstash-forwarder.crt /etc/pki/tls/certs/
5. Add filebeat repository
   >> echo "deb https://packages.elastic.co/beats/apt stable main" | sudo tee -a /etc/apt/sources.list.d/beats.list
   >> sudo apt-get update
6. Add the filebeat repository key
   >> wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
7. Install filebeat
   >> sudo apt-get install filebeat
8. Edit the filebeat configuration to set up processing server address
   - Open the filebeat.yml file
   - Need the top of file find the line that says “paths:”
   - Under the path line fine the line that says “/var/log/auth.log” and change it to where the ssh logs are located if different. Do the same for the mail log path.
   - Search for “The Logstash hosts” (It will commented out)
   - About two lines under the previous instrucctions there will be a line that looks like this: “” hosts: ["136.145.59.139:5044"] ””

- Change the ip of the previous line to the one were the ELK server is located at.
- Save the file

9. Copy the filebeat configuration
   >> sudo cp -f ./filebeat.yml /etc/filebeat/filebeat.yml
10. Restart filebeat service
    >> sudo service filebeat restart
    >> sudo update-rc.d filebeat defaults 95 10

➢ Install and start NADS service

1. Change into the deliverables folder
   >> cd nadsdeliverable/
2. Edit the configuration file for the system to have the correct paths
   >> vi Platform_Module/src/resources/config.json
   - Edit the folder variables and any other path that might be pre written, which does not exist
3. Change into the service wrapper directory
   >> cd Platform_Module/src/resources/yajsw-beta-12.05/
4. Edit the service configuration
   >> vi conf/wrapper.conf
   - Change the variable wrapper.working.dir to where the Platform_Module/target folder is located
     - You can find this directory under Platform_Module/target
   - Change the wrapper.app.parameter.1 variable to point towards either the processing server configuration (config.json) or the node client (confignode.json)
     - You can find these under Platform_Module/src/resources
5. Change into the bin folder
   >> cd bin/
6. Install the service
   >> sudo ./installDaemon.sh
7. Start the service (use either one of the following options)
   >> sudo ./startDaemon.sh
   >> sudo service nads start
   >> sudo /etc/init.d/nads start

F. Restarting/Changing Configuration File

Whenever there is a change in the configuration file, the NADS service must be restarted. This can be done in different manners:

- Run stop service script and start service script
- Run Debian service restart
- Run Fedora service restart

G. Most Common Errors

Here is a list of the most common errors and some possible solutions:

1. Filebeat ssl error problem
   - The certificate used during the installation is not the one that is being used in the processing node or the ip address in the openssl.cnf that is used to generate the certificates is not the one of the processing server. To remedy this, regenerate the ssl certificates in the processing server and put them in the appropriate location in the node that needs to be monitored.

2. Service start problem
   - Some installations may have a problem with starting the service. This can be due to one of two things.
     - The configuration of the service found under /path/to/release/Platform_Module/src/resources/yajsw-12/conf/wrapper.conf may be wrong. Revise the paths in this configuration file to make sure they exist
     - The service may not be registered after installing it through the install script. To remedy this simply start the service manually: sudo /etc/init.d/nads start
3. Startup problem
   - Sometimes the configuration of nads is not correct or an unforseen dependency has not been met. To remedy this issue revise the nads configuration under /path/to/release/Platform_Module/src/resources/config.json or confignode.json.
   - To verify if there is an unmet dependency run this
     >> java -jar /path/to/release/Platform_Module/target/Platform*.jar /path/to/release/Platform_Module/src/resources/config.json logfile.log
     - Here any additional problems may pop up. If there is a missing Python dependency simply run pip install missingdependency
4. Port in use problem
   - This problem may arise if the system was somehow not able to shut down correctly or if there is another program using one of the systems ports. To solve this problem either kill the process using the port or change the NADS configuration to use another port.

V. System Checks/Test

A. During Operation

1. Check if the NADS logfile does not contain any errors.
2. If running the SSH/SMTP detection algorithm, verify that example.log contains data. If not, review the common errors section.
3. Run “ps -aux | grep python” and make sure one instance of the notification and the loop detection scripts are running as well as multiple instances of the commander script.
4. Run “ps -aux | grep java” to verify if the Java process of the wrapper service is running.
5. Run sudo service nads status to verify if NADS is running or stopped.

B. After Installation

1. You can run some of the system’s parts individually and check for errors.

VI. Final Remarks

A. Adding New Algorithms

In order to add a new algorithm script to the system several things must be done. First the python scripts’ only input should be a json string containing a map of all the necessary parameters. For example “python loop_detec.py “”ips”: [\”192.168.0.1\”]”. It is very important that it accepts only a map of the parameters. Secondly on the config file for nads you must add the scripts and its parameter to the list belonging to the algorithms tuple. An example of how this looks like in the config file is this:

```
"algorithms":
        [
                {
                        "testalgorithm":{
                                "ips":"[\"136.145.59.152\"]",
                                "exampleip":"value",
                                "model":"values",
                                "trap_oid":                                      "value",
                                "folder":"../algorithms/loop_detec/test.py"}
                },
                {
```

```
                "sshdetectionserver":{
                        "server":"true",
                        "serveraddress":"localhost:8003",
                        "verbose":"true",
                        "folder":"../algorithms/sshdetserver/commander.py"
                }
        }
],
```

As mentioned before just add another object to the list for each new script. Make sure all the necessary fields for the script are here as the json map that will be passed to the script will be constructed from these objects.

B. Adding protocol to monitor

To add a protocol to the protocol monitoring algorithm there are some steps that need to be followed.

1. Processing Server

a) Add a new algorithm as mentioned previously to the central processing server configuration. It is simpler to copy one of the existing protocol algorithm entries. Here is an example of a protocol algorithm server entry:

```
{
        "sshdetectionserver":{

                "client":"true",
                "serveraddress":"localhost:8003",
                "Verbose":"true",
                "folder":"/home/pedro/Documents/git/nads/algorithms/ssh
                detserver/commander.py",
                "supported_protocols":"{\"SSH\":\"sshd\",\"SMTP\":\"smtpd\"}"
                ,
                "protocol":"SSH",
                "monitoringfolder":"/var/log/",
                "monitoringfile":"auth.log",
                "dataaddress":"127.0.0.1:8002",
                "serverport":8003,
                "clientport":8004,
                "whitelist":"[\"136.145.*\", \"127.0.0.1\"]"
        }
}
```

b) Edit the serveraddress to an unused port:
   - "serveraddress":"localhost:8010"
c) Edit the serverport to the same port as in server address
   - "serverport":8010
d) Edit the clientport to an unused port
   - "clientport":8011
e) Edit the supported protocols to add the protocol:
   - For convenience add the new protocol name in all caps (NEWPROTOCOL) and the name that it has in the logs (protocoldaemonlogname)
     "supported_protocols":"{\"SSH\":\"sshd\",\"SMTP\":\"smtpd\",\"NEWPROTOCOL\":\"protocoldaemonlogname\"}",
f) Edit the protocol field to reflect the new protocol:
   - “protocol”: “NEWPROTOCOL”
g) Change the algorithm name to something relatable to the protocol
   - Instead of sshdetectionserver something like newprotocoldetectionserver
h) Edit the monitoringfolder field and the monitoring file field
   - Monitoringfolder is the folder that contains the daemon’s log files "monitoringfolder":"/var/log/",
   - Monitoringfile is the actual file that contains the output that the daemon protocol uses "monitoringfile":"newprotocoloutput.log",
i) All in all it should look similar to this:

```
{
        "newprotocoldetectionserver":{
                "server":"true",
                "client":"true",
                "serveraddress":"localhost:8010",
                "verbose":"true",
                ”folder":"/home/pedro/Documents/git/nads/algorithms/sshd
                        etserver/commander.py",
                "supported_protocols":"{\"SSH\":\"sshd\",\"SMTP\":\"smtpd\
                        ",\"NEWPROTOCOL\":\"protocoldaemonlogname\"}",
                "protocol":"NEWPROTOCOL",
                "monitoringfolder":"/var/log/",
                "monitoringfile":"auth.log",
                "dataaddress":"127.0.0.1:8002",
                 "serverport":8010,
                "clientport":8011,
                "whitelist":"[\"136.145.*\", \"127.0.0.1\"]"
        }
}
```

j) Now we need to add a regular expression for the log files of the new protocol. Ideally, these log files should have line entries that say when a login was failed/succeeded and the ip and login that was used in the attempt.

   a. Here is an example of a failed and succeeded line from an ssh log and its corresponding regular expression. More of these can be found in the logstash grok file (10-syslog-filter.conf)

      i. "message", "%{SYSLOGTIMESTAMP:Date} (?:%{SYSLOGFACILITY})?%{SYSLOGHOST:HostName} %{SYSLOGPROG:DaemonName}: %{WORD:Status} %{WORD} %{WORD} (invalid user)?%{WORD:UserName} %{WORD} %{IP:ClientIp} %{WORD} %{NUMBER:ClientPort} %{WORD:ClientDaemon}"

      ii. The grok regular expression can be broken down into simple parts

          Parseable:

              Data that we want to store in a field. We use this kind of regular expression

          %{NUMBER:ClientPort}

%{} Indicates that the content in that point of the string shall be parsed.
NUMBER indicates the type of the content. It could be NUMBER, WORD,IP, and other common types.
:ClientPort indicates the field that it will be stored to

          Non-Useful

          Data that does not need to be stored in a field. The regular expression is just the string content i.e.:
(Invalid user)
It is just part of a string that we know will be there, but we will not utilize.

k) After the regular expressions are set, simply copy the configuration file over to /etc/logstash/conf.d/ and restart the logstash service.

2. Monitoring Node

a) It is the same process as the Server, but in the configuration file should be the confignode.json file and the algorithm should entry should have the field "server":"true" should be "server":"false"

C. Configuration File Explanation

```
{
 This section contains entries that configure the main platform.
  "main":
  {
        "logtofile":"nads.log"
        This entry contains the name of the file that the system will log to. This
        logfile will appear wherever the NADS jar is being run from.  Typically
        within the Platform_Module/target/ directory
  },
This section contains the configurations  for the different algorithms that the
platform will run.
"algorithms":
  [
        {
              "loop_detection":{
                    "ips":"[<<insert Switch IP'S>>]",
                    This field contains the IPs of the switches that will be
                    monitored.
                    "exampleip":"<<insert value>>",

                    "model":"<<insert values>>",
                    "trap_oid": "<<insert value>>",

                    "folder":"<<path to NADS folder>>/nads/algorithms/
                    loop_detec/Main_Script_Prototype.py"
                    This entry contains the path to the loop detection script.
                    }
        },
        This is an example of the configuration for the loop detection Algorithm.
        {
              "sshdetectionserver":{
                    "server":"true",
                    "client":"true",
                    "serveraddress":"localhost:8003",
                    "verbose":"true",
                   "folder":"<<path to NADS folder>>/nads/algorithms/
                   sshdetserver/commander.py",
                   "supported_protocols":"{\"SSH\":\"sshd\",\"SMTP\":\"smtpd\"}",
                    "protocol":"SSH",
```

```
                "monitoringfolder":"/var/log/",
                "monitoringfile":"auth.log",
                "dataaddress":"127.0.0.1:8002",
                "serverport":8003,
                "clientport":8004,
                "whitelist":"[\"136.145.*\", \"127.0.0.1\"]"
        }
},
```

This is an example entry for the SSH protocol attack detection algorithm. See the next entry for a more detailed explanation.

```
{
        "smtpdetectionserver":{
                "server":"true",
```

This field specifies whether it will act as a server node. false deactivates the option.

```
                "client":"true",
```

This field specifies whether it will act as a server node. false deactivates the option.

```
                "serveraddress":"localhost:8005",
```

This field specifies the address that will be used by a client to contact a server.

```
                "verbose":"true",
```

This field provides a more in depth output logging.

```
                "folder":"<<path to NADS folder>>/nads/algorithms/
                sshdetserver/commander.py",
```

This field contains the path to the commander.py script which runs the algorithm.

```
                "supported_protocols":"{\"SSH\":\"sshd\",\"SMTP\":\"smtpd\"}",
```

This field contains a dictionary of the protocols that are currently supported.

```
                "protocol":"SMTP",
```

This field contains the protocol that is being monitored. It must be one from the previous field.

```
                "monitoringfolder":"/var/log/",
```

This field contains the folder that contains the log will be monitored for logins.

```
                "Monitoringfile":"mail.log",
```

This field contains the name of the log file that will be monitored.

```
                "dataaddress":"127.0.0.1:8002",
```

This field contains the address of a server node that can be queried to get data.

```
                "Serverport":8005,
```

This field contains the port that the server node will bind to. It must be the same as above in the serveraddress field.

```
                "Clientport":8006,
```

This field contains the port that the client node will bind to.

```
"whitelist":"[\"136.145.*\", \"127.0.0.1\"]",
```

This field contains a list of IPs and IP ranges that will not be banned. An IP range is specified through an asterisk; i.e. “136.145.*”

```
"parameters":"{\"tuning_mu\":\"3\",
\"tuning_k\":\"2\",\"tuning_h\":\"3\",\"tuning_average_ooc_a
rl\":\"3\",\"tuning_ooc_med_thresh\":\"4\",
\"tuning_event_threshold\":\"10\"}"
```

This field contains a dictionary with parameters that can be tuned to speed up or make more precise the detection algorithm. These can be tweaked and tested to give better performance.

```
                }
        }
  ],
```

This section is currently not being used but contains fields similar to the data retrieval section.

```
  "visual":
  {
        "kibanaport":"70000",
        "kibanaaddress":"127.0.0.0"
  },
```

This section configures the address that will be used to retrieve data from the ELK stack.

```
  "dataretrieval":
  {
        "elasticsearchport":"9200",
```

This entry contains the port of the installation of Elasticsearch.

```
        "elasticsearchaddress":"localhost"
```

This entry contains the address of where the Elasticsearch installation was done. It is very common to have localhost so as to control access to the data retrieval.

```
  },
```

The notification section contains the configuration for users to be notified and by whom.

```
  "notification":
  {
        "users":
        [
                {
                        "name":"Pedro",
```

This field contains the name of the registered user.

```
                        "phonenumber":"7873604991",
```

This field contains the phone number of the registered user. Leave blank if you do not want to be contacted by text message

```
                        "phoneprovider":"att",
```

```
                    This field contains the provider of the phone
                    company.  Examples are verizon, sprint, claro, etc.
                    "email":"pedrocolon93@gmail.com",
                    This field contains the email of the user that will be
                    notified. Leave blank if you do not wish to be notified by
                    email.
                    "Notifiablealgorithms":"[\"sshdetectionserver\",\"smtpdetec
                    tionserver\",\"loop_detection\"]"
                    This field is a list that contains the names of the
                    algorithms that a registered user wants to be notified of.
                },
                This is an example of a user entry.
                {
                    "name":"Pedro2",
                    "phonenumber":"7873604992",
                    "phoneprovider":"att",
                    "email":"pedrocolon93@gmail.com",
                    "notifiablealgorithms":"[\"sshdetectionserver\",\"smtpdetect
                    ionserver\",\"loop_detection\"]"
                }
            ],
            This entry is a list of the users that will be notified in the system
            "notificationemail": "acotto777@gmail.com",
            This entry contains the email that will be the sender for the notifications.
            "Serverhost":"smtps.ece.uprm.edu",
            This entry contains the location of an SMTP server that the app will use
            to send out notifications
            "notificationfilelocation":"<<path to NADS folder>>/nads/notification/
            notificationThroughSMTPserver.py"
            This entry contains the location to the Python script which will act as the
            notifications module
    },

The utilities section contains 2 variables that hold the path to the old
configuration file for the regular expressions and the path to a new configuration
to swap in.  A user would change this if he is using a new set of grok regular
expressions
    "utilities":
    {
        "newpatternfilelocation":"<<path to NADS folder>>/nads/
        Platform_Module/src/resources/10-syslog-filter.conf",
        This entry contains the location of the new configuration file to swap in
        "originalpatternfilelocation":"/etc/logstash/conf.d/10-syslog-filter.conf"
        This entry contains the location of the old configuration file
    }
}
```

See the example configurations config.json and confignode.json found in the Platform_Module/src/resources/ folder.