C1C’s Cavassa, Herbert, & Johnston

CS439, Period T6-7, Captain Collins

5 May 2017

CyOpSE README

NOTE: The purpose of this README is to direct future developers and users on how to use CyOpSE in its current state.

CONTACT INFORMATION:

John Cavassa & Eric Johnston will be at USAFA for casual Lt. work. Ren Herbert will be at pilot training in Vance. Please feel free to contact us if you have any questions.

Additionally, our CUBIC mentors were:

Will McBurnett (McBurnett, Will (US) [McBurnettW@intific.com](mailto:McBurnettW@intific.com))

Beau Albiston (Albiston, Beau (US) [AlbistonB@intific.com](mailto:AlbistonB@intific.com))

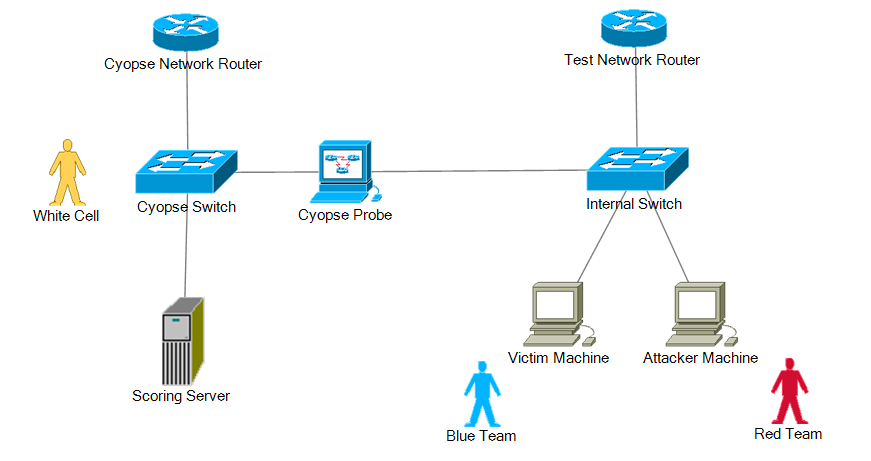
And our USAFA mentor was:

Captain Susan Collins

**Networking & Infrastructure Setup:**

* Required Equipment:
  + Two network routers
  + Minimum 3 Laptops
    - Victim Machine
    - Attacker Machine
    - Server (White Cell) Machine
  + Raspberry Pi Probe
  + Hard wired connections
    - Test network with victim, attacker, and router configured for wired only.
  + Wireless connections
    - CyOpSE probe to CyOpSE server through second router configured for wireless only. The probes must communicate through a wireless connection to the server, which will emulate real world tests and simulations.

**Test Network Layout:**



**Front End Web Interface Mockup:**

* Required:
  + Install Balsamiq version 3.5 or later to run.
* The mockup file is contained in CyOpSE\_Website\_Mockup.bmpr
  + In this file, you’ll see the proposed website layout and how each piece will communicate with each other to provide a cohesive interface for white cell users.
  + The intent is to provide increased levels of information as needed. The battlespace map will be difficult to implement, but will allow users to have a visual grasp on how the exercise is running. It may be impossible to capture a 100% image of the network through a diagram, but at least it can provide a good start for understanding.
* To run, simply open the file and click the play/demo button in the upper right hand corner of the window.
  + Navigate the interface!

**Web Interface:**

* At this point in time, no additional dependencies are required. The current web interface uses HTML5 and CSS pages to run.
  + In the future, functionality will need to be added for XAMPP, or equivalent system in order to run a MySQL database.
  + Code has already been added into the back end of the web interface to be able to import a SQL table from the server, it just needs to be finalized and tested.
    - The intent of this is to import server data (which will already be sorted, compiled, and scored) to the front end for viewing.
* In order to run view the web page, simply navigate to:
  + \CYOPSE Web\html5up-prologue
  + Click index.html to open the web page
  + NOTE: This must be run a computer that can communicate with the server machine. For testing purposes, run the web interface on the same machine that the server software is installed on.
* The contents of the web page are very similar to the mockup, but have been since modified to be more simple. We also used a license free template to start off our design (html prologue).

**CyOpSE Server:**

* Software:
  + Python 3.5
    - Socket Server
    - PyMySQL
    - RE
  + MySQL
    - XAMPP Server
    - MySQL workbench
* To run the server, double click on the CyOpSE\_Server.py file
* Additionally, run CyOpSE\_Client.py on the raspberry pi probe to send data to the server.
* To run the MySQL components, open the XAMPP server first, then start the MySQL piece.
* Next, open MySQL workbench and open Database\_creator.sql
  + Once open, click the lightning bolt to run the database
    - Once the database is running, all stored data will be deleted and the database will be reset and ready to receive input from the client.
    - To add demo values in case the client is non-operational, run the demo\_populator.sql file to fill test values.

**CyOpSE Clients (Bro nodes):**

CyOpSE

VM / Raspberry Pi

user: cyopse

pass: capstone

Clone Into CyOpSE

sudo apt install git git clone https://github.com/WampaStompah/CyOpSE.git

Dependencies:

sudo apt-get install cmake make gcc g++ flex git bison libpcap-dev libssl-dev python-dev swig zlib1g-dev Install Bro Copy broctl to bin

Install Bro

Using http://knowm.org/how-to-install-bro-network-security-monitor-on-ubuntu/ sudo apt-get install cmake make gcc g++ flex bison libpcap-dev libgeoip-dev libssl-dev python-dev zlib1g-dev libmagic-dev swig libgoogle-perftools-dev

sudo mkdir -p /nsm/bro

cd ~

wget https://www.bro.org/downloads/bro-2.5.tar.gz

tar -xvzf bro-2.5.tar.gz

cd bro-2.5

./configure --prefix=/nsm/bro

make

sudo make install

export PATH=/nsm/bro/bin:$PATH

Modify Nodes

sudo nano /nsm/bro/etc/node.cfg

Modify IP range

sudo nano /nsm/bro/etc/networks.cfg

Modify MailTo

sudo nano /nsm/bro/etc/broctl.cfg

Install Broctl

sudo /nsm/bro/bin/broctl install exit

Install Oracle Java 8

Using http://knowm.org/how-to-set-up-the-elk-stack-elasticsearch-logstash-and-kibana/ sudo add-apt-repository -y ppa:webupd8team/java sudo apt-get update echo debconf shared/accepted-oracle-license-v1-1 select true | sudo debconf-set-selections echo debconf shared/accepted-oracle-license-v1-1 seen true | sudo debconf-set-selections sudo apt-get -y install oracle-java8-installer java -version

Install Elasticsearch

cd /var/cache/apt/archives sudo wget https://download.elastic.co/elasticsearch/release/org/elasticsearch/distribution/deb/elasticsearch/2.3.2/elasticsearch-2.3.2.deb sudo dpkg -i elasticsearch-2.3.2.deb

sudo update-rc.d elasticsearch defaults 95 10 sudo /etc/init.d/elasticsearch restart OR sudo systemctl enable elasticsearch sudo systemctl restart elasticsearch

cd /etc/elasticsearch sudo nano /etc/elasticsearch/elasticsearch.yml cd /var/log/elasticsearch

Note: If you want to access your Elasticsearch instance from clients on a different IP address via Javascript, add the following inside elasticsearch.yml: http.cors.enabled: true http.cors.allow-origin: "\*"

Also note that if you want to access Elasticsearch of any of the plugins like kopf from a host besides local host, you’ll need to add the following to elasticsearch.yml: network.bind\_host: 0

Kopf Plugin

sudo /usr/share/elasticsearch/bin/plugin install lmenezes/elasticsearch-kopf/v2.1.2

Integrate Bro IDS with ELK Stack

Using http://knowm.org/integrate-bro-ids-with-elk-stack/ cd /etc/logstash/conf.d/ sudo wget -N https://raw.githubusercontent.com/timmolter/logstash-dfir/master/conf\_files/bro/bro-conn\_log.conf sudo wget -N https://raw.githubusercontent.com/timmolter/logstash-dfir/master/conf\_files/bro/bro-dns\_log.conf sudo wget -N https://raw.githubusercontent.com/timmolter/logstash-dfir/master/conf\_files/bro/bro-files\_log.conf sudo wget -N https://raw.githubusercontent.com/timmolter/logstash-dfir/master/conf\_files/bro/bro-http\_log.conf sudo wget -N https://raw.githubusercontent.com/timmolter/logstash-dfir/master/conf\_files/bro/bro-notice\_log.conf sudo wget -N https://raw.githubusercontent.com/timmolter/logstash-dfir/master/conf\_files/bro/bro-ssh\_log.conf sudo wget -N https://raw.githubusercontent.com/timmolter/logstash-dfir/master/conf\_files/bro/bro-ssl\_log.conf sudo wget -N https://raw.githubusercontent.com/timmolter/logstash-dfir/master/conf\_files/bro/bro-weird\_log.conf sudo wget -N https://raw.githubusercontent.com/timmolter/logstash-dfir/master/conf\_files/bro/bro-x509\_log.conf

In the input section, we need to put all paths to the actual Bro log files on OUR system. In the output section at the end of the config file, we need to push the data to Elasticsearch: elasticsearch { host => localhost }. In the main filter section, a csv filter is assigned and configured for the bro log. You can hand write the csv filters if you want. The other filter sections do a few more manipulations to the data and are explained quite well in the comment sections. Starting Elasticsearch 2.0 it does not support field names with a . (or dot character) in them. Since the bro logs contain fields with dots in their names (id.orig\_p), we need to use a filter to convert the dots to underscores. If not you may see an error like: failed to put mappings on indices [[logstash-2016.05.02]], type [bro-conn\_log] MapperParsingException[Field name [id.orig\_h] cannot contain '.']. The mutate plugin is used to convert the field names containing dots to underscores with the rename command.

Install logstash-filter-translate

The above logstash config uses a plugin called logstash-filter-translate. The following terminal commands show how to install the logstash-filter-translate plugin. For a more in-depth explanation of installing logstash plugins see How to Install Logstash Plugins for Version 1.5.

cd /opt/logstash sudo bin/plugin install logstash-filter-translate

sudo -u logstash /opt/logstash/bin/logstash agent -f /etc/logstash/conf.d --configtest

sudo -u logstash /opt/logstash/bin/logstash -f /etc/logstash/conf.d --debug

sudo /etc/init.d/logstash restart

Modify:

change deploy.sh to current user directories

Start on system Startup

sudo nano /etc/rc.local #add: /nsm/bro/bin/broctl start

Crontab for maintenance

crontab -e #add: 0-59/5 \* \* \* \* /nsm/bro/bin/broctl cron

Ensure Bro is running after restart

tail -f /nsm/bro/logs/current/conn.log