# Open Network Insight

*Solution Guide*

## Table of Contents

Guide Version 0.9

# Overview

## Purpose and Audience

The overview section provides an understanding of the capabilities and potential business value of the Open Network Insight solution. The intended audience is executives or other sponsors of security analytics and big data projects in the organization.

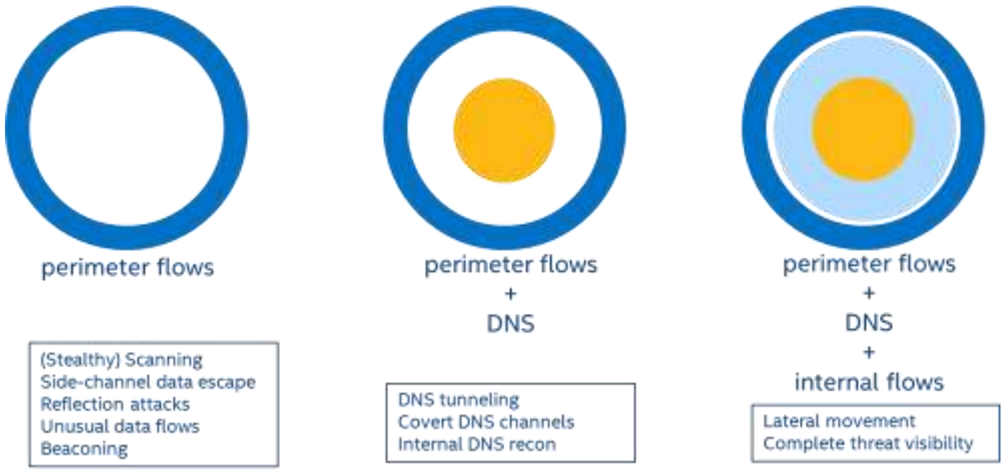
## The Business of Network Security – The "Port Perspective"

With the arrival of big data platforms, security organizations can now make data-driven decisions about how they protect their assets. Records of network traffic, captured as network flows, are often already stored and analyzed for use in network management. An organization can use this same information to gain insight into what channels corporate information flows through. By taking into account additional context such as prevalent attacks and protocols considered key to the company, the security organization can develop a strategy that applies the right amount of per-channel risk mitigation based on the value of the data flowing through it. For an organization, we call this "the port perspective". Two vectors that all organizations should evaluate:

1) A "wide enough, deep enough" protection strategy that involves both edge prevention and sophisticated detection of unusual behavior
2) Perform deep inspection of key protocols, using methods that can scale to the volume of data flowing across that channel.

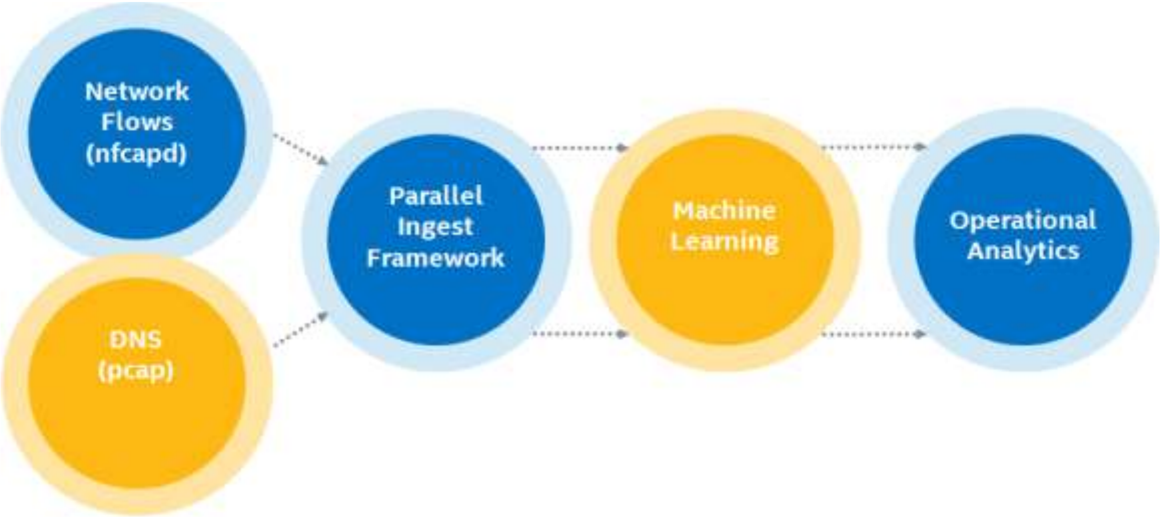
While inspecting specific, unique flows of data that may be important for individual organizations (i.e. order data or B2B communication on a specific port), all organizations can realize significant risk reduction from analysis of network flows for #1 and DNS (domain name service) replies for #2.

The Open Network Insight solution is intended to support this strategy by focusing on "hard security problems" – detecting events such as lateral movement, side-channel data escapes, insider issues, or stealthy behavior in general. It can be deployed incrementally to realize immediate ROI, but is also meant to support an organization's growth and maturity to achieve complete threat visibility as part of its protection strategy. The chart below compares a growth in investment in storage and compute to the level of detection that can be realized.



## Open Network Insight - Technical Overview

Open Network Insight is a solution built to leverage strong technology in both "big data" and scientific computing disciplines.  While the solution solves problems end-to-end, components may be leveraged individually or integrated into other solutions.  All components can output data in CSV format, maximizing interoperability.



**Parallel Ingest Framework**.  The system uses decoders, optimized from open source, that decode binary flow and packet data, then loading into HDFS and data structures inside Hadoop.  The decoded data is stored in multiple formats so it is available for searching, use by machine learning, transfer to law enformcement, or inputs to other systems.

**Machine Learning**.  The system uses a combination of Apache Spark and optimized C code to run scalable machine learning algorithms.  The machine learning component functions not only as a filter for separating bad traffic from benign, but also as a way to characterize the unique behavior of network traffic in an organization.

**Operational Analytics**.  In addition to machine learning, a proven process of context enrichment, noise filtering, whitelisting, and heuristics are applied to network data to produce a short list of the most likely patterns, which may be security threats.

## How to Use this Document

The rest of this document is composed of guides that cover each phase of the deployment:  Planning, Deployment, Installation, and Ongoing Use.  Each section (guide) is for use in that order.

# Planning Guide

## Purpose and Audience

The planning section provides an understanding of the capabilities and general design of the system. The audience is the anyone tasked with scoping out resources (hardware, software, people) required to install and operate the Open Network Insight solution.

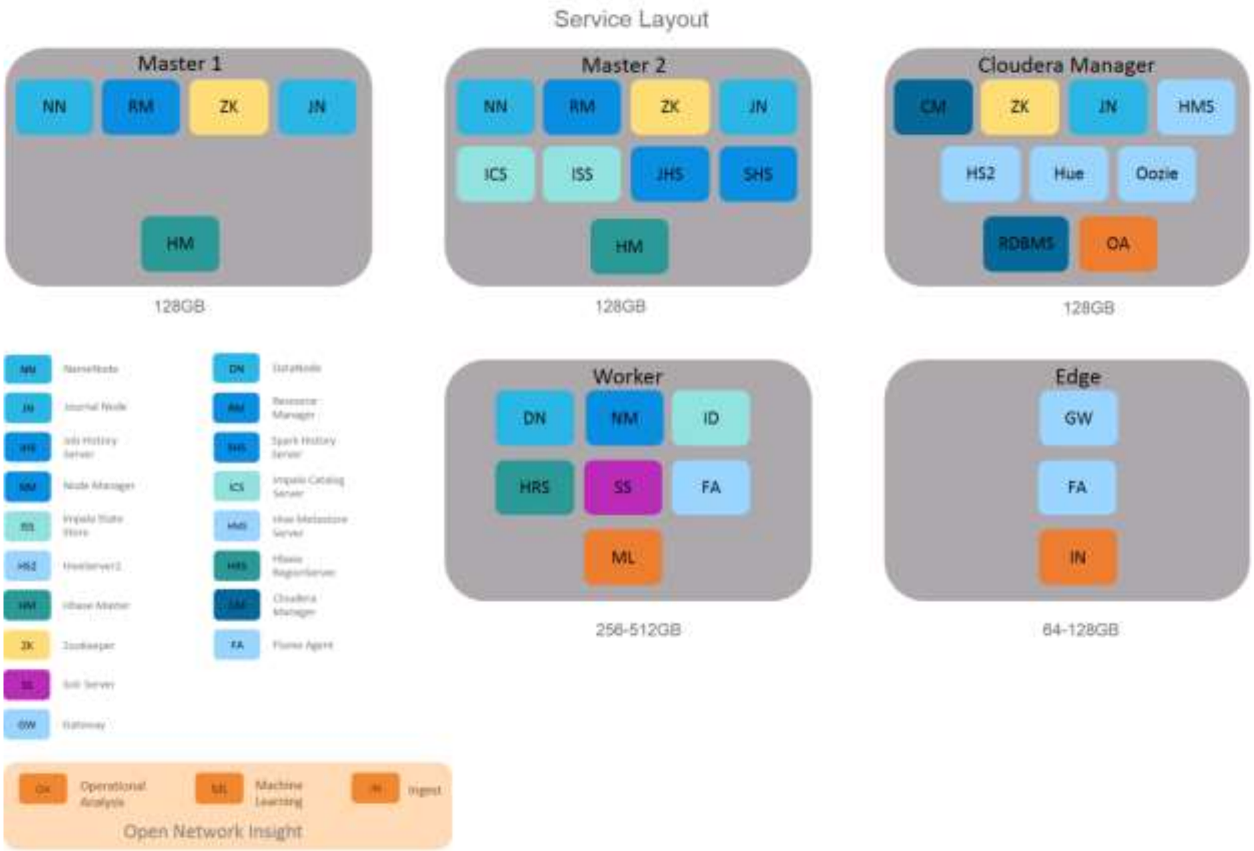
## Prerequisites

Cloudera Distribution of Hadoop

## Deployment Option 1: Pure Hadoop

Open Network Insight can be installed on a new or existing Hadoop cluster, its components viewed as services and distributed according to common roles in the cluster. One approach, based on the recommended deployment of CDH, is in the diagram below.

This approach is recommended for customers with a dedicated cluster for use of the solution or a security data lake; it takes advantage of existing investment in hardware and software. The disadvantage of this approach is that it does require the installation of software on Hadoop nodes not managed by systems like Cloudera Manager.

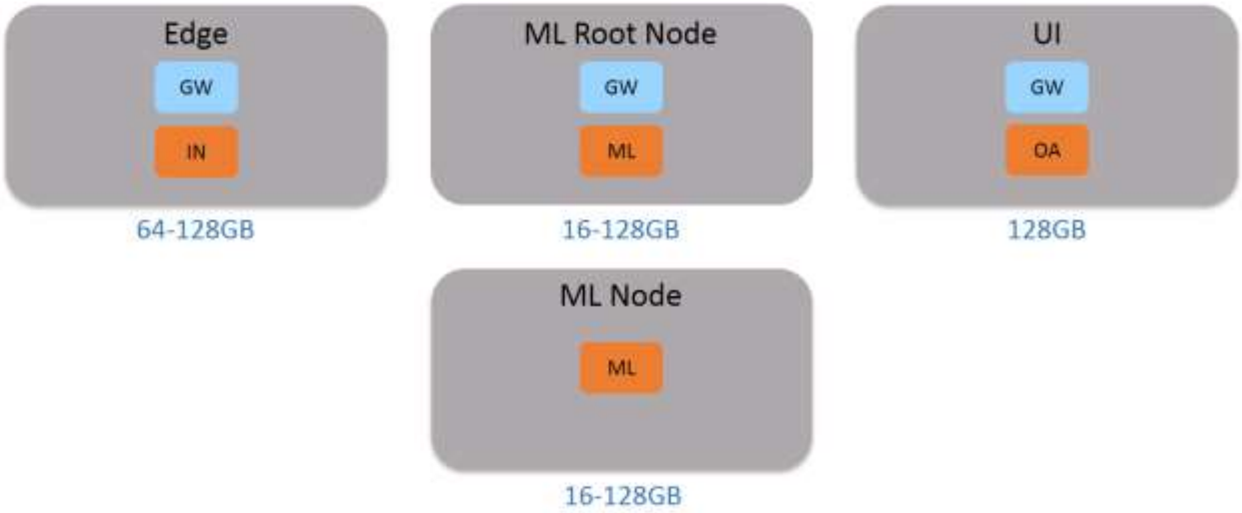


In the Pure Hadoop deployment scenario, the ingest component is run on an edge node, which is an expected use of this role. Installation of some non-Hadoop software is required for the ingest component. The Operational Analytics run on a node intended for browser-based management and user applications, so that all user interfaces are located on a node or nodes with the same role. The Machine Learning (ML) component installs on nodes in a worker role, as the resource management for an ML pipeline is similar for functions inside and outside Hadoop.

Although both of these deployment options are both validated and supported, additional scenarios that combine these approaches are certainly possible.

## Deployment Option 2: Hybrid Hadoop / Virtual

On existing Hadoop installations, a different approach involves using additional virtual machines and interacting with Hadoop components (Spark, HDFS) as a gateway node. This approach is recommended for customers with a Hadoop environment hosting heterogeneous use cases, where minimal deviation from node roles is desired. The disadvantage is that virtual machines must be sized appropriately according to workload.



In addition to the services deployed on the existing cluster, additional Virtual Machines (VM's) are required to host the non-Hadoop functions of the solution. The gateway service is required for some of these VM's to allow for interaction with Spark, Hive, and HDFS.

*Note: while the above is a recommended layout for production, pilot deployments may choose to combine the above roles into fewer VM's*. Each component of the Open Network Insight solution has integral interactions with Hadoop, but its non-Hadoop processing and memory requirements are separable with this approach.

# Deployment Guide

## Purpose and Audience

The deployment section provides a summarized view of the installation and recommended locations. The intended audience is the person responsible for leading the install.

## Component Distribution

There are three components to the Open Network Insight solution:

- Ingest – binary files are captured or transferred onto the Hadoop cluster, where they are transformed and loaded into solution data stores
- Machine Learning – machine learning algorithms are used to add additional learning information to the ingest data, which is used to filter and sort raw data.
- Operational Analytics – data output from the machine learning component that is augmented with context (example, geographic data) and heuristics, then available to the user for interacting with it

While in a development or test scenario all of the components can be installed on the same server, the recommended configuration in production is to map the components to specific server roles in a Hadoop cluster.

| **Component** | **Node / Key Role** |
|---|---|
| Ingest | Edge Server (Gateway) |
| Machine Learning | YARN Node Manager (Gateway) |
| Operational Analytics | Node with Cloudera Manager / Hue (Gateway) |

During the install, each of the three components installs in the `/home/<sol-user>/` folder in the appropriate node. This will require the creation of the solution user on each node.

### *Ingest*

Six subcomponents install on the edge server:

- nfdump (http://nfdump.sourceforge.net/): a set of utilities for capturing and decoding flow data
- tshark: (packet only) a CLI component of wireshark (https://www.wireshark.org/) for decoding packet data
- rabbitMQ – message queueing framework
- ingest workflow – bash script or Oozie workflow
- ingest master and workers – python code for data ingest
- ingest directory structure – local file system

There are also required changes to the Hadoop configuration:

- create HDFS path for binary data
- create HDFS path for Hive tables
- create solution Hive tables(staging, search)

### *Machine Learning (ML)*

There are multiple sub-components installed in each DataNode / NodeManager used for the solution:

- Scala scripts to run spark pre- and post-processing jobs
- Python scripts used for local transformation
- Algorithm code written in C/C++
- MPI (Message Passing Interface) libraries – used to parallelize algorithm code
- ML workflow – bash script
- ML directory structure on local file system

Some changes are required on the Hadoop Cluster as well:

- Spark configuration settings will need to be reviewed or modified
- YARN configuration settings will need to be reviewed or modified
- Directory structure for machine learning data

### *Operational Analytics (OA)*

Multiple subcomponents are required for installation on the Cloudera Manager/Hue server:

- Jupyter – provides a server for static html and JavaScript, as well as Jupyter notebooks, the key interface and the Hadoop cluster
- matplotlib (optional) – provides rich charting and plotting within Jupyter notebooks
- D3js and other JavaScript libraries – provide dynamic behavior and interactivity in the user interface
- Solution code – static html, JavaScript, and Jupyter notebooks used to access the operational analytics and information about the system
- Ops directory structure on the local file system

Some changes may be required on the Hadoop Cluster as well:

- YARN configuration settings will need to be reviewed or modified (for Hive query optimization)

Because the top-level components of the solution can be used independently or together, we recommend the following approach to installation.  For each component (ingest, machine learning, operational analytics):

- Identify deployment target nodes
- Install prerequisites on local file system
- Install solution component on local file system
- Make configuration/installation changes to Hadoop
- Validate and Test

# Installation Guide

## Purpose and Audience

The installation section contains the detailed steps to build, install, and configure the solution. The intended audience is the person or persons performing the steps in this section.

### *Step 0: Initial Configuration*

Before beginning the install, first identify the nodes that will participate in the solution. For a normal install, this will include an edge (gateway) node, a node to host the UI (operational analytics), and one or more nodes to host the machine learning component. You also need to identify a user account to run the solution with. The recommended approach is to use a non-administrator account, and either the same local account for all nodes or an account authenticated with Kerberos.

### *Configure User Accounts*

For each node that is part of the solution, create the solution user. This will also create the /home/<solution-user> directory. Set the same password for each account.

```
sudo adduser <solution-user>

passwd <solution-user>
```

For unattended execution of the ML pipeline, public key authentication will be required. Log on to the first ML node (usually the lowest-numbered node). On the first ML node, create a private key for the solution user. You will then need to copy those credentials to each node used for ML, starting with the first ML node.

```
[soluser@node04] ssh-keygen -t rsa

[soluser@node04] ssh-copy-id soluser@node04

[soluser@node04] ssh-copy-id soluser@node05
 …..:
[soluser@node04] ssh-copy-id soluser@node15
```

The sample above assumes that the <solution-user> is "soluser" and there are 12 nodes used for ML. Now do the same for the UI node.

```
[soluser@node04] ssh-copy-id soluser@node03
```

### *Edit Solution Configuration*

**Edit <solution>.conf** Copy the template config file to /etc and then edit it for your machine configuration. *We recommend only changing the HUSER, UINODE, LUSER, and NODES variables*.

```
[soluser@node-04]$ tar xvf oni-setup.tar.gz

[soluser@node-04]$ cd setup

[soluser@node-04 setup]$ sudo cp <solution>.conf /etc/.

[soluser@node-04 setup]$ sudo vim /etc/<solution>.conf
```

Once the file has been edited, copy it to the three nodes named as UINODE, MLNODE, and GWNODE in the config file.

Below is what the default configuration file looks like:

```
#node configuration
NODES=('node-01' 'node-02')
UINODE='node03'
MLNODE='node04'
GWNODE='node16'

#HDFS - base user and data source config
HUSER='/user/duxbury'
DSOURCES=('flow' 'dns')
DFOLDERS=('binary' 'csv' 'hive' 'stage')
DPATH=${HUSER}/${DSOURCE}/${DFOLDER}/y=${YR}/m=${MH}/d=${DY}
HPATH=${HUSER}/${DSOURCE}/${DFOLDER}/lda'${FDATE}
DBNAME=’duxbury’

KRB_AUTH=false
KINITPATH=
KINITOPTS=
KEYTABPATH=
KRB_USER=

#LOCAL FS base user and data source config
LUSER='/home/duxbury'
LPATH=${LUSER}/ml/${FDATE}
RPATH=${LUSER}/ipython/user/${FDATE}
LDAPATH=${LUSER}/ml/lda-c-parallel
LIPATH=${LUSER}/ingest

SPK_EXEC='400'
SPK_EXEC_MEM='2048m'
```

The following variables are needed throughout the ML pipeline:

HUSER – HDFS user path that will be the base path for the solution; this is usually the same user that you created to run the solution

DSOURCES – data sources enabled in this installation

DFOLDERS – built-in paths for the directory structure in HDFS

DPATH – the path to the flow records in Hive; this will be dynamically built within the pipeline with values for ${YR}, ${MH} and ${DY}

DBNAME – the name of the database used by the solution

LUSER – the local filesystem path for the solution, '/home/<solution-user>/'

LPATH – the local path for the ML intermediate and final results, dynamically built when the pipeline runs

RPATH – the path on the Operational Analytics node where the pipeline output will be delivered

SPK_EXEC – number of spark executors

SPK_EXEC_MEM – size (in MB) of spark executor

UINODE – the node that runs the Operational Analytics will run (aka, user interface node).

MLNODE- the node that runs the ML pipeline, controlling the other nodes.

GWNODE – the node that runs the ingest process.

NODES – a space delimited list of the Data Nodes that will run the C/MPI part of the pipeline. *Be very careful* to keep the variable in the format ('host1' 'host2' 'host3' …).

KRB_AUTH – (default: false) turn Kerberos authentication features on/off

KINITPATH=

KINITOPTS=

KEYTABPATH=

KRB_USER=

### Setup HDFS

Once the user has been created and local file system has been setup, run the hdfs_setup.sh script to set up the folder structures and hive tables

```
[soluser@node-04 setup]$ ./hdfs_setup.sh
```

## Step 1: Ingest Component Installation

### *Install Prerequisites*

Installing prerequisites should be done in a directory created under the /home/<solution-user>/ directory. It is recommended to create a temporary folder called "src" so that it can be deleted easily after successful validation.

```
mkdir src

cd src
```

First copy the modified nfdump source code and tshark source code to the /src directory. For nfdump, follow the steps below to build and install it.

```
tar –zxvf nfdump_duxburybay.tar.gz

cd nfdump_duxburybay

sudo ./install_nfdump.sh

cd ..
```

Install the prerequisites for the ingest queue (if the version in your yum repository is the same or later, you may also use yum to install a precompiled binary).

```
yum install –y python-pip

yum install –y watchdog


tar –zxvf pika-0.10.0b2.tar.gz
wget pika-0.10.0b2.tar.gz


cd pika-0.10.0b2

make install

wget erlang-17.4-1
rpm –i erlang-17.4-1.el6.x86_64.rpm
wget rabbitmq-server
rpm –i rabbitmq-server-3.5.3-1.noarch.rpm
```

For tshark, follow the steps on the web site to install it. Tshark must be downloaded and built from http://www.wireshark.orgWireshark

The screen utility is used to capture output from the ingest component for logging, troubleshooting, etc. You can check if screen is installed on the node.

```
which screen
```

If screen is not available, install it.

```
[soluser@edge-node]  sudo yum install screen
```

### *Install and Configure Component*

First copy the solution code (ingest.tar.gz) to the /home/<solution-user> directory.

```
tar -zxvf oni-ingest.tar.gz

cd ingest

...
```

Two configuration files must be edited:

- The master configuration (etc/master.json), which controls where and when the ingest component looks for new files
- The worker configuration (etc/worker.json), which tells ingest workers where to get file data and how to decode.

You only need to edit the configuration sections (flow, dns) for the data sources that the solution will use.

**Master Configuration**

Open the master_ingest.json file for editing.

```
cd etc

vi master_ingest.json
```

Here is a sample of the master_ingest.json  file, with an explanation of the configuration variables. *We recommend only changing the collector_path and pcap_split_staging variables*.

```
cat master_ingest.json
{
"dns":{

       "collector_path":"/mnt/sec_shared.nfs/dns",
       "pkt_num":"650000",
       "pcap_split_staging":"/mnt/sec_shared.nfs/dns",
       "time_to_wait":3600,
       "queue_name":"dns_ingest_queue"

},

"flow":{

       "collector_path":"/mnt/sec_shared.nfs/nfcapd",
        "queue_name":"flow_ingest_queue"

}
}
```

collector_path: this should be a path on the local file system where binary files are staged, from either the nfcapd service or as a staging area for pcap files

pkt_num: (packet only) number of packets per file (passed to editcap when splitting pcap files that are larger than 1GB)

pcap_split_staging: (packet only) this is where split files will be placed on the local file system before they are loaded into HDFS

time_to_wait: (packet only) the polling interval of the master ingest process on the local file system

queue_name: the name of the queue that will appear in the queue list

**Worker Configuration**

Open the worker_ingest.json file for editing.

```
vi worker_ingest.json
```

Here is a sample of the worker_ingest.json file, with an explanation of the configuration variables. *We recommend only changing the rabbitmq_server variable. Modifying the process_opt variable can require breaking changes to the solution code (skilled developers only!).*

```
cat worker_ingest.json
{
"rabbitmq_server":"10.10.10.10",
"dns":{
       "queue_name":"dns_ingest_queue",
       "process_opt":"-E  separator=,  -E  header=y  -T  fields  -e  frame.time  -e
frame.len   -e   ip.src   -e   ip.dst   -e   dns.resp.name   -e   dns.resp.type   -e
dns.resp.class -e dns.flags -e dns.flags.rcode -e dns.a 'dns.flags.response ==
1'"
},

"flow":{

       "queue_name":"flow_ingest_queue",
       "process_opt":""
}
}
```

rabbitmq_server: this is the IP address of the edge node used for the ingest component (in most cases, the master and workers are all on the same node).

process_opt: the flags passed to the binary decoders

queue_name:   the name of the queue that will appear in the queue list

[set up cron job (optional)]

### *Validate*

Start rabbitmq-server and validate that the service is running

```
service rabbitmq-server start
rabbitmqctl list_queues
```

Validate that nfdump is properly installed.

```
$ nfdump -V
nfdump: Version: 1.6.12 $Date: 2014-04-02 20:08:48 +0200 (Wed, 02 Apr 2014)
```

Validate that tshark is properly installed.

```
$ tshark -V
nfdump: Version: 1.6.12 $Date: 2014-04-02 20:08:48 +0200 (Wed, 02 Apr 2014)
```

Start the ingest framework, and validate that is running.

```
$ cd ..
$ ./start_ingest.sh flow 3
$ screen -ls
$ screen -x OniIngest_”ingest type”_”%H_%M_%S”
```

## Step 2: Machine Learning Component Installation

### *Install Prerequisite (MPI).*

Building or installing prerequisites should be done in a directory created under the /home/<solution-user>/ directory. It is recommended to create a temporary folder called “src” so that it can be deleted easily after successful validation. Building from source should be done on an edge node, **not a DataNode!**

```
mkdir src
cd src
```

A development version of MPI will be required on the machine where you build the source code; only a runtime engine is required on the nodes in the ML pipeline. The solution has been tested on MPICH 3.1.4 (http://www.mpich.org) and the Intel MPI Library 5.1 (https://software.intel.com/en-us/intel-mpi-library). Please refer to the specific release documentation for installation steps.

### *Install and Configure Solution*

On the edge server, copy the oni-ml.tar.gz to the /home/<solution-user>/ directory.

```
[soluser@edge-server]$ tar –zxvf oni-ml-tar.gz

cd ml/lda-c-parallel/

make clean

make
```

**Edit machinefile.** This file tells the MPI engine how many workers will be created and on which host.

```
[soluser@edge-server]$ vim /lda-c-parallel/machinefile
```

Modify the machine file to contain the exact same nodes that you used for the NODES environment variable, along with the number of workers. The file will have the same format (watch the ‘:’):

```
[soluser@edge-server]$ cat /lda-c-parallel/machinefile
Host1:5
Host2:5
Host3:5
Host4:5
```

*Note: If you are using the same configuration (4 hosts, 5 workers per host), you may skip the next steps and build the algorithm code*. After updating the machine file, you will need to edit the file lda-estimate.c, lines 190-191 to match the number of hosts and the number of workers per host. Using the example above, the values would be look like:

```
vi lda_estimate.c
…
int wkr = 4;
int wproc = 5;
…
```

You must also update the ml_ops.sh script with the total number of workers. For 4 hosts and 5 workers, the script would appear as it does below.

```
….
Time mpiexec –n 20
```

Build the algorithm code.

```
cd ml/lda-c-parallel/

[soluser@edge-server]$

make
```

Copy the entire ml folder to the primary ML node, the node that will launch the mpiexec command and do the local processing. For simplicity, this is often the lowest-numbered node in this role. Log in to the primary ML node.

```
[soluser@edge-server]$ scp -r ml node-04:/home/<soluser>/.
[soluser@edge-server]$ ssh node-04
[soluser@node-04]$cd /home/soluser/ml
```

The completed and configured ML pipeline needs to be copied to all the nodes. The script install_ml.sh does this with the help of the NODES variable

```
[soluser@node-04]$ ./install_ml.sh
```

[confirm spark settings]

[set up cron job (optional)]

### *Validate*

[check MPI version]

[run pipeline script]

The ML component can be tested by running the ml_ops.sh script with the following syntax:

```
[soluser@node-04]$ ./ml_ops.sh YYYYMMDD YYYY MM DD
```

## Step 3: Operational Analytics Component Installation:

### *Install Prerequisites*

Building or installing prerequisites should be done in a directory created under the /home/<solution-user>/ directory. It is recommended to create a temporary folder called “src” for easy deletion after successful validation.

```
mkdir src

cd src
```

Python 2.7 is required for the operational analytics component.

```
wget http://python.org/ftp/python/2.7.6/Python-2.7.6.tar.xz

tar xf Python-2.7.6.tar.xz

cd Python-2.7.6

make && make install

cd ..
```

Pip will be needed to install various python dependencies

```
wget --no-check-certificate https://bootstrap.pypa.io/get-pip.py

python2.7 get-pip.py
```

Install Jupyter to create the notebook server.

```
$ wget --no-check-certificate
https://pypi.python.org/packages/source/i/ipython/ipython-3.2.0.tar.gz#md5=41aa9b34f39484861e77c46ffb29b699

$ tar xvf ipython-3.2.0.tar.gz

$ cd ipython-3.2.0

$ sudo python2.7 setup.py install

$ cd ..
```

There are usually several python modules needed for the notebook server that are not part of a normal python install. Install these with pip.

```
 pip2.7 install pyzmq

 pip2.7 install jinja2

 pip2.7 install tornado

 pip2.7 install jsonschema

 cd ..
```

[matplotlib (optional)]

### *Install Solution*

On the UI node (similar to node with Cloudera Manager, Hue) copy the oni-oa.tar.gz to the /home/<solution-user>/ directory. Build the algorithm code.

```
[soluser@edge-server]$ tar –zxvf oni-oa.tar.gz
```

[set up cron job (optional)]

### *Validate*

[check ipython version]

[start ipython, open demo data link]

```
[soluser@edge-server]$ cd ipython
$ ./runIpython.sh
```

Open the link http://<server-ip>:8889/files/vast/index_sconnects.html

You should see the Suspicious Connects Analyst View, backed by the demo data.

[run pipeline script]

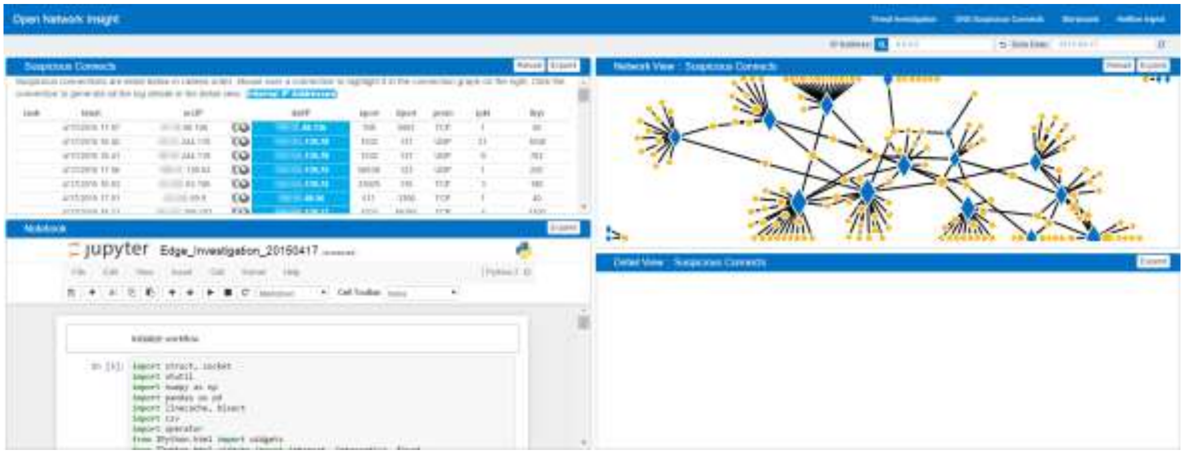
# User Guide

## Purpose and Audience

This section contains a walkthrough of the Suspicious Connects analyst view. The intended audience is the person or persons who will be reviewing the results for potential threats. Each section of this guide refers to a particular screen or view in the solution.

### *Suspicious Connects – Analyst View*

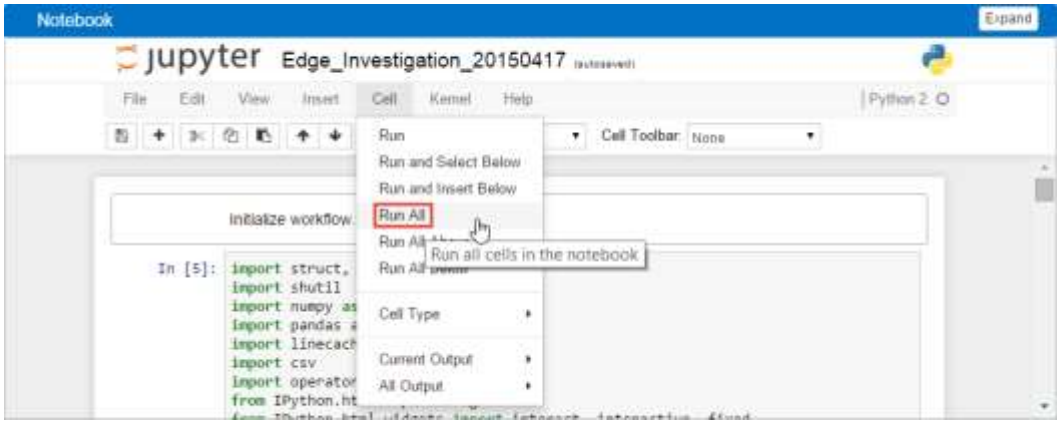
1. Log in to the analyst view for suspicious connects: http://<server-ip>:8889/files/index_sconnects.html. Select the date that you want to review. Your view should now look like this:


Open Network Insight
Suspicious Connects
Notebook
jupyter Edge_Investigation_20160417
Cell Toolbar
Network View : Suspicious Connects
Detail View : Suspicious Connects

```
import struct, socket
import shutil
import numpy as np
import pandas as pd
import linecache, bisect
import csv
import operator
from IPython.html import widgets
```

2. By clicking on a suspicious connect in the top left section, you will cause the corresponding edge to highlight in the network view (top right), and a log extract will appear in the detail view (lower right). If there is any additional traffic that is part of the communication it will appear along with the details of the suspicious connect. Your view should look similar to the one below.

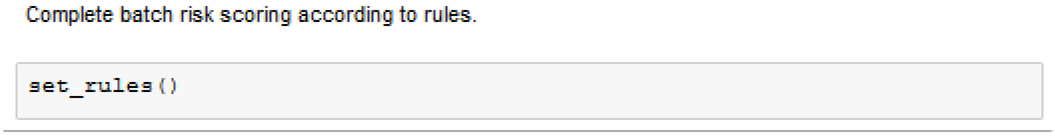
3. To make the best use of the analyst's time, the recommended first step is to do the auto-scoring steps, followed by examining the output of attack heuristics.
   a. First, you need to initialize and load the Jupyter notebook (lower left). The fastest way is to select Run All from the Cell menu. Note: this will also execute the auto-score and attack heuristics steps. If you want a finer grain of control, execute each step (including markdown cells) by clicking the play button or using the shortcut Shift+Enter.



Initialize workflow.

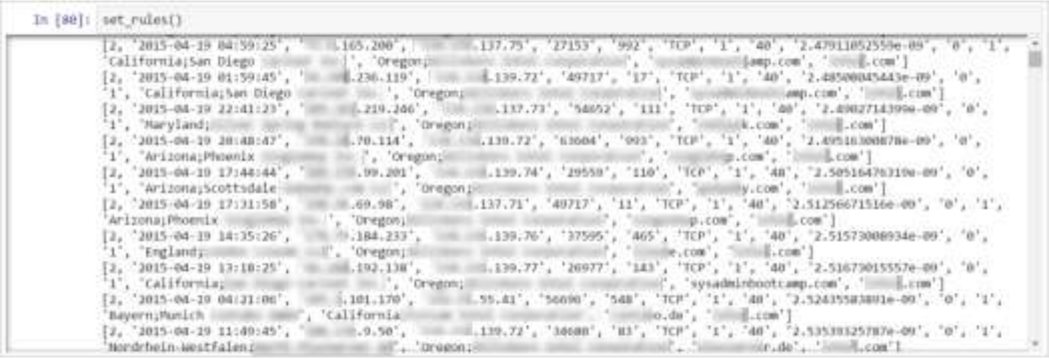
```
In [5]: import struct,
        import shutil
        import numpy as
        import pandas a
        import linecach
        import csv
        import operator
        from IPython.ht
```

b. Scroll down to the cell that contains the code set_rules. In the output window below, you will see a list of connections that were auto-scored based on general rules (example: inbound connections on low ports, with small packet sizes, are scored as a 2). Review these briefly to make sure they are good choices for auto-scoring.



Complete batch risk scoring according to rules.

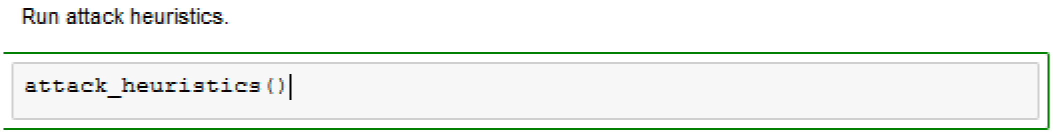
```
set_rules()
```



```
In [80]: set_rules()
[2, '2015-04-19 04:59:25', '[redacted].165.200', '[redacted].137.75', '27153', '992', 'TCP', '1', '40', '2.47911052559e-09', '0', '1',
'California;San Diego [redacted]', 'Oregon;[redacted]', '[redacted]amp.com', '[redacted].com']
[2, '2015-04-19 01:59:45', '[redacted].236.119', '[redacted].139.72', '49717', '17', 'TCP', '1', '40', '2.48500045443e-09', '0',
'1', 'California;San Diego [redacted]', 'Oregon;[redacted]', '[redacted]amp.com', '[redacted].com']
[2, '2015-04-19 22:41:23', '[redacted].219.246', '[redacted].137.73', '54652', '111', 'TCP', '1', '40', '2.49027143990e-09', '0',
'1', 'Maryland;[redacted]', 'Oregon;[redacted]', '[redacted]k.com', '[redacted].com']
[2, '2015-04-19 20:48:47', '[redacted].70.114', '[redacted].139.72', '63604', '993', 'TCP', '1', '40', '2.49518300078e-09', '0',
'1', 'Arizona;Phoenix [redacted]', 'Oregon;[redacted]', '[redacted].com', '[redacted].com']
[2, '2015-04-19 17:44:44', '[redacted].99.201', '[redacted].139.74', '29559', '110', 'TCP', '1', '40', '2.50516476310e-09', '0',
'1', 'Arizona;Scottsdale [redacted]', 'Oregon;[redacted]', '[redacted]y.com', '[redacted].com']
[2, '2015-04-19 17:31:58', '[redacted].69.98', '[redacted].137.71', '49717', '11', 'TCP', '1', '40', '2.51256671516e-09', '0', '1',
'Arizona;Phoenix [redacted]', 'Oregon;[redacted]', '[redacted]p.com', '[redacted].com']
[2, '2015-04-19 14:35:26', '[redacted].184.233', '[redacted].139.76', '37595', '465', 'TCP', '1', '40', '2.51573008934e-09', '0',
'1', 'England;[redacted]', 'Oregon;[redacted]', '[redacted]e.com', '[redacted].com']
[2, '2015-04-19 13:10:25', '[redacted].192.138', '[redacted].139.77', '26977', '143', 'TCP', '1', '40', '2.51673015557e-09', '0',
'1', 'California;[redacted]', 'Oregon;[redacted]', 'sysadminbootcamp.com', '[redacted].com']
[2, '2015-04-19 04:21:06', '[redacted].101.170', '[redacted].55.41', '56090', '548', 'TCP', '1', '40', '2.52435583801e-09', '0', '1',
'Bayern;Munich [redacted]', 'California[redacted]', '[redacted]o.de', '[redacted].com']
[2, '2015-04-19 11:49:45', '[redacted].9.50', '[redacted].139.72', '34080', '83', 'TCP', '1', '40', '2.53533325787e-09', '0', '1',
'Nordrhein-Westfalen;[redacted]', 'Oregon;[redacted]', '[redacted]r.de', '[redacted].com']
```

c. If you want to hide the output of this step, click inside the cell shown above. Click ESC to enter edit mode and type O to toggle the output

d. Scroll down to the cell that contains the code attack_heuristics(). In the output window below it, you will see a list of IP addresses with corresponding suspicious connections. These are patterns that the operational analytics have prioritized for review. Use the following step to review these patterns, particularly the path of "IP filtered review".



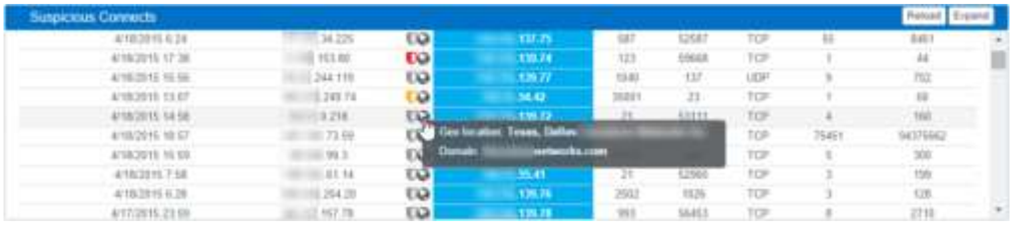
Run attack heuristics.

```
attack_heuristics()
```

e. A best practice to review the patterns found by attack heuristics:

i. Type (paste) an IP address from a pattern into the filter box (top right)

ii. Review relevant suspicious connects, log extracts, and geo IP info (The world icon will display the geolocation information) to make risk determination.

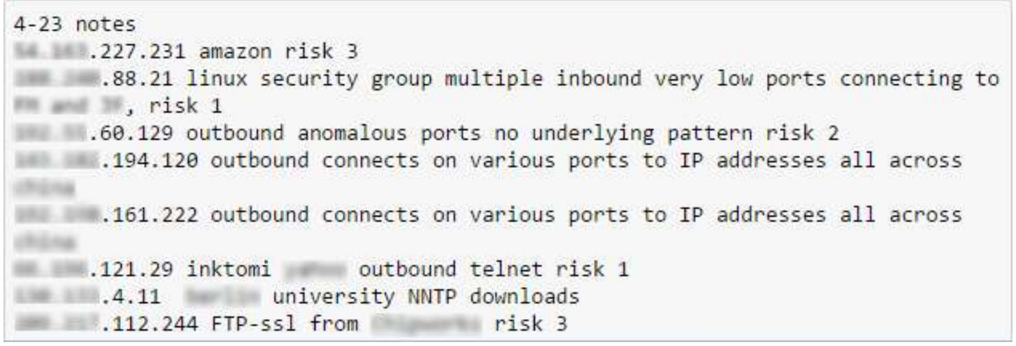


Suspicious Connects

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4/18/2015 6:24 | [redacted].34.225 | [redacted].137.75 | 587 | 52587 | TCP | 55 | 8461 |
| 4/18/2015 17:38 | [redacted].153.80 | [redacted].139.74 | 123 | 59668 | TCP | 1 | 44 |
| 4/16/2015 15:56 | [redacted].244.119 | [redacted].139.77 | 1940 | 137 | UDP | 9 | 702 |
| 4/18/2015 13:07 | [redacted].249.74 | [redacted].54.42 | 35891 | 23 | TCP | 1 | 68 |
| 4/18/2015 14:58 | [redacted].9.218 | [redacted].139.72 | 21 | 53111 | TCP | 4 | 160 |
| 4/18/2015 10:07 | [redacted].73.69 | | | | TCP | 75451 | 94375662 |
| 4/18/2015 16:59 | [redacted].99.3 | | | | TCP | 5 | 300 |
| 4/18/2015 7:58 | [redacted].61.14 | [redacted].35.41 | 21 | 52990 | TCP | 3 | 159 |
| 4/18/2015 6:28 | [redacted].254.20 | [redacted].139.76 | 2502 | 1926 | TCP | 3 | 128 |
| 4/17/2015 23:59 | [redacted].157.78 | [redacted].139.78 | 993 | 56463 | TCP | 8 | 2718 |

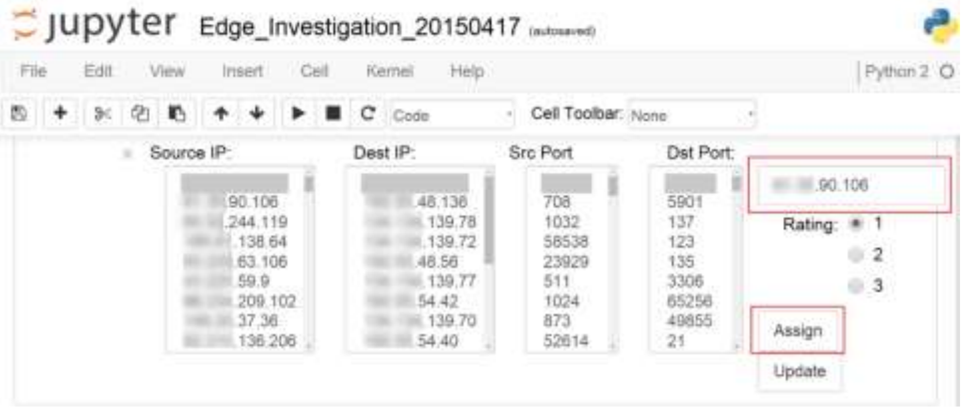
Geo location: Texas, Dallas [redacted]
Domain: [redacted]networks.com

iii. Make a note of your analysis in the notes section.



```
4-23 notes
[redacted].227.231 amazon risk 3
[redacted].88.21 linux security group multiple inbound very low ports connecting to
[redacted], risk 1
[redacted].60.129 outbound anomalous ports no underlying pattern risk 2
[redacted].194.120 outbound connects on various ports to IP addresses all across
[redacted]

[redacted].161.222 outbound connects on various ports to IP addresses all across
[redacted]

[redacted].121.29 inktomi [redacted] outbound telnet risk 1
[redacted].4.11  [redacted] university NNTP downloads
[redacted].112.244 FTP-ssl from [redacted] risk 3
```

iv. Multiple connections associated with the same pattern can be scored by typing (pasting) the IP address into the top right box above the risk scoring buttons.



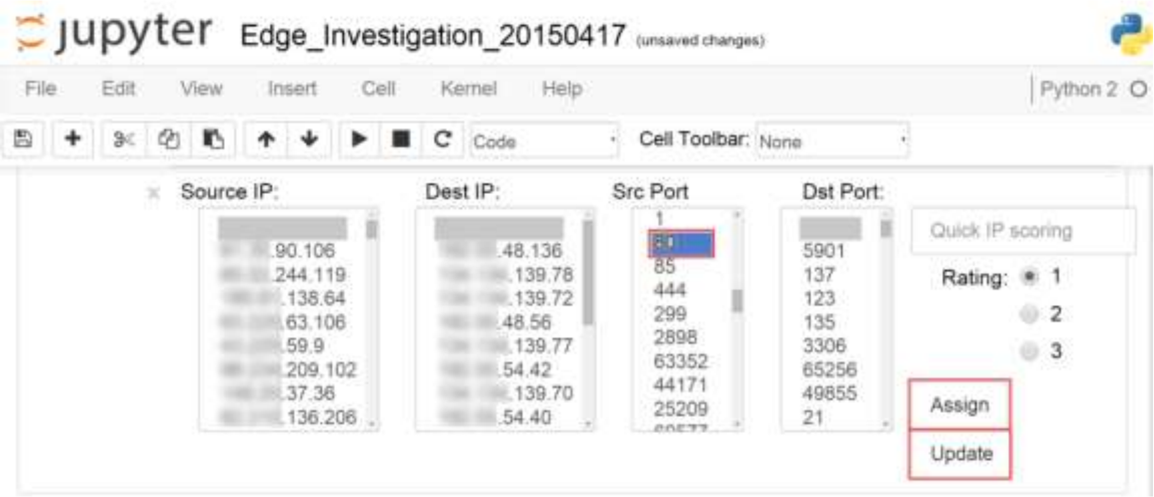
| Source IP: | Dest IP: | Src Port | Dst Port: |
|---|---|---|---|
| [redacted].90.106 | [redacted].48.136 | 708 | 5901 |
| [redacted].244.119 | [redacted].139.78 | 1032 | 137 |
| [redacted].138.64 | [redacted].139.72 | 58538 | 123 |
| [redacted].63.106 | [redacted].48.56 | 23929 | 135 |
| [redacted].59.9 | [redacted].139.77 | 511 | 3306 |
| [redacted].209.102 | [redacted].54.42 | 1024 | 65256 |
| [redacted].37.36 | [redacted].139.70 | 873 | 49855 |
| [redacted].136.206 | [redacted].54.40 | 52614 | 21 |

[redacted].90.106

Rating: ● 1 ○ 2 ○ 3

v. Click on the Assign button to Rate the connection.

4. In this way, the analyst can continue to score remaining connections. In order to assign a risk, select the specific connection using a combination of all the combo boxes, select the correct risk rating (1=true risk, 2 = potential risk, 3 = accepted risk). If the analyst wishes to score all connections with the same attribute (i.e., src port 80), then select only the combo boxes that are relevant and leave the rest at the empty slot at the top.



Source IP:
[redacted]
[redacted].90.106
[redacted].244.119
[redacted].138.64
[redacted].63.106
[redacted].59.9
[redacted].209.102
[redacted].37.36
[redacted].136.206

Dest IP:
[redacted]
[redacted].48.136
[redacted].139.78
[redacted].139.72
[redacted].48.56
[redacted].139.77
[redacted].54.42
[redacted].139.70
[redacted].54.40

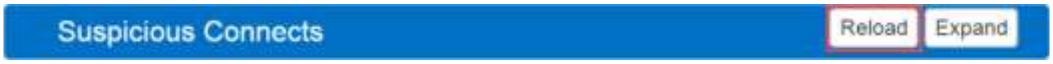
Src Port
1
80
85
444
299
2898
63352
44171
25209

Dst Port:
[redacted]
5901
137
123
135
3306
65256
49855
21

Rating: ● 1 ○ 2 ○ 3

5. Click on the Assign button to Rate the connection.

Note: If you click on the Update button the connections that you already rate will disappear on the suspicious connects page. Please use the Reload button on the suspicious connects page after using Update button.

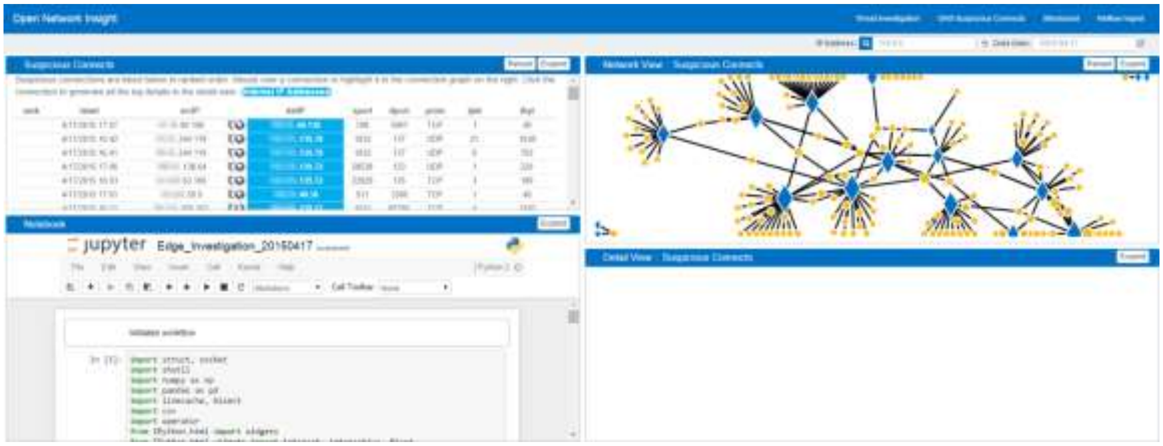


## *Threat Investigation Analyst View*

### Purpose and Audience

This section contains a walkthrough of the Threat Investigation analyst view. The intended audience is the person or persons who will be reviewing the results for potential threats.

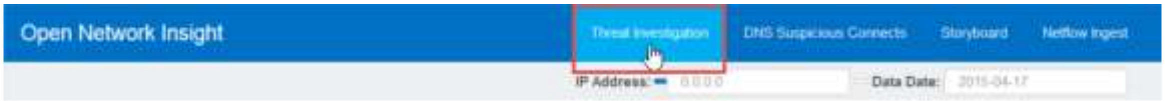
1. Log in to the analyst view for suspicious connects: http://servername:8889/files/index_sconnects.html. Select the date that you want to review. Your view should now look like this:


Open Network Insight
Suspicious Connects
Network View - Suspicious Connects
Notebook
Detail View - Suspicious Connects
jupyter Edge_Investigation_20150417
Python 2
Cell Toolbar: None

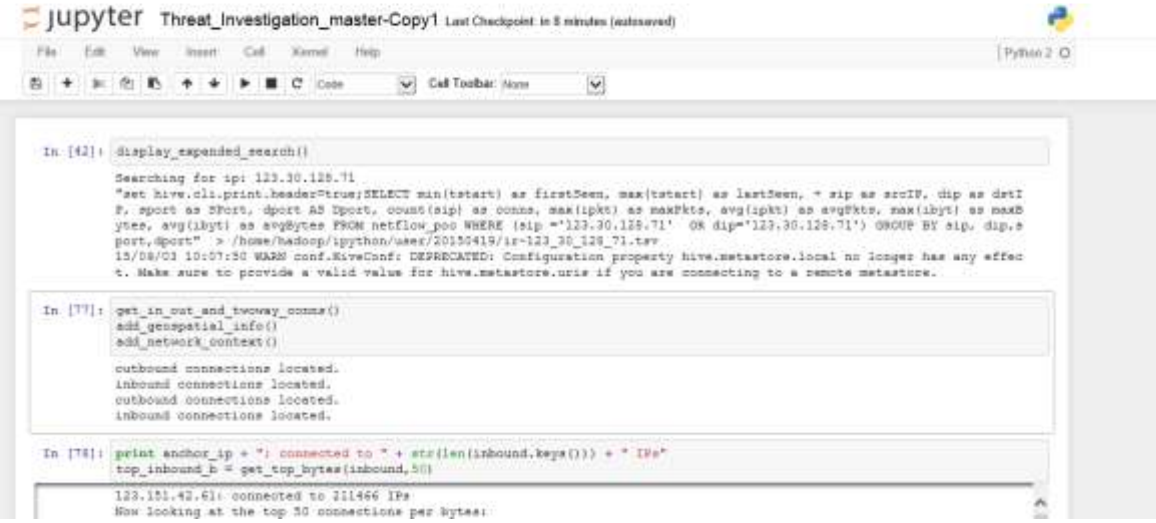
```
In [1]: import struct, socket
        import shutil
        import numpy as np
        import pandas as pd
        import linecache, bisect
        import csv
        import operator
        from IPython.html import widgets
```

   Note: The analyst must score the suspicious connections before moving into Threat Investigation View, please refer to Suspicious Connects Analyst View walkthrough

2. Select the option Threat Investigation from Open Network Insight Menu

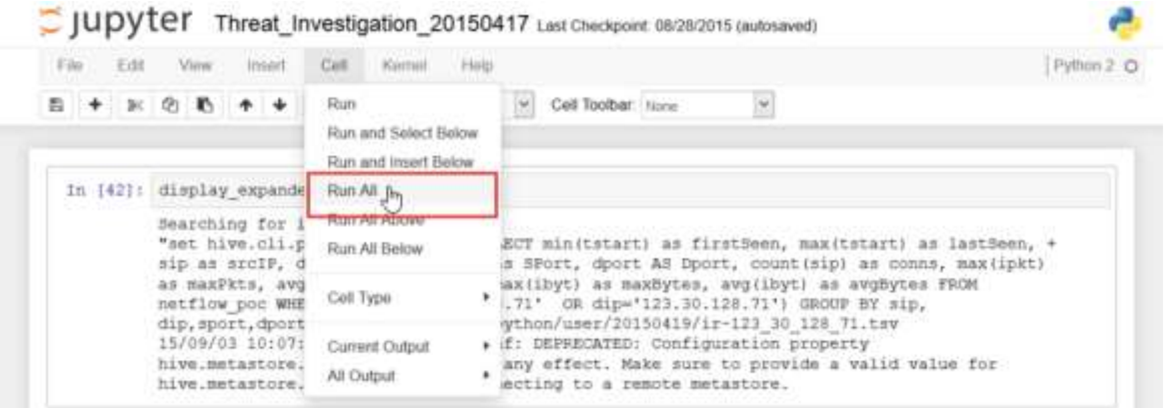


3. New web page will be opened Threat Investigation Jupyter Interface.


jupyter Threat_Investigation_master-Copy1 Last Checkpoint: in 8 minutes (autosaved)
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar: None

```
In [42]: display_expended_search()

         Searching for ip: 123.30.128.71
         "set hive.cli.print.header=true;SELECT min(tstart) as firstSeen, max(tstart) as lastSeen, + sip as srcIP, dip as dstI
         P, sport as SPort, dport AS Dport, count(sip) as conns, max(ipkt) as maxPkts, avg(ipkt) as avgPkts, max(ibyt) as maxB
         ytes, avg(ibyt) as avgBytes FROM netflow_poo WHERE (sip ='123.30.128.71'  OR dip='123.30.128.71') GROUP BY sip, dip,s
         port,dport"  > /home/hadoop/ipython/user/20150419/ir-123_30_128_71.tsv
         15/08/03 10:07:50 WARN conf.HiveConf: DEPRECATED: Configuration property hive.metastore.local no longer has any effec
         t. Make sure to provide a valid value for hive.metastore.uris if you are connecting to a remote metastore.

In [77]: get_in_out_and_twoway_conns()
         add_geospatial_info()
         add_network_context()

         outbound connections located.
         inbound connections located.
         outbound connections located.
         inbound connections located.

In [78]: print anchor_ip + ": connected to " + str(len(inbound.keys())) + " IPs"
         top_inbound_b = get_top_bytes(inbound,50)

         123.151.42.61: connected to 211466 IPs
         Now looking at the top 50 connections per bytes:
```

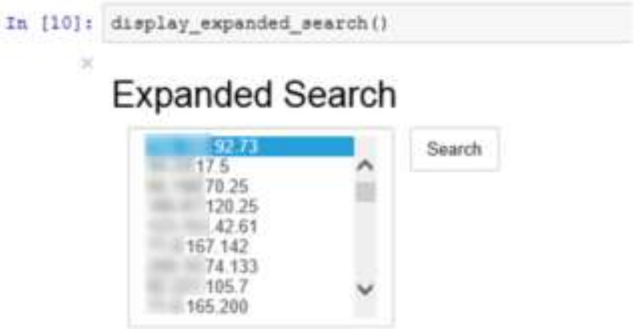
4. Initialize and load Jupyter notebook. The fastest way is to select Run All from the Cell menu.



```
In [42]: display_expande
         Searching for i
         "set hive.cli.p                      ECT min(tstart) as firstSeen, max(tstart) as lastSeen, +
         sip as srcIP, d                      s SPort, dport AS Dport, count(sip) as conns, max(ipkt)
         as maxPkts, avg                      ax(ibyt) as maxBytes, avg(ibyt) as avgBytes FROM
         netflow_poc WHE                      .71'  OR dip='123.30.128.71') GROUP BY sip,
         dip,sport,dport                      ython/user/20150419/ir-123_30_128_71.tsv
         15/09/03 10:07:                      f: DEPRECATED: Configuration property
         hive.metastore.                      any effect. Make sure to provide a valid value for
         hive.metastore.                      ecting to a remote metastore.
```

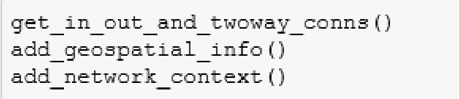
5. Each code section generates information that will be used in the Story Board Module. The following section represents the code section description and the outputs generated:

   a. In this module we can search the IP address(es) that was categorized as High risk connections in the Suspicious Connects Analyst View. Click on the IP address that you are interested and want to include as part of the Story Board module. Click Search button after selection.



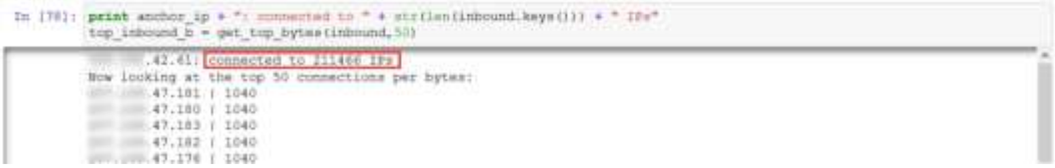
```
In [10]: display_expanded_search()
```

   b. The next code section will generate: Inbound, Outbound, and 2Way Connections based on the flow information that resides in the cluster. The geospatial_info is the information extracted from geolocalization's database. The add_network_context is the Network context information previously uploaded.



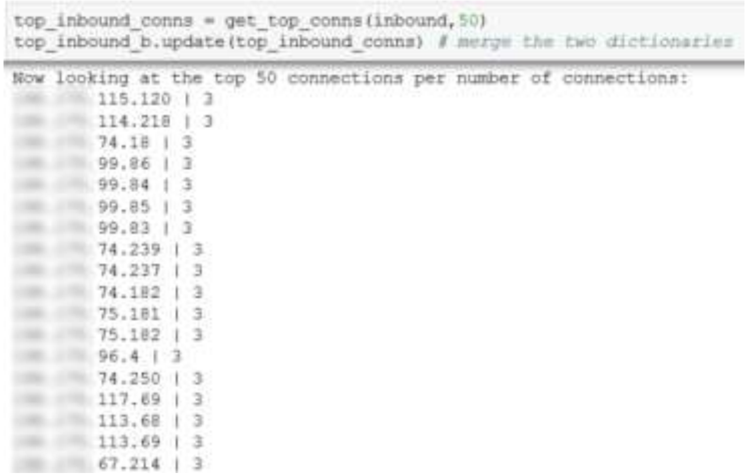
```
get_in_out_and_twoway_conns()
add_geospatial_info()
add_network_context()
```

   c. After performing the search for a single IP address, the quantity of different connections from that IP address will be displayed and the Top 50 connections (per bytes transfer).



```
In [78]: print anchor_ip + ": connected to " + str(len(inbound.keys())) + " IPs"
         top_inbound_b = get_top_bytes(inbound, 50)

         [redacted].42.61: connected to 211466 IPs
         Now looking at the top 50 connections per bytes:
         [redacted].47.181 | 1040
         [redacted].47.180 | 1040
         [redacted].47.183 | 1040
         [redacted].47.182 | 1040
         [redacted].47.176 | 1040
```

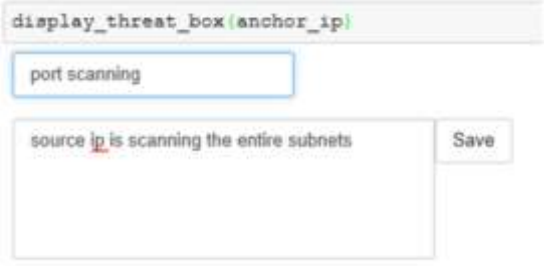
d. In addition, a list of the top 50 connections per number of connections that the external IP address established will be presented.



```
top_inbound_conns = get_top_conns(inbound,50)
top_inbound_b.update(top_inbound_conns) # merge the two dictionaries
```

```
Now looking at the top 50 connections per number of connections:
[redacted]115.120 | 3
[redacted]114.218 | 3
[redacted]74.18 | 3
[redacted]99.86 | 3
[redacted]99.84 | 3
[redacted]99.85 | 3
[redacted]99.83 | 3
[redacted]74.239 | 3
[redacted]74.237 | 3
[redacted]74.182 | 3
[redacted]75.181 | 3
[redacted]75.182 | 3
[redacted]96.4 | 3
[redacted]74.250 | 3
[redacted]117.69 | 3
[redacted]113.68 | 3
[redacted]113.69 | 3
[redacted]67.214 | 3
```

e. Threat Summary section. This code section allows the analyst to enter a Title & Description of the kind of attack/behavior described by the particular IP address that is under investigation. Click on the Save button after entering the data to write it into a CSV file, that eventually will be used in the Storyboard Analyst View.

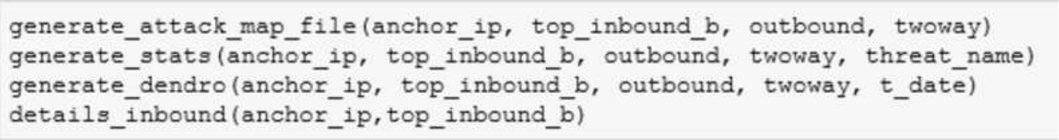


```
display_threat_box(anchor_ip)
```

port scanning

source ip is scanning the entire subnets

f. The following section will generate CSV files / context information that populates the Story Board Analyst View.



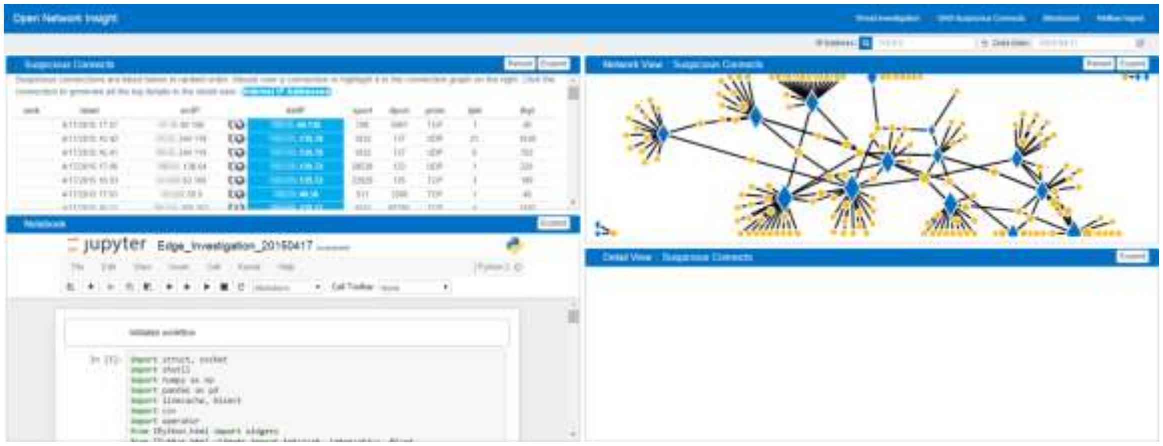
```
generate_attack_map_file(anchor_ip, top_inbound_b, outbound, twoway)
generate_stats(anchor_ip, top_inbound_b, outbound, twoway, threat_name)
generate_dendro(anchor_ip, top_inbound_b, outbound, twoway, t_date)
details_inbound(anchor_ip,top_inbound_b)
```

## *Ingest Summary Analyst View*

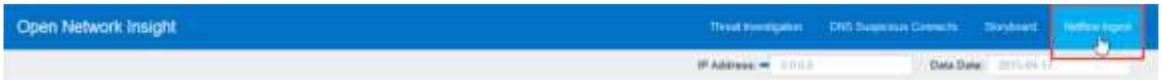
### Purpose and Audience

This section contains a walkthrough of the Threat Investigation analyst view. The intended audience is the person or persons who will be reviewing the results for potential threats.

1. Log in to the analyst view for suspicious connects: http://servername:8889/files/index_sconnects.html. Select the date that you want to review. Your view should now look like this:

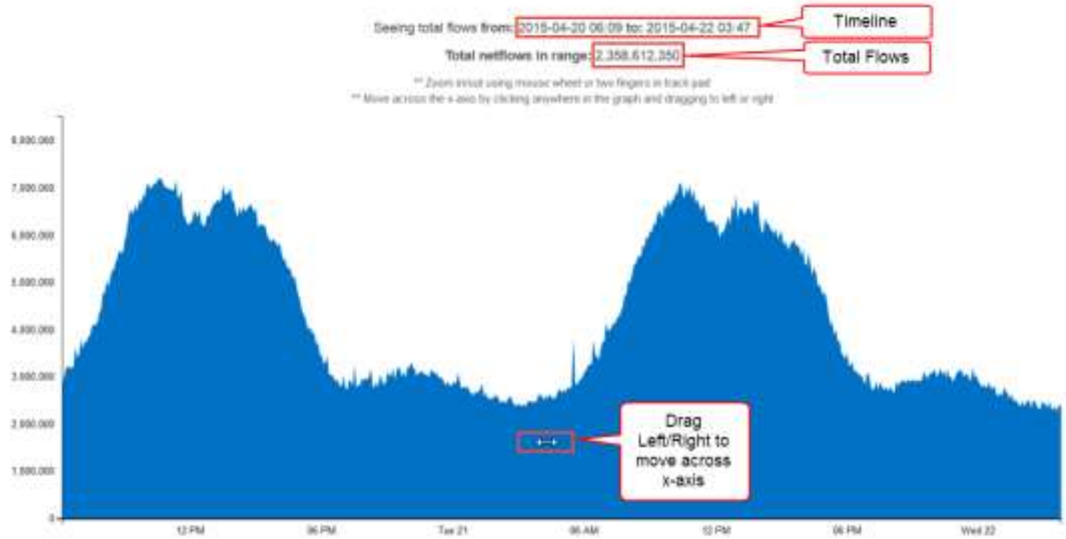

Open Network Insight
Suspicious Connects
Network View - Suspicious Connects
Notebook
jupyter Edge_Investigation_20150417
Detail View - Suspicious Connects

2. Select the option Netflow Ingest from Open Network Insight Menu.



3. Ingest Summary presents the Flows ingestion timeline, showing the Total Flows in range for a particular period of time.

Note: Analyst can zoom in/out on the graph. As well, move across the graph by dragging to left or right.


Seeing total flows from: 2015-04-20 06:09 to: 2015-04-22 03:47

Total netflows in range: 2,358,612,350

** Zoom in/out using mouse wheel or two fingers in track pad

** Move across the x-axis by clicking anywhere in the graph and dragging to left or right