

COURSE: Code Path Cybersecurity

WEEK(S): 10,11

PROJECT: Honeypot

AUTHOR: Rohit Chandra Tejaswi Kashibatla

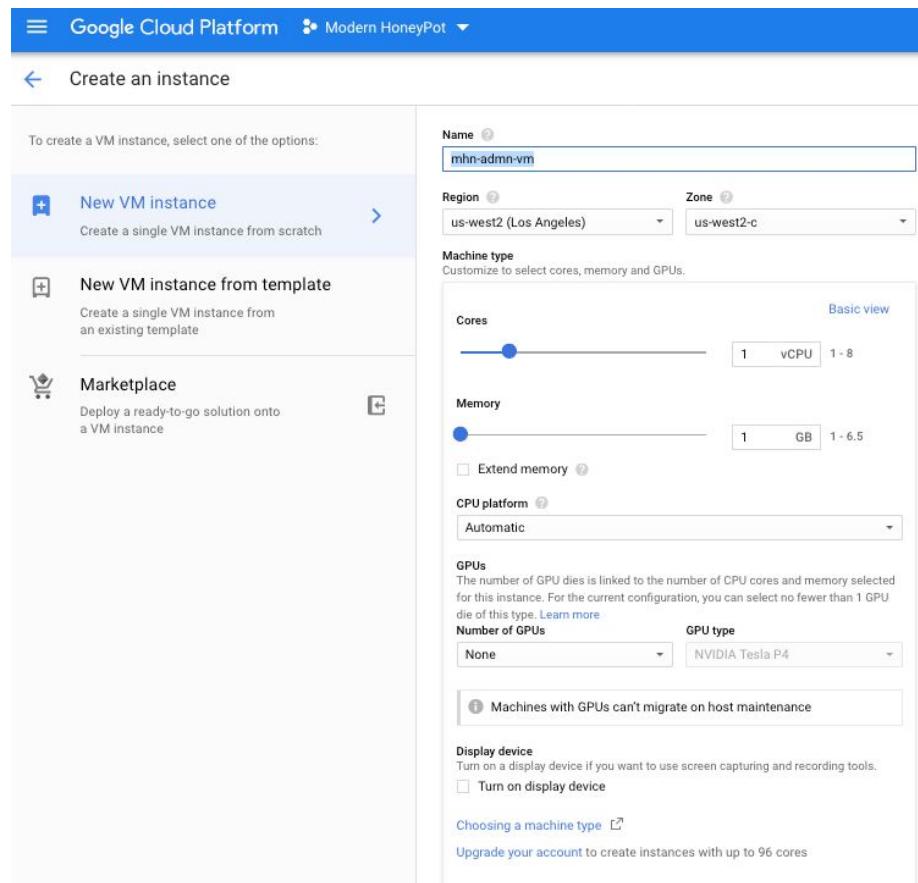
STUDENT ID: 012530830

EMAIL: rohitchandratejaswi.kashibatla@sjsu.edu

Cloud Hosting Provider: Google Cloud Platform

### MHN ADMIN VM

- 1) **Instance Creation:** Using Compute Engine Create an Ubuntu 14.04 VM Instance to deploy the MHN Admin VM.



**Boot disk**

New 10 GB standard persistent disk  
Image  
Ubuntu 14.04 LTS Change

**Identity and API access**

**Service account**   
Compute Engine default service account ▼

**Access scopes**   
 Allow default access  
 Allow full access to all Cloud APIs  
 Set access for each API

**Firewall**   
Add tags and firewall rules to allow specific network traffic from the Internet

Allow HTTP traffic  
 Allow HTTPS traffic

Management   Security   Disks   **Networking**   Sole Tenancy

**Network tags** (Optional)  
mhn-admn-vm

**Hostname**   
Set a custom hostname for this instance or leave it default  
mhn-admn-vm.us-west2-c.c.vertical-backup-240509.internal

**Network interfaces**   
default default (10.168.0.0/20)

Add network interface

To create another network interface you need to have a new network first.

Less

---

Your free trial credit will be used for this VM instance. [GCP Free Tier](#)

Create Cancel

VM instances

CREATE INSTANCE IMPORT VM REFRESH START STOP RESET DELETE

Filter VM instances

Name: mhn-admin-vm

Zone: us-west2-a

Internal IP: 10.168.0.3 (nic0)

External IP: 35.236.80.41

SSH

MHN Admin VM Successfully Created

2) **Create Firewall Rules:** Using the VPC Network's Firewall Rules create Firewall Rules to allow Ingress Traffic to the TCP Ports 80,3000, 10000 used by the Conpot Industrial Control System Honeypot.

mhn-admin-vm-allow-ingress-traffic-tcp-ports-3000-10000

Description  
Allow Ingress Traffic to mhn-admin-vm tcp ports 3000, 10000

Logs  
Off  
view

Network  
default

Priority  
1000

Direction  
Ingress

Action on match  
Allow

Targets

Target tags http-server

Source filters

IP ranges 0.0.0.0/0

Protocols and ports  
tcp:3000  
tcp:10000

Enforcement  
Enabled

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network
mhn-allow-any-traffic-dionaea-honeypot	Ingress	dionaea-honeypot, 3 more	IP ranges: 0.0.0.0/0	all	Allow	1	default
default-allow-http	Ingress	http-server	IP ranges: 0.0.0.0/0	tcp:80	Allow	1000	default
default-allow-https	Ingress	https-server	IP ranges: 0.0.0.0/0	tcp:443	Allow	1000	default
mhn-admin-vm-allow-ingress-traffic-tcp-ports-3000-10000	Ingress	http-server	IP ranges: 0.0.0.0/0	tcp:3000,10000	Allow	1000	default
default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	default
default-allow-internal	Ingress	Apply to all	IP ranges: 10.128.0.0/9, 10.168.0.0/24	tcp:0-65535 udp:0-65535 icmp	Allow	65534	default
default-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	65534	default

3) Basic NMAP Scan to test the Deployed Firewall Rule

```

rkashi@Rohits-MacBook-Pro:~$ nmap 35.236.80.41
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-14 22:55 PDT
Nmap scan report for 41.80.236.35.bc.googleusercontent.com (35.236.80.41)
Host is up (0.050s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
3000/tcp  open  ppp
10000/tcp open  snet-sensor-mgmt

Nmap done: 1 IP address (1 host up) scanned in 6.35 seconds

```

#### 4) Remote Administration of the MHN Admin VM using the SSH (Secure Shell)

**Protocol:** An SSH Public Key was added to the Project's Metadata Section.

Username	Key
mhn-admin	ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDA...iYX1vts40Nh8En5482cFM+liqVd1// mhn-admin

### Honeypot 1 Combo

Dionaea Honeypot with HTTP

Snort IDS Sensor

P0f TCP/IP Fingerprinting Sensor

- 1) **Instance Creation:** Using Compute Engine Create an Ubuntu 14.04 VM Instance

[Create an instance](#)

To create a VM instance, select one of the options:

[New VM instance](#)

Create a single VM instance from scratch

[New VM instance from template](#)

Create a single VM instance from an existing template

[Marketplace](#)

Deploy a ready-to-go solution onto a VM instance

Name [?](#)

mhn-ubuntu-1404-dionaea-honeypot-with-http

Region [?](#)

us-west2 (Los Angeles)

Zone [?](#)

us-west2-a

## Machine type

Customize to select cores, memory and GPUs.

## Cores

Basic view  
1 vCPU 1 - 8

## Memory

1 GB 1 - 6.5

 Extend memory [?](#)CPU platform [?](#)

Automatic

## GPUs

The number of GPU dies is linked to the number of CPU cores and memory selected for this instance. For the current configuration, you can select no fewer than 1 GPU die of this type. [Learn more](#)

## Number of GPUs

## GPU type

**⚠ GPUs aren't available in the zone you selected**

## Display device

Turn on a display device if you want to use screen capturing and recording tools.

 Turn on display device[Choosing a machine type ↗](#)[Upgrade your account](#) to create instances with up to 96 cores

### Boot disk ?



New 10 GB standard persistent disk

Image

Ubuntu 14.04 LTS

Change

### Identity and API access ?

#### Service account ?

Compute Engine default service account



#### Access scopes ?

- Allow default access
- Allow full access to all Cloud APIs
- Set access for each API

### Firewall ?

Add tags and firewall rules to allow specific network traffic from the Internet

- Allow HTTP traffic
- Allow HTTPS traffic

Management Security Disks **Networking** Sole Tenancy

**Network tags** [?](#) (Optional)

dionaea-honeypot [X](#)

**Hostname** [?](#)

Set a custom hostname for this instance or leave it default

mhn-ubuntu-1404-dionaea-honeypot-with-http.us-west2-a.c.vertical-backup-240!

**Network interfaces** [?](#)

default default (10.168.0.0/20) [Edit](#)

[+ Add network interface](#)

**i** To create another network interface you need to have a new network first.

[^ Less](#)

Your free trial credit will be used for this VM instance. [GCP Free Tier](#)

[Create](#)

[Cancel](#)

<input type="checkbox"/>	<input checked="" type="checkbox"/> mhn-ubuntu-1404-dionaea-honeypot-with- http	us- west2-a	10.168.0.7 (nic0)	35.235.87.133	SSH	
--------------------------	--	----------------	----------------------	---------------	-----	--

[Dionaea Honeypot with HTTP VM Successfully Created](#)

2) **Create A Firewall Rule:** Using the VPC Network's Firewall Rules create a Firewall Rule to allow any Ingress Traffic to Dionaea Honeypot with HTTP.

Google Cloud Platform Modern HoneyPot

VPC network

Firewall rules

Create a firewall rule

Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. [Learn more](#)

Name

Description (Optional)

Logs  
Turning on firewall logs can generate a large number of logs which can increase costs in Stackdriver. [Learn more](#)

On  
 Off

Network

Priority   
Priority can be 0 - 65535 [Check priority of other firewall rules](#)

Direction of traffic  Ingress  
 Egress

Action on match  Allow  
 Deny

Targets

Target tags

Source filter

Source IP ranges

Second source filter

Protocols and ports  Allow all  
 Specified protocols and ports

Disable rule

**3a) Deploy the Ubuntu Dionaea Script from the MHN Admin VM to the Honeypot VM:** The Conpot ICS Honeypot can be Installed by Executing the following command on the newly created “mhn-ubuntu-1404-dionaea-honeypot-with-http” virtual machine.

The screenshot shows the MHN Server interface at the URL 35.236.80.41/ui/manage-deploy/?script\_id=2. The top navigation bar includes links for MHN Server, Map, Deploy, Attacks, Payloads, Rules, Sensors, and Charts. The main area is titled "Select Script" with a dropdown menu showing "Ubuntu - Dionaea with HTTP". Below it is a "Deploy Command" section containing the following bash command:

```
wget "http://35.236.80.41/api/script/?text=true&script_id=2" -O deploy.sh && sudo bash deploy.sh  
http://35.236.80.41 CgwZ2d70
```

Under the "Deploy Script" section, there is a "Name" field containing "Ubuntu - Dionaea with HTTP". The "Script" section displays the contents of the "deploy.sh" file:

```
#!/bin/bash  
  
set -e  
set -x  
  
if [ $# -ne 2 ]  
then  
    echo "Wrong number of arguments supplied."  
    echo "Usage: $0 <server_url> <deploy_key>."  
    exit 1  
fi  
  
server_url=$1  
deploy_key=$2  
  
wget $server_url/static/registration.txt -O registration.sh  
chmod 755 registration.sh  
# Note: this will export the _HF_* variables  
. ./registration.sh $server_url $deploy_key "dionaea"  
  
# Add ppa to apt sources (Needed for Dionaea).  
apt-get update  
apt-get install -y python-software-properties  
add-apt-repository -y ppa:honeysoft/nightly  
apt-get update
```

The "Notes" section contains the text: "Initial deploy script for Ubuntu - Dionaea with HTTP".

**Command to Deploy the Dionaea with HTTP Sensor:**

```
 wget "http://35.236.80.41/api/script/?text=true&script_id=2" -O deploy.sh && sudo bash deploy.sh  
http://35.236.80.41 CgwZ2d70
```

```
mhn-admin@mhn-ubuntu-1404-dionaea-honeypot-with-http:~$ wget "http://35.236.80.41/api/script/?text=true&script_id=2" -O deploy.sh && sudo bash deploy.sh http://35.236.80.41 CgwZ2d70
```

**dionaea: added process group**

```
mhn-admin@mhn-ubuntu-1404-dionaea-honeypot-with-http:~$ ps -ef | grep "dionaea"
mhn-admin 17186 17165 0 06:14 pts/0    00:00:00 grep --color=auto dionaea
nobody    22429  2292 0 01:18 ?        00:00:11 dionaea -c /etc/dionaea/dionaea.conf -w /var/dionaea -u nobody -g nogroup
root     22430 22429 0 01:18 ?        00:00:00 dionaea -c /etc/dionaea/dionaea.conf -w /var/dionaea -u nobody -g nogroup
```

Additionally the Ubuntu Snort IDS Sensor was deployed from the MHN Admin VM to the Honeypot VM

3b) Deploy the Ubuntu Snort IDS Sensor from the MHN Admin VM to the Honeypot VM: The Snort IDS Sensor can be Installed by Executing the following command on the newly created “mhn-ubuntu-1404-dionaea-honeypot-with-http” virtual machine.

The screenshot shows the MHN Server interface at the URL 35.236.80.41/ui/manage-deploy/?script\_id=6. The top navigation bar includes links for MHN Server, Map, Deploy, Attacks, Payloads, Rules, Sensors, and Charts. The main area is titled "Select Script" with a dropdown menu containing "Ubuntu - Snort". Below this is the "Deploy Command" section, which contains the command: wget "http://35.235.87.133/api/script/?text=true&script\_id=6" -O deploy.sh && sudo bash deploy.sh followed by the URL http://35.235.87.133 CgwZ2d70. Further down, there is a "Deploy Script" section with a "Name" field containing "Ubuntu - Snort".

## Deploy Script

### Name

```
Ubuntu - Snort
```

### Script

```
#!/bin/bash

INTERFACE=eth0

set -e
set -x

if [ $# -ne 2 ]
then
    echo "Wrong number of arguments supplied."
    echo "Usage: $0 <server_url> <deploy_key>."
    exit 1
fi

server_url=$1
deploy_key=$2

apt-get update
DEBIAN_FRONTEND=noninteractive apt-get -y install build-essential libpcap-dev libjansson-dev
libpcre3-dev libdnet-dev libdumbnet-dev libdaq-dev flex bison python-pip git make automake libtool
zlib1g-dev

pip install --upgrade distribute
pip install virtualenv
```

### Notes

```
Initial deploy script for Ubuntu - Snort
```

#### Deployment Command:

```
wget "http://35.236.80.41/api/script/?text=true&script_id=6" -O deploy.sh && sudo bash deploy.sh
http://35.236.80.41 CgwZ2d70
```

```
mhn-admin@mhn-ubuntu-1404-dionaea-honeypot-with-http:~$ wget "http://35.236.80.41/api/script/?text=true&script_id=6" -O deploy.sh
&& sudo bash deploy.sh http://35.236.80.41 CgwZ2d70 ┌
```

```
mhn-admin@mhn-ubuntu-1404-dionaea-honeypot-with-http:~$ ps -efaf | grep "snort"
root      15638  2292  90  04:24 ?          00:00:20 /opt/snort/bin/snort -c /opt/snort/etc/snort.conf -i eth0
```

Snort IDS Sensor Successfully Deployed

Additionally the Ubuntu p0f Sensor was deployed rom the MHN Admin VM to the Honeypot VM

3c) Deploy the Ubuntu p0f Sensor from the MHN Admin VM to the Honeypot VM: The p0f Sensor can be Installed by Executing the following command on the newly created "mhn-ubuntu-1404-dionaea-honeypot-with-http" virtual machine.

MHN Server   Map   Deploy   Attacks   Payloads   Rules   Sensors   Charts

Select Script  
Ubuntu - p0f

Deploy Command  

```
wget "http://35.235.87.133/api/script/?text=true&script_id=11" -O deploy.sh && sudo bash deploy.sh
http://35.235.87.133 CgwZ2d70
```

---

Deploy Script  
Name  
Ubuntu - p0f

Script  

```
#!/bin/bash

set -e
set -x

if [ $# -ne 2 ]
then
    echo "Wrong number of arguments supplied."
    echo "Usage: $0 <server_url> <deploy_key>."
    exit 1
fi

server_url=$1
deploy_key=$2

apt-get update
apt-get -y install git supervisor libpcap-dev libjansson-dev gcc

# install p0f
cd /opt
git clone https://github.com/threatstream/p0f.git
cd p0f
git checkout origin/hpfeeds
./build.sh
useradd -d /var/empty/p0f -M -r -s /bin/nologin p0f-user || true
```

Notes  
Initial deploy script for Ubuntu - p0f

#### Deployment Command:

```
wget "http://35.236.80.41/api/script/?text=true&script_id=11" -O deploy.sh
&& sudo bash deploy.sh http://35.236.80.41/ CgwZ2d70
mhn-admin@mhn-admin-vm:~$ wget "http://35.236.80.41/api/script/?text=true&script_id=11" -O deploy.sh
&& sudo bash deploy.sh http://35.236.80.41/ CgwZ2d70
```

+ supervisorctl update  
p0f: added process group

```
mhn-admin@mhn-admin-vm:~$ ps -ef | grep "p0f"
p0f-user 28594 1491 0 05:24 ? 00:00:00 /opt/p0f/p0f -i eth0 -u p0f-user -h 35.236.80.41 -P 10000 -I ed9f2186-779a-11e9-95b5-42010aa80003 -K 3ryNMF9sg7a4X5oX -C p0f.events !(src host 10.168.0.3)
mhn-admin 29028 29011 0 06:00 pts/0 00:00:00 grep --color=auto p0f
```

p0f Sensor Successfully Deployed

≡ Google Cloud Platform Modern HoneyPot ▾

VPC network Firewall rules + CREATE FIREWALL RULE REFRESH DELETE

VPC networks Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. [Learn more](#)

External IP addresses Note: App Engine firewalls are managed [here](#).

<input checked="" type="checkbox"/> mhn-allow-any-traffic-dionaea-honeypot	Ingress	dionaea-honeypot	IP ranges: 0.0.0.0/0	all	Allow	1000	default
--	---------	------------------	----------------------	-----	-------	------	---------

```
rkashi@Rohits-MacBook-Pro:~$ nmap 35.235.87.133
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-14 22:51 PDT
Nmap scan report for 133.87.235.35.bc.googleusercontent.com (35.235.87.133)
Host is up (0.038s latency).
Not shown: 987 closed ports
PORT      STATE    SERVICE
21/tcp     open     ftp
22/tcp     open     ssh
25/tcp     filtered smtp
42/tcp     open     nameserver
80/tcp     open     http
135/tcp    filtered msrpc
139/tcp    filtered netbios-ssn
443/tcp    open     https
445/tcp    filtered microsoft-ds
1433/tcp   open     ms-sql-s
3306/tcp   open     mysql
5060/tcp   open     sip
5061/tcp   open     sip-tls

Nmap done: 1 IP address (1 host up) scanned in 2.44 seconds
```

## Sensors

Name	Hostname	IP	Honeypot	UUID	Attacks
1-	mhn-ubuntu-1404-dionaea-h	35.235.87.133	dionaea	6f905700-76d3-11e9-95b5-42010aa80003	6345

## Attack Stats

Attacks in the last 24 hours: **7,276**

TOP 5 Attacker IPs:

1.  [38.141.45.50 \(2,359 attacks\)](#)
2.  [77.247.110.36 \(989 attacks\)](#)
3.  [111.230.73.43 \(615 attacks\)](#)
4.  [75.25.131.207 \(344 attacks\)](#)
5.  [193.188.22.115 \(199 attacks\)](#)

TOP 5 Attacked ports:

1. [5060 \(3,463 times\)](#)
2. [80 \(706 times\)](#)
3. [8088 \(497 times\)](#)
4. [3389 \(273 times\)](#)
5. [23 \(223 times\)](#)

TOP 5 Honey Pots:

1. [dionaea \(6,319 attacks\)](#)
2. [drupot \(626 attacks\)](#)
3. [snort \(186 attacks\)](#)
4. [p0f \(113 attacks\)](#)
5. [conpot \(32 attacks\)](#)

TOP 5 Sensors:

1. [mhn-ubuntu-1404-dionaea-honeypot-with-http \(6,319 attacks\)](#)
2. [mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot \(626 attacks\)](#)
3. [mhn-ubuntu-1404-dionaea-honeypot-with-http \(165 attacks\)](#)
4. [mhn-admin-vm \(71 attacks\)](#)
5. [mhn-ubuntu-1404-conpot-ics-honeypot \(41 attacks\)](#)

TOP 5 Attacks Signatures:

1. [ET SCAN SipCLI VOIP Scan \(89 times\)](#)
2. [ET DROP Dshield Block Listed Source group 1 \(45 times\)](#)
3. [ET CINS Active Threat Intelligence Poor Reputation IP TCP group 81 \(14 times\)](#)
4. [ET CINS Active Threat Intelligence Poor Reputation IP TCP group 80 \(9 times\)](#)
5. [ET CINS Active Threat Intelligence Poor Reputation IP TCP group 37 \(4 times\)](#)

## Honeypot 2 Combo

Conpot Industrial Control Systems Honeypot

Snort IDS Sensor

P0f TCP/IP Fingerprinting Sensor

- 1) **Instance Creation:** Using Compute Engine Create an Ubuntu 14.04 VM Instance

Name 

mhn-ubuntu-1404-conpot-ics-honeypot

Region 

us-west2 (Los Angeles)

Zone 

us-west2-c

Machine type

Customize to select cores, memory and GPUs.

[Basic view](#)

Cores



1 vCPU 1 - 8

Memory



1 GB 1 - 52

Extend memory 

Management Security Disks Networking Sole Tenancy

Network tags ? (Optional)

conpot-ics-honepot x

Hostname ?

Set a custom hostname for this instance or leave it default

mhn-ubuntu-1404-conpot-ics-honepot.us-west2-c.c.vertical-backup-240509.inte

Network interfaces ?

default default (10.168.0.0/20) edit

+ Add network interface

i To create another network interface you need to have a new network first.

^ Less

Your free trial credit will be used for this VM instance. [GCP Free Tier](#) ?

Create

Cancel

<input type="checkbox"/>	<span>✓</span> mhn-ubuntu-1404-conpot-ics-honepot	us-west2-c	10.168.0.8 (nic0)	35.236.28.214	SSH	<span>⋮</span>
--------------------------	---	------------	----------------------	---------------	-----	----------------

Conpot ICS Honepot VM Successfully Created

2) **Create A Firewall Rule:** Using the VPC Network's Firewall Rules create a Firewall Rule to allow Ingress Traffic to the TCP and UDP Ports used by the Conpot Industrial Control System Honeypot.

 <b>VPC network</b>   VPC networks  External IP addresses  <b>Firewall rules</b>  Routes  VPC network peering  Shared VPC  Serverless VPC access	<p> <b>Create a firewall rule</b></p> <p>Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. <a href="#">Learn more</a></p> <p><b>Name</b>  <input type="text" value="mhn-allow-restricted-ingress-traffic-conpot-ics-honeypot"/></p> <p><b>Description (Optional)</b> <input type="text" value="Allow Ingress Traffic to TCP &amp; UDP Port's used by the Conpot Industrial Control System Honeypot"/></p> <p><b>Logs</b> Turning on firewall logs can generate a large number of logs which can increase costs in Stackdriver. <a href="#">Learn more</a> <input checked="" type="radio"/> On <input type="radio"/> Off</p> <p><b>Network</b>  <input type="text" value="default"/></p> <p><b>Priority</b>  Priority can be 0 - 65535 <a href="#">Check priority of other firewall rules</a> <input type="text" value="1000"/></p> <p><b>Direction of traffic</b>  <input checked="" type="radio"/> Ingress <input type="radio"/> Egress</p> <p><b>Action on match</b>  <input checked="" type="radio"/> Allow <input type="radio"/> Deny</p>
--	--

## Targets

Specified target tags

## Target tags

conpot-ics-honepot 

## Source filter

IP ranges

## Source IP ranges

0.0.0.0/0 

## Second source filter

None

## Protocols and ports

Allow all

Specified protocols and ports

tcp :

21,80,102,502,2121,5020,8800,10201,20000,44818

udp :

69,161,623,6230,6969,16100,20000,47808

Other protocols

protocols, comma separated, e.g. ah, sctp

## Disable rule

**Create**

**Cancel**

<input checked="" type="checkbox"/> mhn-allow-restricted-ingress-traffic-conpot-ics-honeypot	ingress	conpot-ics-honeypot	IP ranges: 0.0.0.0/0	tcp:21,80,102,502,2121,5020,8800,10201,20000,44818 udp:69,161,623,6230,6969,16100,20000,47808	Allow	1000	default
--	---------	---------------------	----------------------	--	-------	------	---------

**Firewall Rule**

## References:

- + <https://buildmedia.readthedocs.org/media/pdf/conpot/latest/conpot.pdf>
- + <https://pythonhosted.org/Conpot/installation/ubuntu.html>

3a) Deploy the Ubuntu Conpot Script from the MHN Admin VM to the Honeypot VM: The Conpot ICS Honeypot can be Installed by Executing the following command on the newly created “mhn-ubuntu-1404-conpot-ics-honeypot” virtual machine.

MHN Server Map Deploy Attacks Payloads Rules Sensors Charts

Select Script

Ubuntu - Conpot

Deploy Command

```
wget "http://35.235.87.133/api/script/?text=true&script_id=15" -O deploy.sh && sudo bash deploy.sh
http://35.235.87.133 CgwZ2d70
```

Deploy Script

Name

Ubuntu - Conpot

Script

```
#!/bin/bash

if [ $# -ne 2 ]
then
    echo "Wrong number of arguments supplied."
    echo "Usage: $0 <server_url> <deploy_key>."
    exit 1
fi

server_url=$1
deploy_key=$2

echo "deb http://en.archive.ubuntu.com/ubuntu precise main multiverse" | sudo tee -a /etc/apt
/sources.list
apt-get update
apt-get install -y git libmysqlclient-dev libsmi2lowl snmp-mibs-downloader python-dev libevent-dev
libxml2-dev python-pip python-mysqldb pkg-config libvirt-dev supervisor
apt-get install -y zlib1g-dev # needed for Ubuntu 14.04
pip install --upgrade distribute
pip install virtualenv

CONPOT_HOME=/opt/conpot
mkdir -p $CONPOT_HOME
cd $CONPOT_HOME
virtualenv env
```

Notes

Initial deploy script for Ubuntu - Conpot

```
wget "http://35.236.80.41/api/script/?text=true&script_id=15" -O deploy.sh && sudo bash deploy.sh
http://35.236.80.41 CgwZ2d70
```

```
mhn-admin@mhn-ubuntu-1404-conpot-ics-honeypot:~$ wget "http://35.236.80.41/api/script/?text=true&script_id=15" -O deploy.sh  
&& sudo bash deploy.sh http://35.236.80.41 CgwZ2d70
```

```
--2019-05-15 07:19:56-- http://35.236.80.41/static/registration.txt  
Connecting to 35.236.80.41:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 1430 (1.4K) [text/plain]  
Saving to: 'registration.sh'  
  
100%[=====] 1,430 --.-K/s in 0s  
  
2019-05-15 07:19:56 (177 MB/s) - 'registration.sh' saved [1430/1430]  
  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
python is already the newest version.  
curl is already the newest version.  
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.  
Created sensor: d7fffb7a-76e1-11e9-95b5-42010aa80003  
conpot: added process group  
mhn-admin@mhn-ubuntu-1404-conpot-ics-honeypot:~$
```

Conpot Honeypot Successfully Registered  
Sensor UID d7fffb7a-76e1-11e9-95b5-42010aa80003

3b) Deploy the Ubuntu Snort IDS Sensor from the MHN Admin VM to the Honeypot VM: The Snort IDS Sensor can be Installed by Executing the following command on the newly created “mhn-ubuntu-1404-conpot-ics-honeypot” virtual machine.

The screenshot shows the MHN Server interface at the URL [http://35.236.80.41/ui/manage-deploy/?script\\_id=6](http://35.236.80.41/ui/manage-deploy/?script_id=6). The top navigation bar includes links for MHN Server, Map, Deploy, Attacks, Payloads, Rules, Sensors, and Charts. The main area is titled "Select Script" and contains a dropdown menu with "Ubuntu - Snort" selected. Below this is a "Deploy Command" section containing the command: 

```
wget "http://35.235.87.133/api/script/?text=true&script_id=6" -O deploy.sh && sudo bash deploy.sh  
http://35.235.87.133 CgwZ2d70
```

. A "Deploy Script" section follows, with a "Name" field containing "Ubuntu - Snort".

## Deploy Script

### Name

Ubuntu - Snort

### Script

```
#!/bin/bash

INTERFACE=eth0

set -e
set -x

if [ $# -ne 2 ]
then
    echo "Wrong number of arguments supplied."
    echo "Usage: $0 <server_url> <deploy_key>."
    exit 1
fi

server_url=$1
deploy_key=$2

apt-get update
DEBIAN_FRONTEND=noninteractive apt-get -y install build-essential libpcap-dev libjansson-dev
libpcre3-dev libdnet-dev libdumbnet-dev libdaq-dev flex bison python-pip git make automake libtool
zlib1g-dev

pip install --upgrade distribute
pip install virtualenv
```

### Notes

Initial deploy script for Ubuntu - Snort

#### Deployment Command:

```
wget "http://35.236.80.41/api/script/?text=true&script_id=6" -O deploy.sh && sudo bash deploy.sh
http://35.236.80.41 CgwZ2d70
```

```
mhn-admin@mhn-ubuntu-1404-conpot-ics-honeypot:~$ wget "http://35.236.80.41/api/script/?text=true&script_id=6" -O deploy.sh
&& sudo bash deploy.sh http://35.236.80.41 CgwZ2d70
```

```
mhn-admin@mhn-ubuntu-1404-conpot-ics-honeypot:~$ ps -ef | grep "snort"
root      16652 11149  9 05:51 ?        00:00:24 /opt/snort/bin/snort -c /opt/snort/etc/snort.conf -i eth0
```

Snort IDS Sensor Successfully Deployed

Additionally the Ubuntu p0f Sensor was deployed rom the MHN Admin VM to the Honeypot VM

3c) Deploy the Ubuntu p0f Sensor from the MHN Admin VM to the Honeypot VM: The p0f Sensor can be Installed by Executing the following command on the newly created “mhn-ubuntu-1404-conpot-ics-honeypot” virtual machine.

MHN Server   Map   Deploy   Attacks   Payloads   Rules   Sensors   Charts

Select Script  
Ubuntu - p0f

Deploy Command  

```
wget "http://35.235.87.133/api/script/?text=true&script_id=11" -O deploy.sh && sudo bash deploy.sh
http://35.235.87.133 CgwZ2d70
```

---

Deploy Script  
Name  
Ubuntu - p0f

Script  

```
#!/bin/bash

set -e
set -x

if [ $# -ne 2 ]
then
    echo "Wrong number of arguments supplied."
    echo "Usage: $0 <server_url> <deploy_key>."
    exit 1
fi

server_url=$1
deploy_key=$2

apt-get update
apt-get -y install git supervisor libpcap-dev libjansson-dev gcc

# install p0f
cd /opt
git clone https://github.com/threatstream/p0f.git
cd p0f
git checkout origin/hpfeeds
./build.sh
useradd -d /var/empty/p0f -M -r -s /bin/nologin p0f-user || true
```

Notes  
Initial deploy script for Ubuntu - p0f

#### Deployment Command:

```
wget "http://35.236.80.41/api/script/?text=true&script_id=11" -O deploy.sh
&& sudo bash deploy.sh http://35.236.80.41/ CgwZ2d70
```

```
mhn-admin@mhn-ubuntu-1404-conpot-ics-honeypot:~$ wget "http://35.236.80.41/api/script/?text=true&script_id=11" -O deploy.sh
&& sudo bash deploy.sh http://35.236.80.41 CgwZ2d70
```

+ supervisorctl update  
p0f: added process group

```
mhn-admin@mhn-ubuntu-1404-conpot-ics-honeypot:~$ ps -ef | grep "p0f"
mhn-admin 16718 25004  0 05:58 pts/0    00:00:00 grep --color=auto p0f
p0f-user 32729 11149  0 05:43 ?      00:00:00 /opt/p0f/p0f -i eth0 -u p0f-user -h 35.236.80.41 -P 10000 -I 8a7aa1e0-779d-11e9-95b5-42010aa80003 -K LDAKm89rZXN48jWa -C p0f.events !(src host 10.168.0.6)
```

p0F Sensor Successfully Deployed

- 4) Perform a Network Port Scan using the NMAP Utility: An NMAP Scan was performed to simulate a Basic Port Scan Attack which would reflect in the MHN Server Dashboard.

```
rkashi@Rohits-MacBook-Pro:~$ nmap 35.236.28.214
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-15 00:23 PDT
Nmap scan report for 214.28.236.35.bc.googleusercontent.com (35.236.28.214)
Host is up (0.052s latency).
Not shown: 994 filtered ports
PORT      STATE SERVICE
21/tcp    closed  ftp
22/tcp    open   ssh
80/tcp    open   http
2121/tcp  closed ccproxy-ftp
8800/tcp  closed sunwebadmin
20000/tcp closed  dnp

Nmap done: 1 IP address (1 host up) scanned in 7.37 seconds
```

## Sensors

---

```
2- [ ] mhn-ubuntu-1404-conpot-ics mhn-ubuntu-1404-conpot-ics-honeypot 35.236.28.214 conpot d7ffb7a-76e1-11e9-95b5-42010aa80003 32
```

# Honeypot 3 Combo

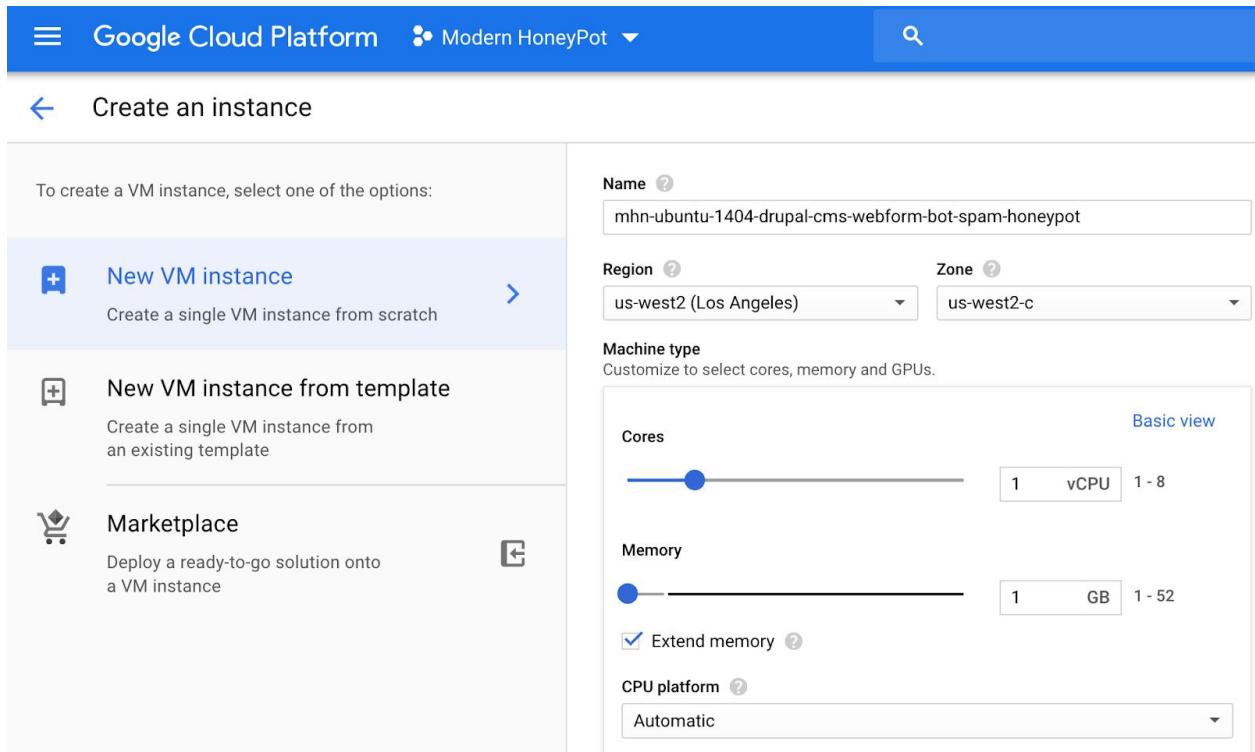
## Drupot (Drupal CMS Webform Bot Spam Honeypot)

### Snort IDS Sensor

### P0f TCP/IP Fingerprinting Sensor

**Instance Creation:** Using Compute Engine Create an Ubuntu 14.04 VM Instance

**Firewall Rules:** Firewall Rules were created to allow HTTP and HTTPS Traffic to this VM.



The screenshot shows the Google Cloud Platform interface for creating a new VM instance. On the left, there's a sidebar with options: 'New VM instance' (selected), 'New VM instance from template', and 'Marketplace'. The main area has fields for 'Name' (mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot), 'Region' (us-west2), 'Zone' (us-west2-c), and 'Machine type' settings. It includes sliders for 'Cores' (1 vCPU) and 'Memory' (1 GB), with an option to 'Extend memory'. A 'Basic view' link is visible above the memory slider.

To create a VM instance, select one of the options:

- New VM instance** >  
Create a single VM instance from scratch
- New VM instance from template**  
Create a single VM instance from an existing template
- Marketplace** [ ]  
Deploy a ready-to-go solution onto a VM instance

Name ? mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot

Region ? us-west2 (Los Angeles) Zone ? us-west2-c

Machine type  
Customize to select cores, memory and GPUs.

Cores Basic view 1 vCPU 1 - 8

Memory 1 GB 1 - 52

Extend memory ?

CPU platform ? Automatic

### Firewall ?

Add tags and firewall rules to allow specific network traffic from the Internet

- Allow HTTP traffic
- Allow HTTPS traffic

Management

Security

Disks

**Networking**

Sole Tenancy

### Network tags ? (Optional)

drupal-cms-webform-botspam-honepot

### Hostname ?

Set a custom hostname for this instance or leave it default

mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honepot.us-west2-c.c.vertica

### Network interfaces ?

default default (10.168.0.0/20) 

 Add network interface

 To create another network interface you need to have a new network first.

 Less

Your free trial credit will be used for this VM instance. [GCP Free Tier](#) 

**Create**

Cancel

<input type="checkbox"/>	 mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honepot	us-west2-c	10.168.0.9 (nic0)	35.235.101.217 	SSH 	
--------------------------	---	------------	----------------------	--	---	---

**Drupal CMS Webform Spambot Honeypot VM Successfully Created**

2a) [Deploy the Ubuntu Drupal Script from the MHN Admin VM to the Honeypot VM:](#)

The Drupal Honeypot can be Installed by Executing the following command on the newly created “mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honepot” virtual machine.

## Select Script

Ubuntu - Drupot

## Deploy Command

```
wget "http://35.235.87.133/api/script/?text=true&script_id=14" -O deploy.sh && sudo bash deploy.sh  
http://35.235.87.133 CgwZ2d70
```

## Deploy Script

### Name

Ubuntu - Drupot

### Script

```
set -e
set -x

if [ $# -ne 2 ]
then
    echo "Wrong number of arguments supplied."
    echo "Usage: $0 <server_url> <deploy_key>."
    exit 1
fi

server_url=$1
deploy_key=$2

apt-get update
apt-get -y install git supervisor

if [ "$(uname -m)" == "x86_64" ] ;
then
    GO_PACKAGE="gol.12.1.linux-amd64.tar.gz"
else
    GO_PACKAGE="gol.12.1.linux-386.tar.gz"
fi

cd /usr/local/
wget https://storage.googleapis.com/golang/${GO_PACKAGE}
```

### Notes

Initial deploy script for Ubuntu - Drupot

```
mhn-admin@mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot:~$ wget "http://35.236.80.41/api/script/?text=true&script_id=14"
-O deploy.sh && sudo bash deploy.sh http://35.236.80.41 CgwZ2d70
```

```
+ supervisorctl update
drupot: added process group
+ supervisorctl restart all
drupot: stopped
drupot: started
```

Drupot Honeypot Successfully Registered

Additionally the Ubuntu Snort IDS Sensor was deployed from the MHN Admin VM to the Honeypot VM

2b) Deploy the Ubuntu Snort IDS Sensor from the MHN Admin VM to the Honeypot VM: The Snort IDS Sensor can be Installed by Executing the following command on the newly created “mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot” virtual machine.

The screenshot shows the MHN Admin UI interface. At the top, there's a navigation bar with links for MHN Server, Map, Deploy, Attacks, Payloads, Rules, Sensors, and Charts. Below the navigation, a URL bar shows the address: 35.236.80.41/ui/manage-deploy/?script\_id=6. The main content area has a title "Select Script" with a dropdown menu containing "Ubuntu - Snort". Under "Deploy Command", there is a code block with the following content:

```
wget "http://35.235.87.133/api/script/?text=true&script_id=6" -O deploy.sh && sudo bash deploy.sh
http://35.235.87.133 CgwZ2d70
```

---

Below this, there's a section titled "Deploy Script" with a "Name" field containing "Ubuntu - Snort".

## Deploy Script

### Name

Ubuntu - Snort

### Script

```
#!/bin/bash

INTERFACE=eth0

set -e
set -x

if [ $# -ne 2 ]
then
    echo "Wrong number of arguments supplied."
    echo "Usage: $0 <server_url> <deploy_key>."
    exit 1
fi

server_url=$1
deploy_key=$2

apt-get update
DEBIAN_FRONTEND=noninteractive apt-get -y install build-essential libpcap-dev libjansson-dev
libpcre3-dev libdnet-dev libdumbnet-dev libdaq-dev flex bison python-pip git make automake libtool
zlib1g-dev

pip install --upgrade distribute
pip install virtualenv
```

### Notes

Initial deploy script for Ubuntu - Snort

#### Deployment Command:

```
wget "http://35.236.80.41/api/script/?text=true&script_id=6" -O deploy.sh && sudo bash deploy.sh
http://35.236.80.41 CgwZ2d70
```

```
mhn-admin@mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot:~$ wget "http://35.236.80.41/api/script/?text=true&script_id=6" -O deploy.sh
&& sudo bash deploy.sh http://35.236.80.41 CgwZ2d70
```

```
mhn-admin@mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot:~$ ps -efaf | grep "snort"
mhn-admin+ 1286 17702 0 06:42 pts/1 00:00:00 grep --color=auto snort
root 24320 3027 1 06:13 ? 00:00:22 /opt/snort/bin/snort -c /opt/snort/etc/snort.conf -i eth0
```

Snort IDS Sensor Successfully Deployed

Additionally the Ubuntu p0f Sensor was deployed from the MHN Admin VM to the Honeypot VM

3c) Deploy the Ubuntu p0f Sensor from the MHN Admin VM to the Honeypot VM: The p0f Sensor can be Installed by Executing the following command on the newly created “mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot” virtual machine.

MHN Server    Map    Deploy    Attacks    Payloads    Rules    Sensors    Charts

Select Script  
Ubuntu - p0f

Deploy Command  

```
wget "http://35.235.87.133/api/script/?text=true&script_id=11" -O deploy.sh && sudo bash deploy.sh
http://35.235.87.133 CgwZ2d70
```

---

Deploy Script  
Name  
Ubuntu - p0f

Script  

```
#!/bin/bash

set -e
set -x

if [ $# -ne 2 ]
then
    echo "Wrong number of arguments supplied."
    echo "Usage: $0 <server_url> <deploy_key>."
    exit 1
fi

server_url=$1
deploy_key=$2

apt-get update
apt-get -y install git supervisor libpcap-dev libjansson-dev gcc
# install p0f
cd /opt
git clone https://github.com/threatstream/p0f.git
cd p0f
git checkout origin/hpfeeds
./build.sh
useradd -d /var/empty/p0f -M -r -s /bin/nologin p0f-user || true
```

Notes  
Initial deploy script for Ubuntu - p0f

#### Deployment Command:

```
wget "http://35.236.80.41/api/script/?text=true&script_id=11" -O deploy.sh
&& sudo bash deploy.sh http://35.236.80.41/CgwZ2d70
```

```
mhn-admin@mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot:~$ wget "http://35.236.80.41/api/script/?text=true&script_id=11" -O deploy.sh && sudo bash deploy.sh http://35.236.80.41 CgwZ2d70
```

+ supervisorctl update  
p0f: added process group

```
mhn-admin@mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot:~$ ps -ef | grep "p0f"
p0f-user 5495 3027 0 06:10 ? 00:00:00 /opt/p0f/p0f -l eth0 -u p0f-user -h 35.236.80.41 -P 10000 -I 481fe2ac-77a1-11e9-95b5-42010aa80003 -K f1EoKk7kQYKjoXFE -C p0f.events !(src host 10.168.0.9)
mhn-admin+ 6200 17614 0 06:10 pts/0 00:00:00 grep --color=auto p0f
```

p0F Sensor Successfully Deployed

3) Perform a Network Port Scan using the NMAP Utility: An NMAP Scan was performed to simulate a Basic Port Scan Attack which would reflect in the MHN Server Dashboard.

```
rkashi@Rohits-MacBook-Pro:~$ nmap 35.235.101.217
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-15 01:08 PDT
Nmap scan report for 217.101.235.35.bc.googleusercontent.com (35.235.101.217)
Host is up (0.051s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    closed http
443/tcp   closed https
3000/tcp  closed ppp
10000/tcp closed snet-sensor-mgmt

Nmap done: 1 IP address (1 host up) scanned in 7.32 seconds
```

MHN Server    Map    Deploy    Attacks    Payloads    Rules ▾    Sensors ▾    Charts ▾

## Sensors

3- mhn-ubuntu-1404-drupal-cm mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot 35.235.101.217 drupot 38965a42-76e7-11e9-95b5-42010aa80003 626

References:

- <https://www.drupal.org/project/honeypot>
- <https://www.drupal.org/docs/8/modules/honeypot/using-honeypot>
- <https://github.com/geerlingguy/drupal-honeypot>
- <https://www.drupal.org/docs/8/modules/honeypot/honeypot-success-stories>

## Honeypot 4: Cowrie SSH/Telnet Honeypot

- 1) **Instance Creation:** Using Compute Engine Create an Ubuntu 14.04 VM Instance

Name 

Region 



Zone 



**Machine type**  
Customize to select cores, memory and GPUs.

[Basic view](#)

Cores

1 vCPU 1 - 8

Memory

1 GB 1 - 52

Extend memory 

### Boot disk



New 10 GB standard persistent disk

Image

Ubuntu 14.04 LTS

[Change](#)

### Identity and API access

#### Service account

Compute Engine default service account

▼

#### Access scopes

- Allow default access
- Allow full access to all Cloud APIs
- Set access for each API

### Firewall

Add tags and firewall rules to allow specific network traffic from the Internet

- Allow HTTP traffic
- Allow HTTPS traffic

Management

Security

Disks

**Networking**

Sole Tenancy

### Network tags (Optional)

cowrie-ssh-telnet-honeypot 

### Hostname

Set a custom hostname for this instance or leave it default

mhn-ubuntu-1404-cowrie-ssh-telnet-honeypot.us-west2-a.c.vertical-backup-240f

### Network interfaces

default default (10.168.0.0/20) 

<input type="checkbox"/>	 mhn-ubuntu-1404-cowrie-ssh-telnet-honeypot	us-west2-a	10.168.0.10 (nic0)	35.236.5.125	SSH	
--------------------------	--	------------	-----------------------	--------------	-----	---

Cowrie SSH, Telnet Honeypot VM Successfully Created

2) **Firewall Rule:** Modify the Existing Firewall Rule to allow any Ingress Traffic to this VM.

mhn-allow-any-traffic-dionaea-honeypot      Ingress      dionaea-honeypot, 2 more      IP ranges: 0.0.0.0/0      all      Allow      1      default

Since this virtual machine has the network tag “**cowrie-ssh-telnet-honeypot**” all that needs to be done is to add this network tag to the existing firewall rule to allow any ingress traffic to the to the newly created “**mhn-ubuntu-1404-cowrie-ssh-telnet-honeypot**”.

[Firewall rule details](#) [EDIT](#) [DELETE](#)

**mhn-allow-any-traffic-dionaea-honeypot**

**Description**  
Allow Any Traffic to the Conpot, Dionaea, Drupal Honeypots

**Logs**  
Turning on firewall logs can generate a large number of logs which can increase costs in Stackdriver. [Learn more](#)

On  
 Off

**Network**  
default

**Priority** ⓘ  
Priority can be 0 - 65535 [Check priority of other firewall rules](#)  
1

**Direction**  
Ingress

**Action on match**  
Allow

**Targets**  
Specified target tags

**Target tags**  
dionaea-honeypot × durpal-cms-webform-botspam-honeypot ×  
conpot-ics-honeypot × cowrie-ssh-telnet-honeypot ×

**Source filter** ⓘ  
IP ranges

**Source IP ranges** ⓘ  
0.0.0.0/0 ×

**Second source filter** ⓘ  
None

**Protocols and ports**  
 Allow all  
 Specified protocols and ports

[Disable rule](#)

[Save](#) [Cancel](#)

### 3) Deploy the Ubuntu Cowrie Script from the MHN Admin VM to the Cowrie Honeypot VM:

The Drupal Honeypot can be Installed by Executing the following command on the newly created "mhn-ubuntu-1404-cowrie-ssh-telnet-honeypot" virtual machine.

MHN Server Map Deploy Attacks Payloads Rules Sensors Charts

Select Script  
Ubuntu - Cowrie

Deploy Command  
wget "http://35.235.87.133/api/script/?text=true&script\_id=5" -O deploy.sh && sudo bash deploy.sh  
http://35.235.87.133 CgwZ2d70

## Deploy Script

### Name

Ubuntu - Cowrie

### Script

```
#!/bin/bash

set -e
set -x

if [ $# -ne 2 ]
then
    echo "Wrong number of arguments supplied."
    echo "Usage: $0 <server_url> <deploy_key>."
    exit 1
fi

apt-get update
apt-get install -y python

server_url=$1
deploy_key=$2

apt-get update
apt-get -y install python-dev git supervisor authbind openssl python-virtualenv build-essential
python-gmpy2 libgmp-dev libmpfr-dev libmpc-dev libssl-dev python-pip libffi-dev

pip install -U supervisor
/etc/init.d/supervisor start || true
```

### Notes

Initial deploy script for Ubuntu - Cowrie

```
mhn-admin@mhn-ubuntu-1404-cowrie-ssh-telnet-honeypot:~$ wget "http://35.236.80.41/api/script/?text=true&script_id=5" -O deploy.sh
&& sudo bash deploy.sh http://35.236.80.41 CgwZ2d70
```

## + supervisorctl update cowrie: added process group

Cowrie Honeypot Successfully Registered

**Issue:** The cowrie process is in a FATAL State which means that there is an issue

```
mhn-admin@mhn-ubuntu-1404-cowrie-ssh-telnet-honeypot:~$ sudo supervisorctl status  
cowrie FATAL Exited too quickly (process log may have details)
```

Error

```
Unhandled Error  
Traceback (most recent call last):  
  File "/opt/cowrie/cowrie-env/local/lib/python2.7/site-packages/twisted/python/usage.py", line 447, in __str__  
    return self.getSynopsis() + '\n' + self.getUsage(width=None)  
  File "/opt/cowrie/cowrie-env/local/lib/python2.7/site-packages/twisted/python/usage.py", line 484, in getUsage  
    for (cmd, short, parser, desc) in self.subCommands:  
  File "/opt/cowrie/cowrie-env/local/lib/python2.7/site-packages/twisted/application/app.py", line 653, in subCommands  
    for plug in sorted(plugins, key=attrgetter('tapname')):  
  File "/opt/cowrie/cowrie-env/local/lib/python2.7/site-packages/twisted/plugin.py", line 213, in getPlugins  
    allDropins = getCache(package)  
--- <exception caught here> ---  
  File "/opt/cowrie/cowrie-env/local/lib/python2.7/site-packages/twisted/plugin.py", line 171, in getCache  
    provider = pluginModule.load()  
  File "/opt/cowrie/cowrie-env/local/lib/python2.7/site-packages/twisted/python/modules.py", line 392, in load  
    return self.pathEntry.pythonPath.moduleLoader(self.name)  
  File "/opt/cowrie/cowrie-env/local/lib/python2.7/site-packages/twisted/python/reflect.py", line 308, in namedAny  
    topLevelPackage = _importAndCheckStack(trialname)  
  File "/opt/cowrie/cowrie-env/local/lib/python2.7/site-packages/twisted/python/reflect.py", line 255, in _importAndCheckStack  
    reraise(excValue, excTraceback)  
  File "/opt/cowrie/src/twisted/plugins/cowrie_plugin.py", line 45, in <module>  
    import cowrie.core.checkers  
  File "/opt/cowrie/src/cowrie/core/checkers.py", line 13, in <module>  
    from twisted.conch.ssh import keys  
  File "/opt/cowrie/cowrie-env/local/lib/python2.7/site-packages/twisted/conch/ssh/keys.py", line 18, in <module>  
    import bcrypt  
exceptions ImportError: No module named bcrypt
```

/opt/cowrie/var/log/cowrie/cowrie.err

<https://github.com/cowrie/cowrie/issues/349> : Is a similar issue

about start cowrie by the way supervisorctl and start.sh ;

Closed duolaaa opened this issue on Nov 14, 2016 · 2 comments



duolaaa commented on Nov 14, 2016

+ ...

Hello everyone  
i meet a problem like this ;

• one:

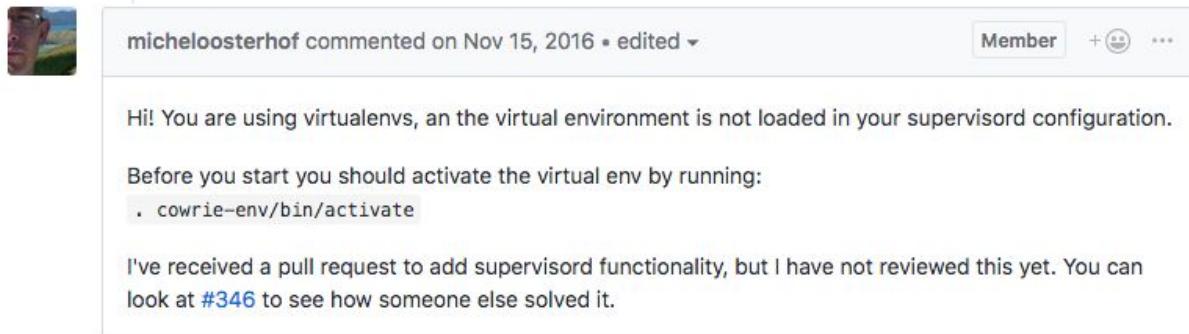
when i change user to cowrie , i can execute ./start.sh to start cowrie,  
ps -aux | grep cowrie

```
cowrie 1325 0.0 2.5 109520 49800 ? S 13:41 0:00 /usr/bin/python /usr/local/bin/twistd -l  
og/cowrie.log --umask 0077 --pidfile var/run/cowrie.pid cowrie
```

• two

when i use supervisorctl  
supervisorctl status  
cowrie FATAL Exited too quickly (process log may have details)  
  
supervisorctl start cowrie  
cowrie: ERROR (abnormal termination)

I tried the suggestion from the github forum **but it did not work**



micheloosterhof commented on Nov 15, 2016 • edited

Member + 88 ...

Hi! You are using virtualenvs, an the virtual environment is not loaded in your supervisord configuration.

Before you start you should activate the virtual env by running:

```
. /opt/cowrie/cowrie-env/bin/activate
```

I've received a pull request to add supervisord functionality, but I have not reviewed this yet. You can look at #346 to see how someone else solved it.

### Suggested Solution

```
mhn-admin@mhn-ubuntu-1404-cowrie-ssh-telnet-honeypot:/$ sudo cat /etc/supervisor/conf.d/cowrie.conf
[program:cowrie]
. /opt/cowrie/cowrie-env/bin/activate
command=/opt/cowrie/bin/cowrie start
directory=/opt/cowrie
stdout_logfile=/opt/cowrie/var/log/cowrie/cowrie.out
stderr_logfile=/opt/cowrie/var/log/cowrie/cowrie.err
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
user=cowrie
mhn-admin@mhn-ubuntu-1404-cowrie-ssh-telnet-honeypot:/$ sudo supervisorctl restart all
cowrie: ERROR (abnormal termination)
```

[/etc/supervisor/conf.d/cowrie.conf](#)

Issue 2: <https://github.com/cowrie/cowrie/issues/412>

## Error running cowrie after install #412

Closed nwcook opened this issue on Jan 18, 2017 · 3 comments

When I start

```
sudo supervisorctl start cowrie
```

I get

```
cowrie: ERROR (abnormal termination)
```

in cowrie.out:

```
/usr/bin/twistd: Unknown command: cowrie
```

```
/opt/cowrie/cowrie-env/bin/twistd: Unknown command: cowrie
(cowrie-env)mhn-admin@mhn-ubuntu-1404-cowrie-ssh-telnet-honeypot:/opt/cowrie$ /opt/cowrie/bin/cowrie
```

Tried the suggestion listed in the github forum, **but it did not work**



lelonnek1 commented on Jan 18, 2017

Contributor + ...

The script from MHN is installing Twisted with apt-get instead of pip, and the version in Ubuntu's repository is too old to run Cowrie. You need to install at least Twisted 15.2.1, and possibly other dependencies. The easiest way to make sure you have everything Cowrie requires is to run `sudo pip install -U -r requirements.txt` from /opt/cowrie.



1

## Suggestion Solution

```
(cowrie-env)mhn-admin@mhn-ubuntu-1404-cowrie-ssh-telnet-honeypot:/opt/cowrie$ cat requirements.txt
twisted>=17.1.0
cryptography>=0.9.1,<=1.8
configparser
pyopenssl
pyparsing
packaging
appdirs>=1.4.0
pyasn1_modules
attrs
service_identity
python-dateutil
tftpy
```

**None of the suggested solution worked for me**

Note: The issue was persistent even when the Ubuntu 16.04 was used as a Base Image.

## Sensors

4-	mhn-ubuntu-1404-cowrie-ssl	mhn-ubuntu-1404-cowrie-ssh-telnet-honeypot	35.236.5.125	cowrie	6de848d6-7778-11e9-95b5-42010aa80003	0
----	----------------------------	--	--------------	--------	--------------------------------------	---

## RESULTS SUMMARY

### Attack Stats

Attacks in the last 24 hours: **7,866**

TOP 5 Attacker IPs:

1.  **38.141.45.50** (2,664 attacks)
2.  **77.247.110.36** (989 attacks)
3.  **111.230.73.43** (615 attacks)
4.  **75.25.131.207** (344 attacks)
5.  **193.188.22.115** (214 attacks)

TOP 5 Attacked ports:

1. **5060** (3,761 times)
2. **80** (729 times)
3. **8088** (530 times)
4. **3389** (290 times)
5. **23** (233 times)

TOP 5 Honey Pots:

1. **dionaea** (6,677 attacks)
2. **drupot** (626 attacks)
3. **snort** (295 attacks)
4. **p0f** (234 attacks)
5. **conpot** (34 attacks)

TOP 5 Sensors:

1. **mhn-ubuntu-1404-dionaea-honeypot-with-http** (6,677 attacks)
2. **mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot** (626 attacks)
3. **mhn-ubuntu-1404-dionaea-honeypot-with-http** (243 attacks)
4. **mhn-ubuntu-1404-conpot-ics-honeypot** (131 attacks)
5. **mhn-admin-vm** (100 attacks)

TOP 5 Attacks Signatures:

1. **ET SCAN SipCLI VOIP Scan** (131 times)
2. **ET DROP Dshield Block Listed Source group 1** (72 times)
3. **ET CINS Active Threat Intelligence Poor Reputation IP TCP group 81** (20 times)
4. **ET CINS Active Threat Intelligence Poor Reputation IP TCP group 80** (15 times)
5. **ET SCAN Potential SSH Scan** (9 times)

[ATTACK STATS](#)

## Attacks Report

Search Filters

Sensor	Honeypot	Date	Port	IP Address																																																																																								
All	All	MM-DD-YYYY	445	8.8.8.8	<b>GO</b>																																																																																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Date</th> <th>Sensor</th> <th>Country</th> <th>Src IP</th> <th>Dst port</th> <th>Protocol</th> <th>Honeypot</th> </tr> </thead> <tbody> <tr><td>1</td><td>2019-05-16 07:05:10</td><td>mhn-ubuntu-1404-dionaea-honeypot-with-http</td><td>USA</td><td>38.141.45.50</td><td>5060</td><td>UDP</td><td>snort</td></tr> <tr><td>2</td><td>2019-05-16 07:05:09</td><td>mhn-ubuntu-1404-dionaea-honeypot-with-http</td><td>USA</td><td>38.141.45.50</td><td>5060</td><td>SipCall</td><td>dionaea</td></tr> <tr><td>3</td><td>2019-05-16 07:05:09</td><td>mhn-ubuntu-1404-dionaea-honeypot-with-http</td><td>USA</td><td>38.141.45.50</td><td>5060</td><td>SipSession</td><td>dionaea</td></tr> <tr><td>4</td><td>2019-05-16 07:04:56</td><td>mhn-ubuntu-1404-conpot-ics-honeypot</td><td>USA</td><td>68.183.16.108</td><td>8088</td><td>pcap</td><td>p0f</td></tr> <tr><td>5</td><td>2019-05-16 07:04:49</td><td>mhn-ubuntu-1404-dionaea-honeypot-with-http</td><td>USA</td><td>38.141.45.50</td><td>5060</td><td>SipCall</td><td>dionaea</td></tr> <tr><td>6</td><td>2019-05-16 07:04:49</td><td>mhn-ubuntu-1404-dionaea-honeypot-with-http</td><td>USA</td><td>38.141.45.50</td><td>5060</td><td>SipSession</td><td>dionaea</td></tr> <tr><td>7</td><td>2019-05-16 07:04:31</td><td>mhn-ubuntu-1404-dionaea-honeypot-with-http</td><td>ROU</td><td>27.92.117.238</td><td>22</td><td>TCP</td><td>snort</td></tr> <tr><td>8</td><td>2019-05-16 07:04:31</td><td>mhn-ubuntu-1404-dionaea-honeypot-with-http</td><td>ROU</td><td>27.92.117.238</td><td>22</td><td>TCP</td><td>snort</td></tr> <tr><td>9</td><td>2019-05-16 07:04:30</td><td>mhn-ubuntu-1404-conpot-ics-honeypot</td><td>RUS</td><td>185.176.27.50</td><td>8956</td><td>TCP</td><td>snort</td></tr> <tr><td>10</td><td>2019-05-16 07:04:30</td><td>mhn-ubuntu-1404-conpot-ics-honeypot</td><td>RUS</td><td>185.176.27.50</td><td>8956</td><td>pcap</td><td>p0f</td></tr> </tbody> </table>						Date	Sensor	Country	Src IP	Dst port	Protocol	Honeypot	1	2019-05-16 07:05:10	mhn-ubuntu-1404-dionaea-honeypot-with-http	USA	38.141.45.50	5060	UDP	snort	2	2019-05-16 07:05:09	mhn-ubuntu-1404-dionaea-honeypot-with-http	USA	38.141.45.50	5060	SipCall	dionaea	3	2019-05-16 07:05:09	mhn-ubuntu-1404-dionaea-honeypot-with-http	USA	38.141.45.50	5060	SipSession	dionaea	4	2019-05-16 07:04:56	mhn-ubuntu-1404-conpot-ics-honeypot	USA	68.183.16.108	8088	pcap	p0f	5	2019-05-16 07:04:49	mhn-ubuntu-1404-dionaea-honeypot-with-http	USA	38.141.45.50	5060	SipCall	dionaea	6	2019-05-16 07:04:49	mhn-ubuntu-1404-dionaea-honeypot-with-http	USA	38.141.45.50	5060	SipSession	dionaea	7	2019-05-16 07:04:31	mhn-ubuntu-1404-dionaea-honeypot-with-http	ROU	27.92.117.238	22	TCP	snort	8	2019-05-16 07:04:31	mhn-ubuntu-1404-dionaea-honeypot-with-http	ROU	27.92.117.238	22	TCP	snort	9	2019-05-16 07:04:30	mhn-ubuntu-1404-conpot-ics-honeypot	RUS	185.176.27.50	8956	TCP	snort	10	2019-05-16 07:04:30	mhn-ubuntu-1404-conpot-ics-honeypot	RUS	185.176.27.50	8956	pcap	p0f
Date	Sensor	Country	Src IP	Dst port	Protocol	Honeypot																																																																																						
1	2019-05-16 07:05:10	mhn-ubuntu-1404-dionaea-honeypot-with-http	USA	38.141.45.50	5060	UDP	snort																																																																																					
2	2019-05-16 07:05:09	mhn-ubuntu-1404-dionaea-honeypot-with-http	USA	38.141.45.50	5060	SipCall	dionaea																																																																																					
3	2019-05-16 07:05:09	mhn-ubuntu-1404-dionaea-honeypot-with-http	USA	38.141.45.50	5060	SipSession	dionaea																																																																																					
4	2019-05-16 07:04:56	mhn-ubuntu-1404-conpot-ics-honeypot	USA	68.183.16.108	8088	pcap	p0f																																																																																					
5	2019-05-16 07:04:49	mhn-ubuntu-1404-dionaea-honeypot-with-http	USA	38.141.45.50	5060	SipCall	dionaea																																																																																					
6	2019-05-16 07:04:49	mhn-ubuntu-1404-dionaea-honeypot-with-http	USA	38.141.45.50	5060	SipSession	dionaea																																																																																					
7	2019-05-16 07:04:31	mhn-ubuntu-1404-dionaea-honeypot-with-http	ROU	27.92.117.238	22	TCP	snort																																																																																					
8	2019-05-16 07:04:31	mhn-ubuntu-1404-dionaea-honeypot-with-http	ROU	27.92.117.238	22	TCP	snort																																																																																					
9	2019-05-16 07:04:30	mhn-ubuntu-1404-conpot-ics-honeypot	RUS	185.176.27.50	8956	TCP	snort																																																																																					
10	2019-05-16 07:04:30	mhn-ubuntu-1404-conpot-ics-honeypot	RUS	185.176.27.50	8956	pcap	p0f																																																																																					

[1](#) [2](#) [3](#) [4](#) [5](#) ... [789](#) [790](#) >

## ATTACKS

## Payloads Report

Search Filters

Payload	Regex Term																																																																														
snort.alerts	port regex	<b>GO</b>																																																																													
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>date</th> <th>sensor</th> <th>source_ip</th> <th>destination_port</th> <th>priority</th> <th>classification</th> <th>signature</th> </tr> </thead> <tbody> <tr><td>7e2382be-7792-11e9-95b5-42010aa80003</td><td></td><td>81.22.45.193</td><td>7876</td><td>2</td><td>30</td><td>ET DROP Dshield Block Listed Source group 1</td></tr> <tr><td>7e2382be-7792-11e9-95b5-42010aa80003</td><td></td><td>81.22.45.193</td><td>7876</td><td>2</td><td>30</td><td>ET CINS Active Threat Intelligence Poor Reputation IP TCP group 80</td></tr> <tr><td>7e2382be-7792-11e9-95b5-42010aa80003</td><td></td><td>38.141.45.50</td><td>5060</td><td>2</td><td>4</td><td>ET SCAN SipCLI VOIP Scan</td></tr> <tr><td>7e2382be-7792-11e9-95b5-42010aa80003</td><td></td><td>27.92.117.238</td><td>22</td><td>2</td><td>30</td><td>ET CINS Active Threat Intelligence Poor Reputation IP TCP group 17</td></tr> <tr><td>7e2382be-7792-11e9-95b5-42010aa80003</td><td></td><td>27.92.117.238</td><td>22</td><td>2</td><td>30</td><td>ET COMPROMISED Known Compromised or Hostile Host Traffic TCP group 16</td></tr> <tr><td>a564a496-7792-11e9-95b5-42010aa80003</td><td></td><td>185.176.27.50</td><td>8956</td><td>2</td><td>30</td><td>ET DROP Dshield Block Listed Source group 1</td></tr> <tr><td>7e2382be-7792-11e9-95b5-42010aa80003</td><td></td><td>38.141.45.50</td><td>5060</td><td>2</td><td>4</td><td>ET SCAN SipCLI VOIP Scan</td></tr> <tr><td>7e2382be-7792-11e9-95b5-42010aa80003</td><td></td><td>38.141.45.50</td><td>5060</td><td>2</td><td>4</td><td>ET SCAN SipCLI VOIP Scan</td></tr> <tr><td>7e2382be-7792-11e9-95b5-42010aa80003</td><td></td><td>38.141.45.50</td><td>5060</td><td>2</td><td>4</td><td>ET SCAN SipCLI VOIP Scan</td></tr> <tr><td>7e2382be-7792-11e9-95b5-42010aa80003</td><td></td><td>77.247.109.231</td><td>5059</td><td>2</td><td>30</td><td>ET CINS Active Threat Intelligence Poor Reputation IP UDP group 70</td></tr> </tbody> </table>			date	sensor	source_ip	destination_port	priority	classification	signature	7e2382be-7792-11e9-95b5-42010aa80003		81.22.45.193	7876	2	30	ET DROP Dshield Block Listed Source group 1	7e2382be-7792-11e9-95b5-42010aa80003		81.22.45.193	7876	2	30	ET CINS Active Threat Intelligence Poor Reputation IP TCP group 80	7e2382be-7792-11e9-95b5-42010aa80003		38.141.45.50	5060	2	4	ET SCAN SipCLI VOIP Scan	7e2382be-7792-11e9-95b5-42010aa80003		27.92.117.238	22	2	30	ET CINS Active Threat Intelligence Poor Reputation IP TCP group 17	7e2382be-7792-11e9-95b5-42010aa80003		27.92.117.238	22	2	30	ET COMPROMISED Known Compromised or Hostile Host Traffic TCP group 16	a564a496-7792-11e9-95b5-42010aa80003		185.176.27.50	8956	2	30	ET DROP Dshield Block Listed Source group 1	7e2382be-7792-11e9-95b5-42010aa80003		38.141.45.50	5060	2	4	ET SCAN SipCLI VOIP Scan	7e2382be-7792-11e9-95b5-42010aa80003		38.141.45.50	5060	2	4	ET SCAN SipCLI VOIP Scan	7e2382be-7792-11e9-95b5-42010aa80003		38.141.45.50	5060	2	4	ET SCAN SipCLI VOIP Scan	7e2382be-7792-11e9-95b5-42010aa80003		77.247.109.231	5059	2	30	ET CINS Active Threat Intelligence Poor Reputation IP UDP group 70
date	sensor	source_ip	destination_port	priority	classification	signature																																																																									
7e2382be-7792-11e9-95b5-42010aa80003		81.22.45.193	7876	2	30	ET DROP Dshield Block Listed Source group 1																																																																									
7e2382be-7792-11e9-95b5-42010aa80003		81.22.45.193	7876	2	30	ET CINS Active Threat Intelligence Poor Reputation IP TCP group 80																																																																									
7e2382be-7792-11e9-95b5-42010aa80003		38.141.45.50	5060	2	4	ET SCAN SipCLI VOIP Scan																																																																									
7e2382be-7792-11e9-95b5-42010aa80003		27.92.117.238	22	2	30	ET CINS Active Threat Intelligence Poor Reputation IP TCP group 17																																																																									
7e2382be-7792-11e9-95b5-42010aa80003		27.92.117.238	22	2	30	ET COMPROMISED Known Compromised or Hostile Host Traffic TCP group 16																																																																									
a564a496-7792-11e9-95b5-42010aa80003		185.176.27.50	8956	2	30	ET DROP Dshield Block Listed Source group 1																																																																									
7e2382be-7792-11e9-95b5-42010aa80003		38.141.45.50	5060	2	4	ET SCAN SipCLI VOIP Scan																																																																									
7e2382be-7792-11e9-95b5-42010aa80003		38.141.45.50	5060	2	4	ET SCAN SipCLI VOIP Scan																																																																									
7e2382be-7792-11e9-95b5-42010aa80003		38.141.45.50	5060	2	4	ET SCAN SipCLI VOIP Scan																																																																									
7e2382be-7792-11e9-95b5-42010aa80003		77.247.109.231	5059	2	30	ET CINS Active Threat Intelligence Poor Reputation IP UDP group 70																																																																									

[1](#) [2](#) [3](#) [4](#) [5](#) ... [29](#) [30](#) >

## SNORT IDS ALERTS

## Sensors

	Name	Hostname	IP	Honeypot	UUID	Attacks
1-	mhn-ubuntu-1404-dionaea-h	mhn-ubuntu-1404-dionaea-honeypot-with-http	35.235.87.133	dionaea	6f905700-76d3-11e9-95b5-42010aa80003	6713
2-	mhn-ubuntu-1404-conpot-ics	mhn-ubuntu-1404-conpot-ics-honeypot	35.236.28.214	conpot	d7ffb7a-76e1-11e9-95b5-42010aa80003	34
3-	mhn-ubuntu-1404-drupal-cm	mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot	35.235.101.217	drupot	38965a42-76e7-11e9-95b5-42010aa80003	626
4-	mhn-ubuntu-1404-cowie-ssl	mhn-ubuntu-1404-cowie-ssh-telnet-honeypot	35.236.5.125	cowie	c79b1e7c-7790-11e9-95b5-42010aa80003	0
5-	mhn-ubuntu-1404-dionaea-h	mhn-ubuntu-1404-dionaea-honeypot-with-http	35.235.87.133	snort	7e2382be-7792-11e9-95b5-42010aa80003	250
6-	mhn-admin-vm-p0f	mhn-admin-vm	35.236.80.41	p0f	ed9f2186-779a-11e9-95b5-42010aa80003	106
7-	mhn-ubuntu-1404-conpot-ics	mhn-ubuntu-1404-conpot-ics-honeypot	35.236.28.214	p0f	8a7aa1e0-779d-11e9-95b5-42010aa80003	134
8-	mhn-ubuntu-1404-conpot-ics	mhn-ubuntu-1404-conpot-ics-honeypot	35.236.28.214	snort	a564a496-779e-11e9-95b5-42010aa80003	54
9-	mhn-ubuntu-1404-drupal-cm	mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot	35.235.101.217	p0f	481fe2ac-77a1-11e9-95b5-42010aa80003	3
10-	mhn-ubuntu-1404-drupal-cm	mhn-ubuntu-1404-drupal-cms-webform-bot-spam-honeypot	35.235.101.217	snort	b9953c2a-77a1-11e9-95b5-42010aa80003	0

## SENSORS

## REFERENCES

- + <https://stackoverflow.com/questions/1434451/what-does-connection-reset-by-peer-mean>
- + <https://www.sqlite.org/cli.html>
- + <https://cloud.google.com/compute/docs/instances/adding-removing-ssh-keys#edit-ssh-metadata>
- + <https://cloud.google.com/compute/docs/instances/adding-removing-ssh-keys#edit-ssh-metadata>
- + <https://stackoverflow.com/questions/20568216/python-handling-socket-error-errno-104-connection-reset-by-peer>
- + <https://cloud.google.com/resource-manager/docs/creating-managing-projects>
- + <https://cloud.google.com/vpc/docs/add-remove-network-tags>
- + <https://unix.stackexchange.com/questions/401547/gpg-keyserver-receive-failed-no-dirmngr>
- + <https://askubuntu.com/questions/347788/how-can-i-install-libpcap-header-files-on-ubuntu-12-04>

END OF REPORT