18-20 OCTOBER 2016

NOTHING IS
BEYOND
OUR POTS

HACK.LU

Dr. Honeypots

How I Learned to Stop Worrying and Know My Enemies

Hack.lu - 2016

Who am I?

Guillaume Arcas - @y0m

- Works as Security & Network Analyst since 1997 primarily - but not only - for French Internet companies. Then specialized in Digital Forensics & Incident Response and joined Sekoia’s CERT.
- Member of the Honeynet Project’s French Chapter since 2010.
- When not hunting for endangered species hanging on the Internet, uses to read (thriller, SF, History & Philosophy in no particular order as long as it is printed) and walk his dog.
- nourish a certain nostalgia for the esheep.exe software hence his Twitter’s avatar.
  https://malwr.com/analysis/NmM4ZTkyYTQzYTdhNDk2ZWI5ODE4ODdkZGZmMzU5ZDk/

A Brief History of Honeypots

1986

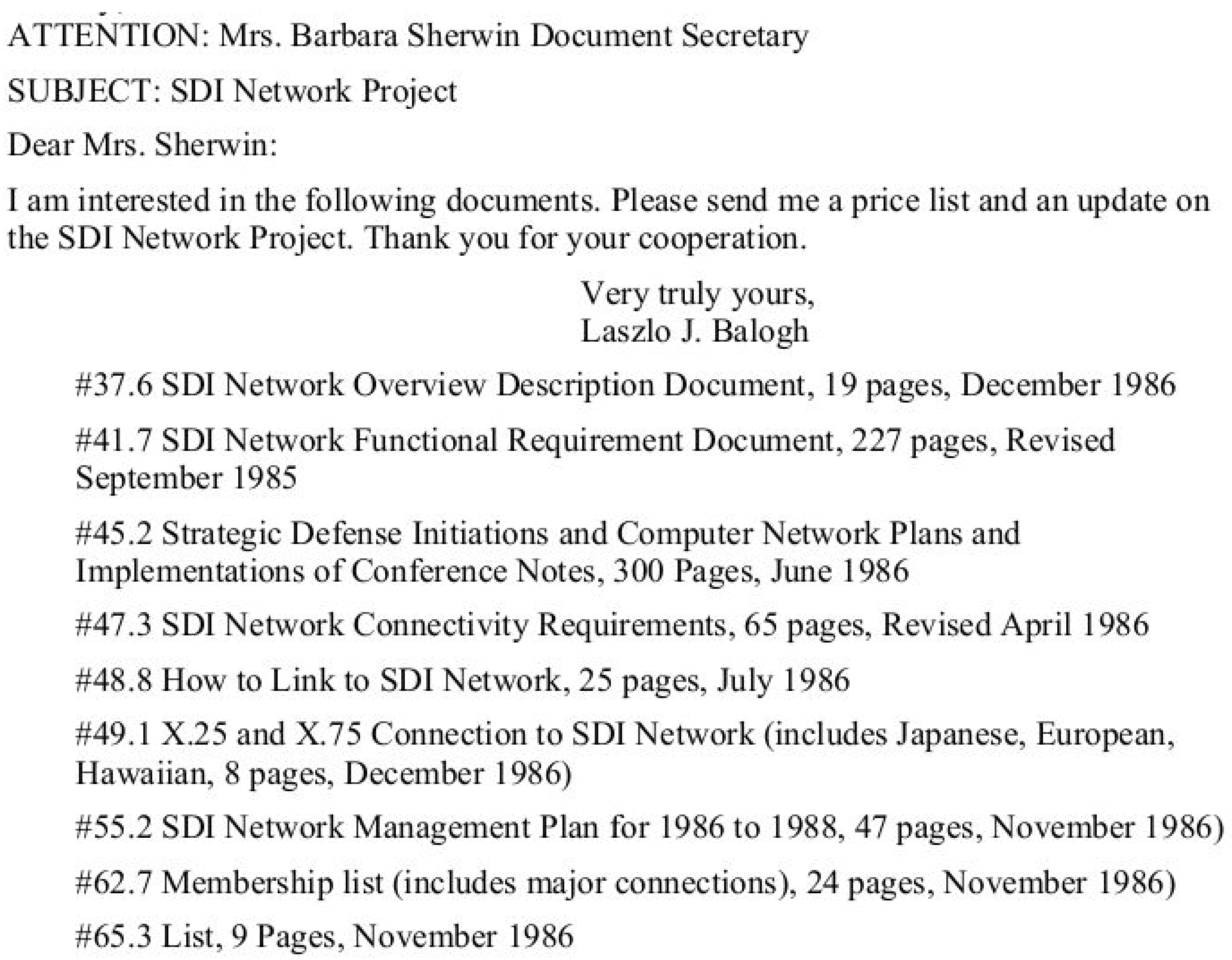
A long time ago in a network far far away…

Clifford Stoll - Astronomer & Computer Wizard

Lawrence Berkeley Lab

“And so it happened that on my second day at work, Dave wandered into my office, mumbling about a hiccup in the Unix accounting system. Someone must have used a few seconds of computing time without paying for it. The computer's books didn't quite balance; last month's bills of $2,387 showed a **75-cent shortfall**.”

At that time computers were expensive shared resources and users were charged for every cycle of computing that was used.



ATTENTION: Mrs. Barbara Sherwin Document Secretary

SUBJECT: SDI Network Project

Dear Mrs. Sherwin:

I am interested in the following documents. Please send me a price list and an update on the SDI Network Project. Thank you for your cooperation.

Very truly yours,
Laszlo J. Balogh

#37.6 SDI Network Overview Description Document, 19 pages, December 1986

#41.7 SDI Network Functional Requirement Document, 227 pages, Revised September 1985

#45.2 Strategic Defense Initiations and Computer Network Plans and Implementations of Conference Notes, 300 Pages, June 1986

#47.3 SDI Network Connectivity Requirements, 65 pages, Revised April 1986

#48.8 How to Link to SDI Network, 25 pages, July 1986

#49.1 X.25 and X.75 Connection to SDI Network (includes Japanese, European, Hawaiian, 8 pages, December 1986)

#55.2 SDI Network Management Plan for 1986 to 1988, 47 pages, November 1986)

#62.7 Membership list (includes major connections), 24 pages, November 1986)

#65.3 List, 9 Pages, November 1986

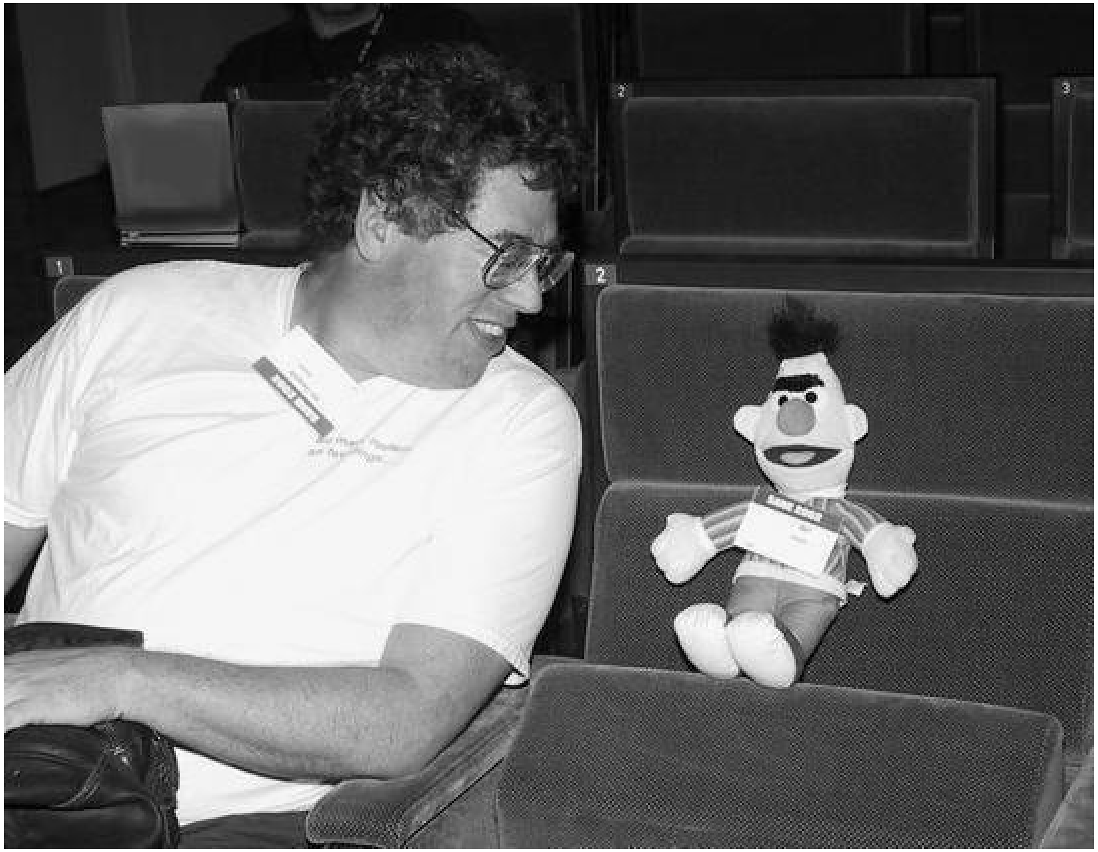
"Hey Mike, remember those carrots I left out for bait in January?"
"You mean those SDI files you concocted?"
"Yeah," I said. "Well, my dear, sweet, **nonexistent secretary** just received a letter."

WARNING
SPOILER
ALERT

“Pengo, with his contacts to hackers across Germany, knew how to use Hess's information. Carrying Hess's printouts, one of the Berlin hackers crossed into East Berlin and met with agents from the **Soviet KGB**. The deal was made: around 30,000 Deutschmarks—$18,000— for printouts and passwords.

The KGB wasn't just paying for printouts, though. Hess and company apparently sold their techniques as well: how to break into Vax computers; which networks to use when crossing the Atlantic; details on how the Milnet operates.
Even more important to the KGB was obtaining research data about Western technology, including integrated circuit design, computer-aided manufacturing, and, especially, operating system software that was under U.S. export control. They offered 250,000 Deutschmarks for copies of Digital Equipment's VMS operating system.”

1991


1
2
3
1
2

An Evening with Berferd
In Which a Cracker is Lured, Endured, and Studied

Bill Cheswick

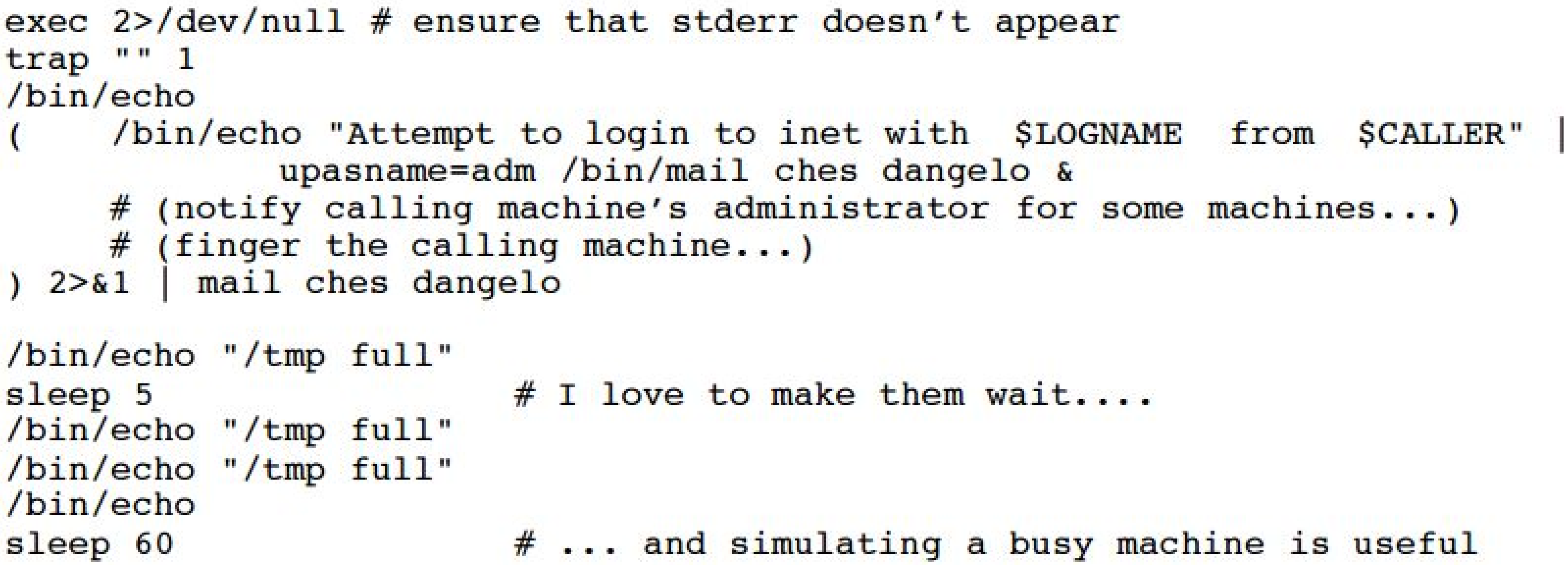
AT&T Bell Laboratories

Abstract

On 7 January 1991 a cracker, believing he had discovered the famous sendmail DEBUG hole in our Internet gateway machine, attempted to obtain a copy of our password file. I sent him one.

For several months we led this cracker on a merry chase in order to trace his location and learn his techniques. This paper is a chronicle of the cracker's "successes" and disappointments, the bait and traps used to lure and detect him, and the chroot "Jail" we built to watch his activities.

Honeypot.sh



```
exec 2>/dev/null # ensure that stderr doesn't appear
trap "" 1
/bin/echo
(     /bin/echo "Attempt to login to inet with  $LOGNAME  from  $CALLER" |
              upasname=adm /bin/mail ches dangelo &
      # (notify calling machine's administrator for some machines...)
      # (finger the calling machine...)
) 2>&1 | mail ches dangelo

/bin/echo "/tmp full"
sleep 5                  # I love to make them wait....
/bin/echo "/tmp full"
/bin/echo "/tmp full"
/bin/echo
sleep 60                 # ... and simulating a busy machine is useful
```

The following log, from 15 Jan 1991, shows decidedly unfriendly activity:

```
19:43:10 smtpd[27466]: <--- 220 inet.att.com SMTP
19:43:14 smtpd[27466]: -------> debug
19:43:14 smtpd[27466]: DEBUG attempt
19:43:14 smtpd[27466]: <--- 200 OK
19:43:25 smtpd[27466]: -------> mail from:</dev/null>
19:43:25 smtpd[27466]: <--- 503 Expecting HELO
19:43:34 smtpd[27466]: -------> helo
19:43:34 smtpd[27466]: HELO from
19:43:34 smtpd[27466]: <--- 250 inet.att.com
19:43:42 smtpd[27466]: -------> mail from: </dev/null>
19:43:42 smtpd[27466]: <--- 250 OK
19:43:59 smtpd[27466]: -------> rcpt to:</dev/^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H
19:43:59 smtpd[27466]: <--- 501 Syntax error in recipient name
19:44:44 smtpd[27466]: -------> rcpt to:<|sed -e ’1,/^$/’d | /bin/sh ; exit 0">
19:44:44 smtpd[27466]: shell characters: |sed -e ’1,/^$/’d | /bin/sh ; exit 0"
19:44:45 smtpd[27466]: <--- 250 OK
19:44:48 smtpd[27466]: -------> data
19:44:48 smtpd[27466]: <--- 354 Start mail input; end with <CRLF>.<CRLF>
19:45:04 smtpd[27466]: <--- 250 OK
19:45:04 smtpd[27466]: /dev/null  sent 48 bytes to  upas.security
19:45:08 smtpd[27466]: -------> quit
19:45:08 smtpd[27466]: <--- 221 inet.att.com Terminating
19:45:08 smtpd[27466]: finished.
```

1992

DRAGONS

There Be Dragons

Steven M. Bellovin
AT&T Bell Laboratories
Murray Hill, NJ
smb@ulysses.att.com

July 30, 1992

Abstract

Our security gateway to the Internet, `research.att.com`, provides only a limited set of services. Most of the standard servers have been replaced by a variety of trap programs that look for attacks. Using these, we have detected a wide variety of pokes, ranging from simple doorknob-twisting to determined assaults. The attacks range from simple attempts to log in as `guest` to forged NFS packets. We believe that many other sites are being probed but are unaware of it: the standard network daemons do not provide administrators with either appropriate controls and filters or with the logging necessary to detect attacks.

8 Conclusions

"Never laugh at live dragons, Bilbo you fool!" he said to himself.

J.R.R. Tolkien, *The Hobbit*

It is all well and good to decry computer security, and to preach the religion of open access. Unfortunately, there are an increasing number of people with access to the Internet who do not share the morality necessary to make such schemes work. One can assume that one is being attacked; the only questions are how, and how often. (Just who the attackers are is in some sense uninteresting; if one group passes on, another is sure to take its place.)

Our goal is to provide information to the community, and to the proper authorities, on just how the crackers are operating. Our specific methods are not for everyone, but our lessons — and our warnings — are.

1999

Sun

HN/P

The Honeynet Project

The Honeynet Project is a leading international 501c3 non-profit security research organization, dedicated to investigating the latest attacks and developing open source security tools to improve Internet security.
With Chapters around the world, our volunteers have contributed to fight against malware (such as Conficker), discovering new attacks and creating security tools used by businesses and government agencies all over the world.

Our mission reads "to learn the tools, tactics and motives involved in computer and network attacks, and share the lessons learned" with three main pillars:

- Research
- Awareness
- Tools

http://www.honeynet.org/about

Everything You Always Wanted to Know About Honeypots But Were Afraid to Ask

What is a Honeypot?

tl;dr

Shortly said: it’s a trap!

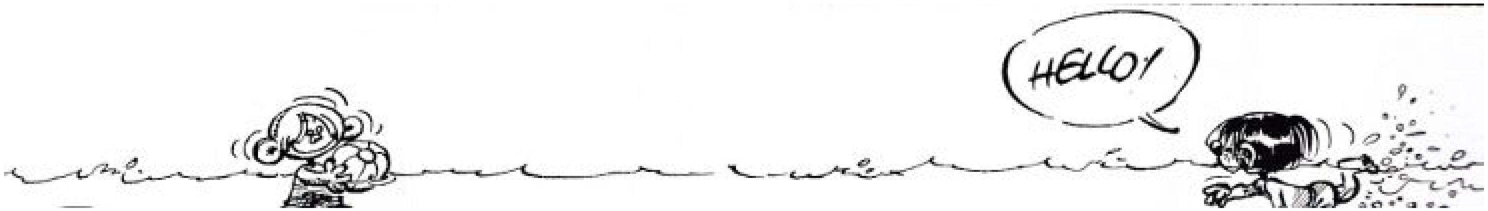
But it is a special trap designed not to catch & kill the mouse but to gather information from her:

- the **Technics** she uses to discover the cheese;
- the **Tools** she uses to get to the cheese;
- the **Protocols** she uses to take the cheese out of the kitchen;
- The kind of cheese she likes the most.

Then, once you know enough about mouse’s TTPs, you can adjust your defenses to catch & kill her !

Disclaimer: no real mouse was harmed in the making of this slide.

Looks innocuous...



But it can bite!

Honeynet Project Definition (2002)

"A honeypot is a single system connected to an existing production network in order to lure attackers."

Honeynet Project Definition (2004)

"A honeypot is a information system resource whose value lies in unauthorized or illicit use of that resource."

ENISA Definition (2012)

"A honeypot is a **computing resource** whose sole task is to be probed, attacked, compromised, used or accessed in any other unauthorized way. The resource can be **of any type**: a service, an application, a system or a set of systems or simply just a piece of information or data."

What is a Honeynet?

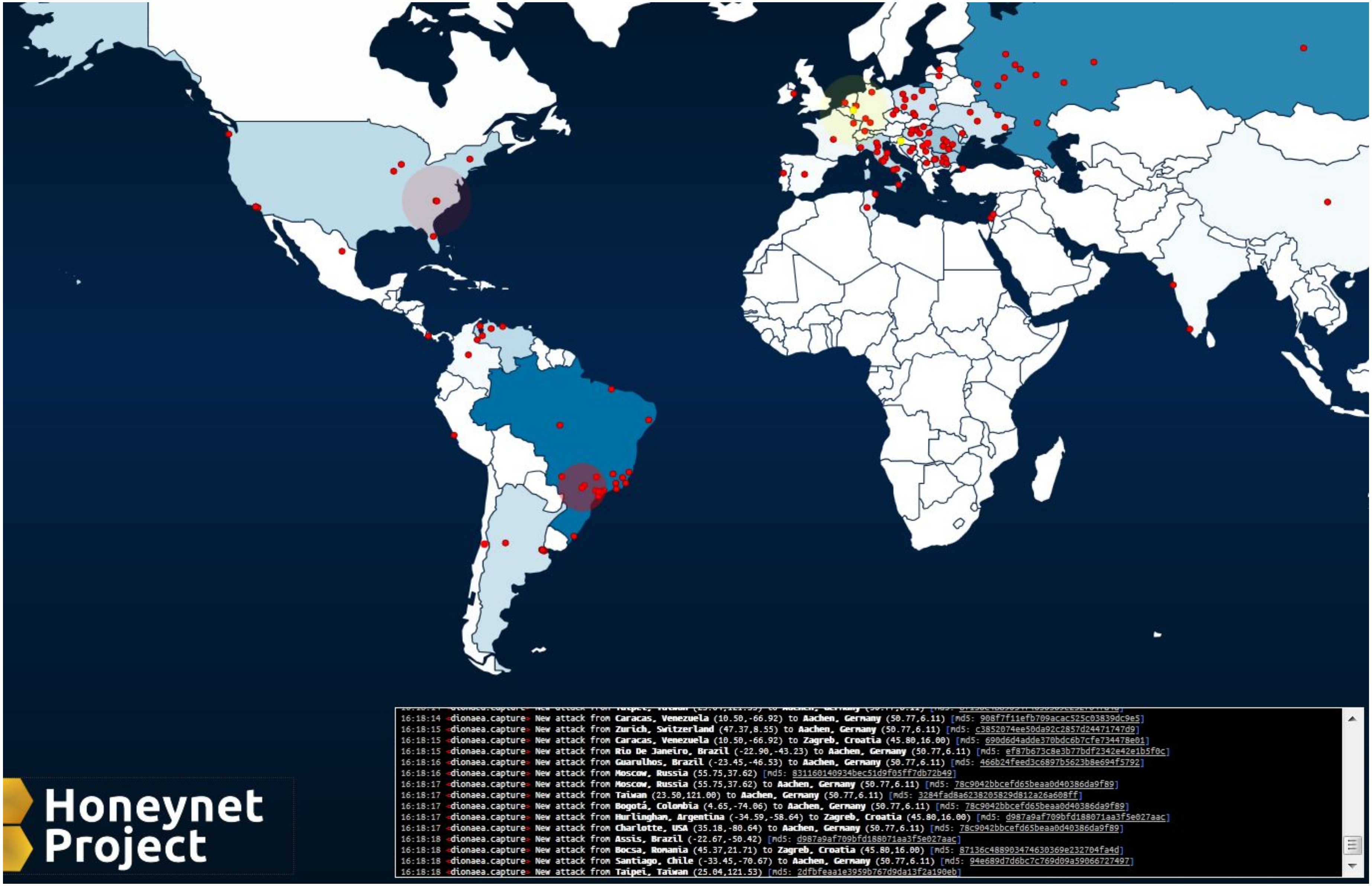
Where?

On the Internet:

- it will generate and collect a lot of noise and often useless information ;
- it can be seen as a metrics of the threat level from the North of the Wall;
- it can help convince the top-management not to decrease IT Security budget.

On the Internet:

- Trends :
  - What vulnerabilites are the most exploited?
  - How soon after their disclosure are they tested/searched?
  - It can help assign priorities
  - It can help to adapt the Patch Management policy



```
16:18:14 <dionaea.capture> New attack from Caracas, Venezuela (10.50,-66.92) to Aachen, Germany (50.77,6.11) [md5: 908f7f11efb709acac525c03839dc9e5]
16:18:15 <dionaea.capture> New attack from Zurich, Switzerland (47.37,8.55) to Aachen, Germany (50.77,6.11) [md5: c3852074ee50da92c2857d24471747d9]
16:18:15 <dionaea.capture> New attack from Caracas, Venezuela (10.50,-66.92) to Zagreb, Croatia (45.80,16.00) [md5: 690d6d4adde370bdc6b7cfe734478e01]
16:18:15 <dionaea.capture> New attack from Rio De Janeiro, Brazil (-22.90,-43.23) to Aachen, Germany (50.77,6.11) [md5: ef87b673c8e3b77bdf2342e42e1b5f0c]
16:18:16 <dionaea.capture> New attack from Guarulhos, Brazil (-23.45,-46.53) to Aachen, Germany (50.77,6.11) [md5: 466b24feed3c6897b5623b8e694f5792]
16:18:16 <dionaea.capture> New attack from Moscow, Russia (55.75,37.62) [md5: 831160140934bec51d9f05ff7db72b49]
16:18:17 <dionaea.capture> New attack from Moscow, Russia (55.75,37.62) to Aachen, Germany (50.77,6.11) [md5: 78c9042bbcefd65beaa0d40386da9f89]
16:18:17 <dionaea.capture> New attack from Taiwan (23.50,121.00) to Aachen, Germany (50.77,6.11) [md5: 3284fad8a6238205829d812a26a608ff]
16:18:17 <dionaea.capture> New attack from Bogotá, Colombia (4.65,-74.06) to Aachen, Germany (50.77,6.11) [md5: 78c9042bbcefd65beaa0d40386da9f89]
16:18:17 <dionaea.capture> New attack from Hurlingham, Argentina (-34.59,-58.64) to Zagreb, Croatia (45.80,16.00) [md5: d987a9af709bfd188071aa3f5e027aac]
16:18:17 <dionaea.capture> New attack from Charlotte, USA (35.18,-80.64) to Aachen, Germany (50.77,6.11) [md5: 78c9042bbcefd65beaa0d40386da9f89]
16:18:18 <dionaea.capture> New attack from Assis, Brazil (-22.67,-50.42) [md5: d987a9af709bfd188071aa3f5e027aac]
16:18:18 <dionaea.capture> New attack from Bocsa, Romania (45.37,21.71) to Zagreb, Croatia (45.80,16.00) [md5: 87136c488903474630369e232704fa4d]
16:18:18 <dionaea.capture> New attack from Santiago, Chile (-33.45,-70.67) to Aachen, Germany (50.77,6.11) [md5: 94e689d7d6bc7c769d09a59066727497]
16:18:18 <dionaea.capture> New attack from Taipei, Taiwan (25.04,121.53) [md5: 2dfbfeaa1e3959b767d9da13f2a190eb]
```

On internal network:

- if something happens then sh*t hit the fan!
- Early Detection Systems for CERT/DFIR teams ;
- If something happens there, no need to argue, no time to lose: you are in trouble and need to investigate.

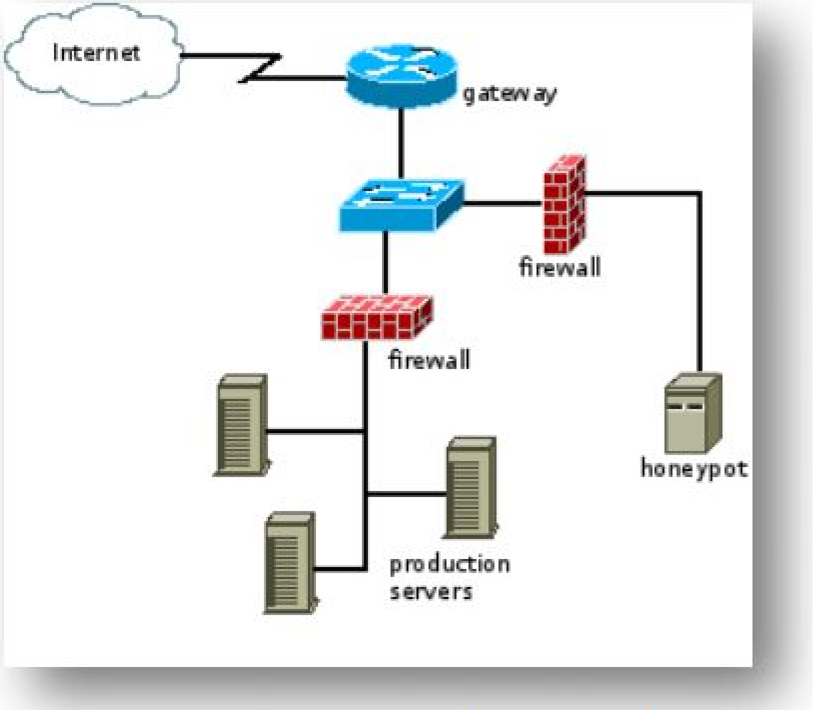


Figure 3: Typical honeypot deployment facing the Internet

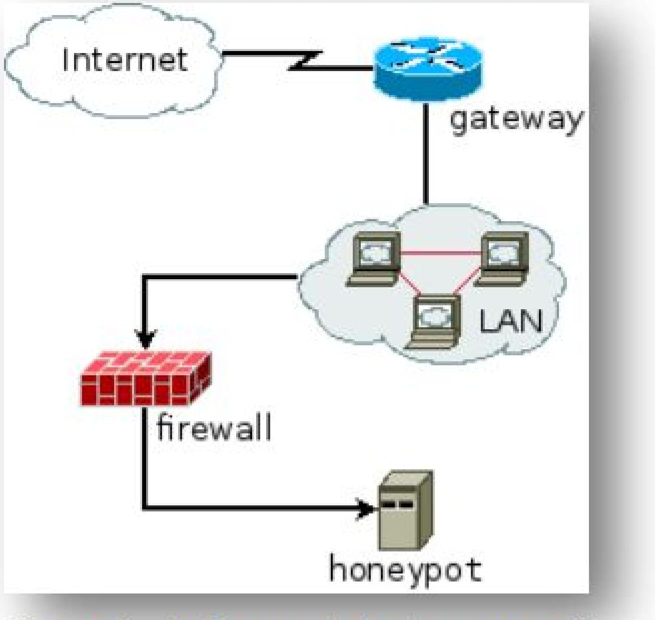


Figure 4: An internal deployment of a honeypot

https://www.enisa.europa.eu/activities/cert/support/proactive-detection/proactive-detection-of-security-incidents-II-honeypots

Taxonomy

Type of attacked resource

- Server-side honeypot
- Client-side honeypot (honeyclient)

Level of interaction

- low-interaction: emulated system
- high-interaction: real system
- hybrid: mix of low & high

Low interaction

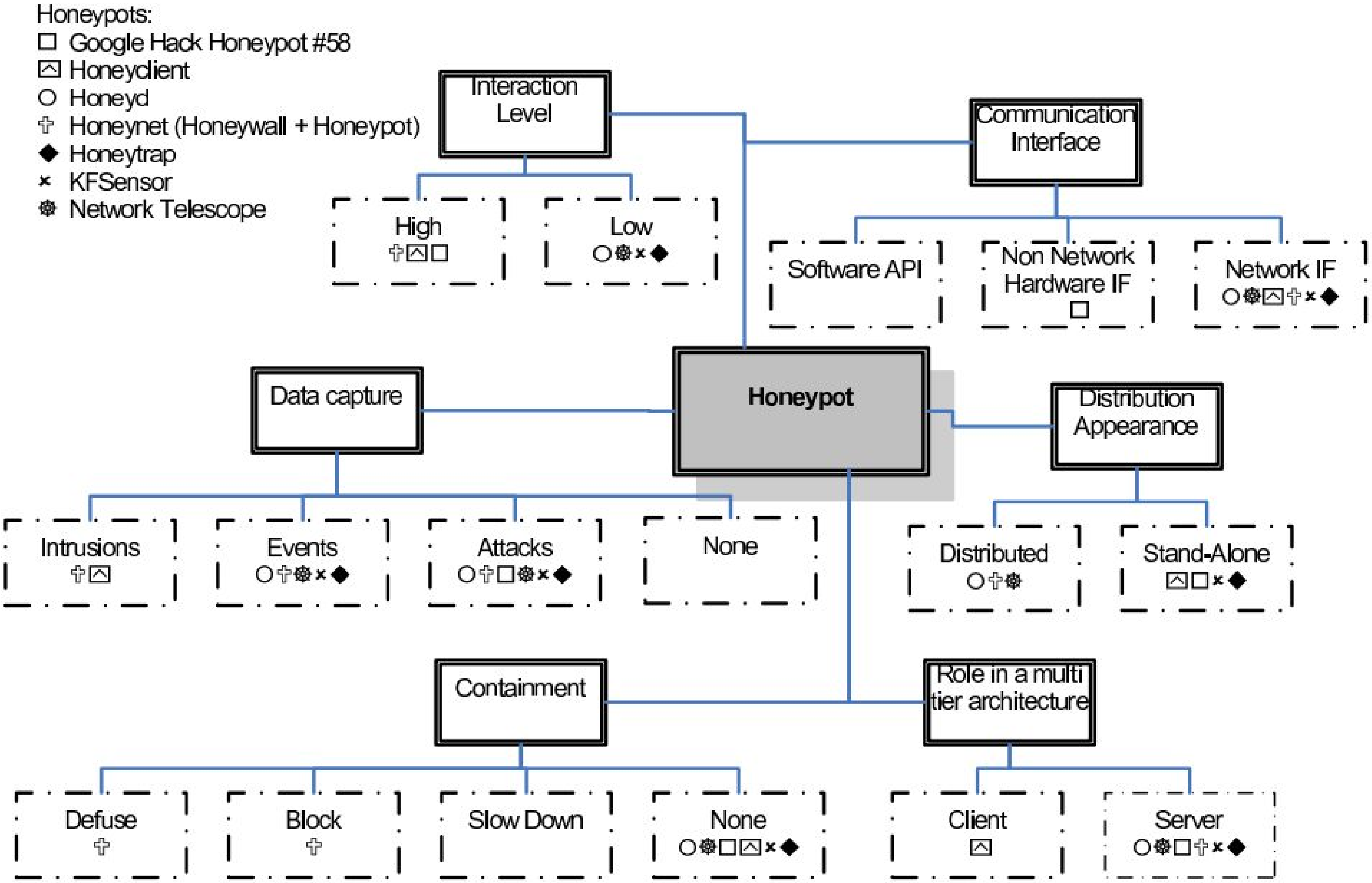
- Emulates a system
- Less risky: you control what the attacker can do
- Easier to deploy
- As attackers are limited in what they can do, it provides less information
- Can only capture known attacks

High interaction

- Real & full-featured system
- More risky: you may not be able to control what the attacker can do
- More complex
- Can capture unknown exploits

Hybrid honeypots

- Combine both low-interaction and high-interaction tools in order to gain the benefits of both.
- Example: HoneySpider Network: a low-interaction honeyclient filters out benign websites, while all others (suspicious or malicious) are analysed with high-interaction honeyclients.



Honeypots:

- □ Google Hack Honeypot #58
- ⍓ Honeyclient
- ○ Honeyd
- ✝ Honeynet (Honeywall + Honeypot)
- ◆ Honeytrap
- × KFSensor
- ✺ Network Telescope

Honeypot

Interaction Level
- High ✝⍓□
- Low ○✺×◆

Communication Interface
- Software API
- Non Network Hardware IF □
- Network IF ○✺⍓✝×◆

Data capture
- Intrusions ✝⍓
- Events ○✝✺×◆
- Attacks ○✝□✺×◆
- None

Distribution Appearance
- Distributed ○✝✺
- Stand-Alone ⍓□×◆

Containment
- Defuse ✝
- Block ✝
- Slow Down
- None ○✺□⍓×◆

Role in a multi tier architecture
- Client ⍓
- Server ○✺□✝×◆

Figure 1: Honeypot Taxonomy

http://www.mcs.vuw.ac.nz/comp/Publications/archive/CS-TR-06/CS-TR-06-12.pdf

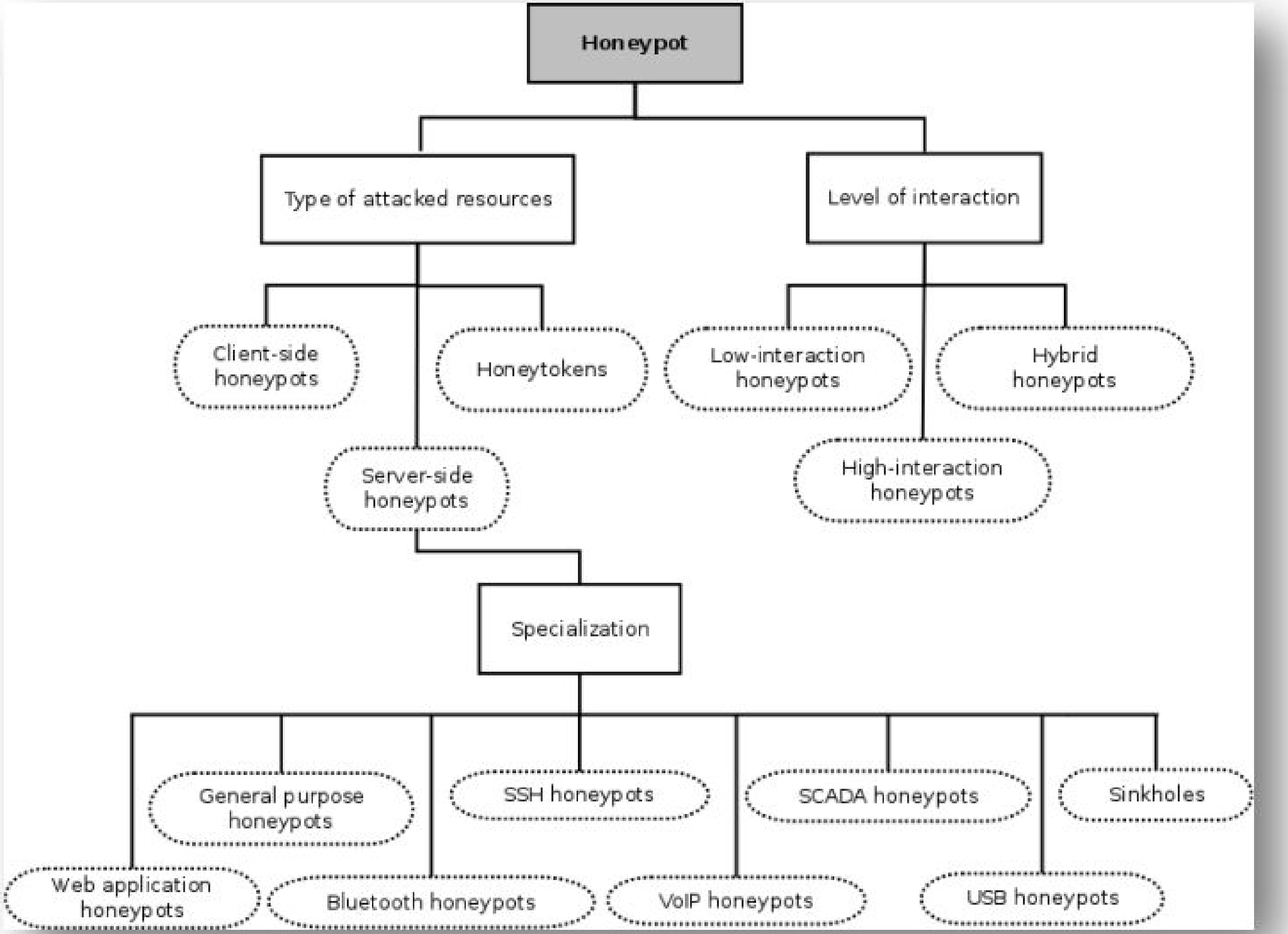


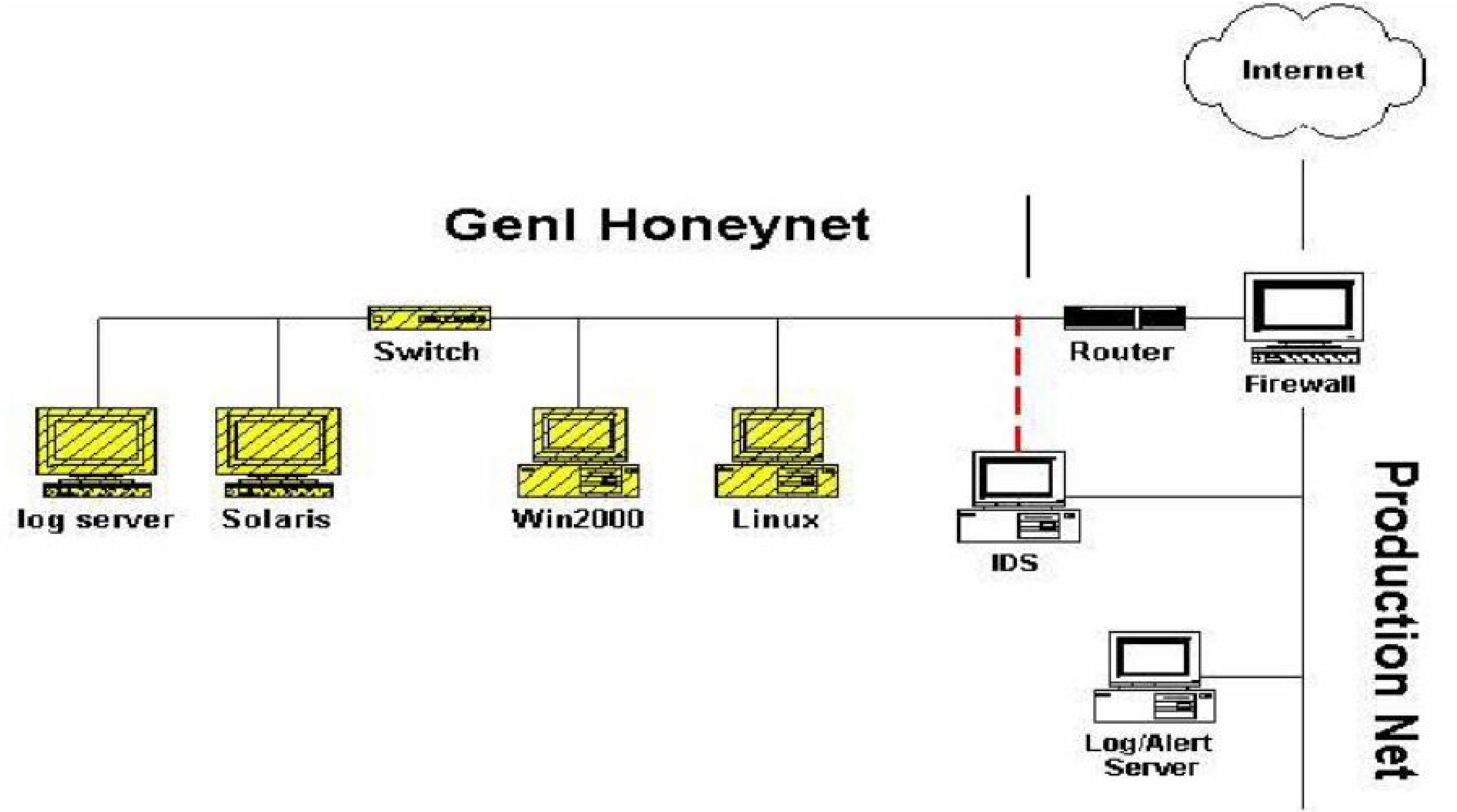
Figure 2: Graphical representation of the classification scheme of taxonomy used in the report

https://www.enisa.europa.eu/activities/cert/support/proactive-detection/proactive-detection-of-security-incidents-II-honeypots

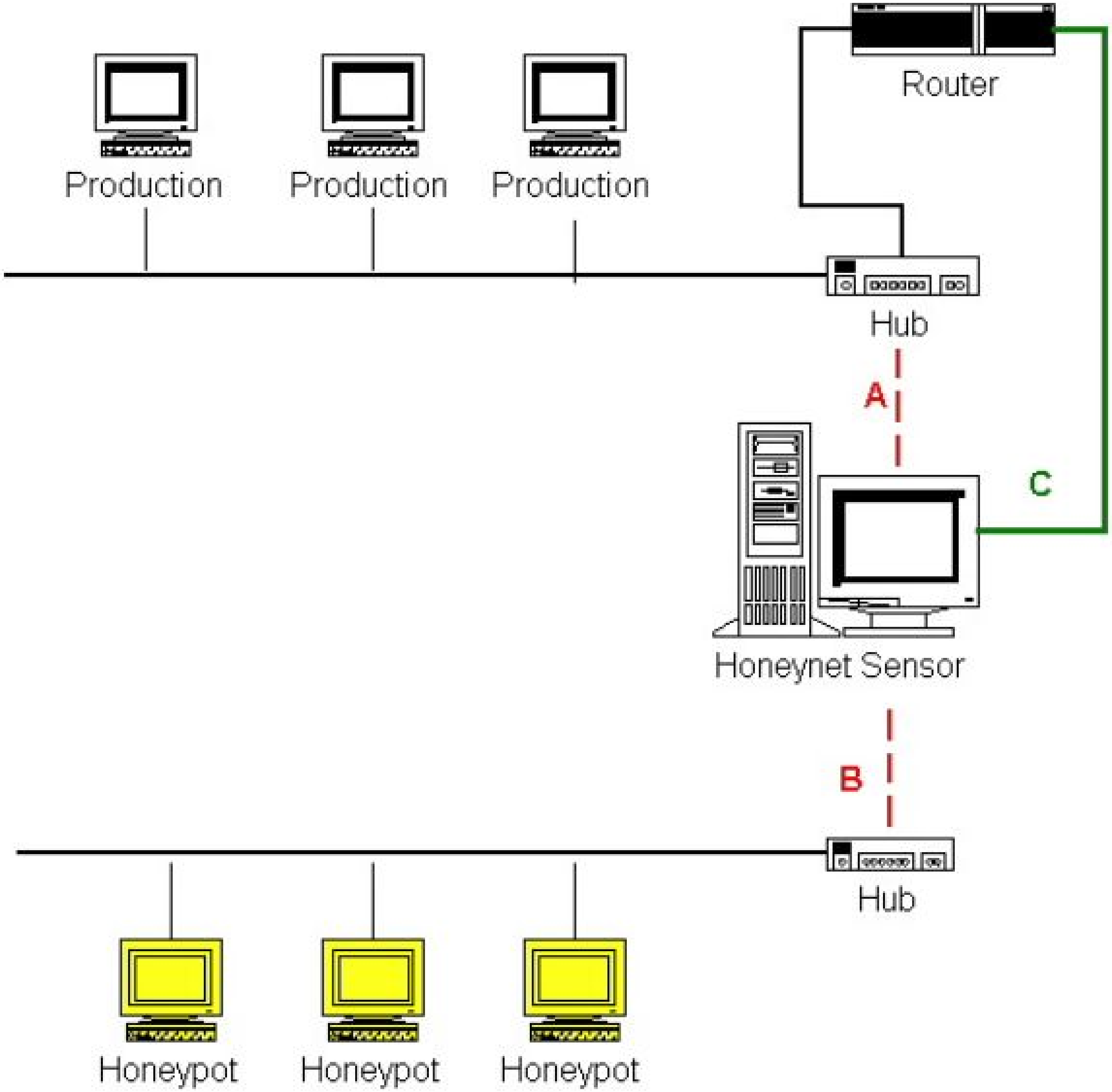
Taxonomy of honeynets

- Gen1 : network of honeypots
- Gen2 : honeypots in a production environment, for example deployed on a dedicated subnet.
    - Honeywall used for routing and filtering attacks.
- Virtual honeynets
- Distributed honeynets
    - HPFeeds/HPFriends for data sharing

Gen1 honeynet



Gen2 honeynet



Honeynet Sensor Diagram

Sensor consists of a single system functioning as both Data Control and Data Capture requirements.

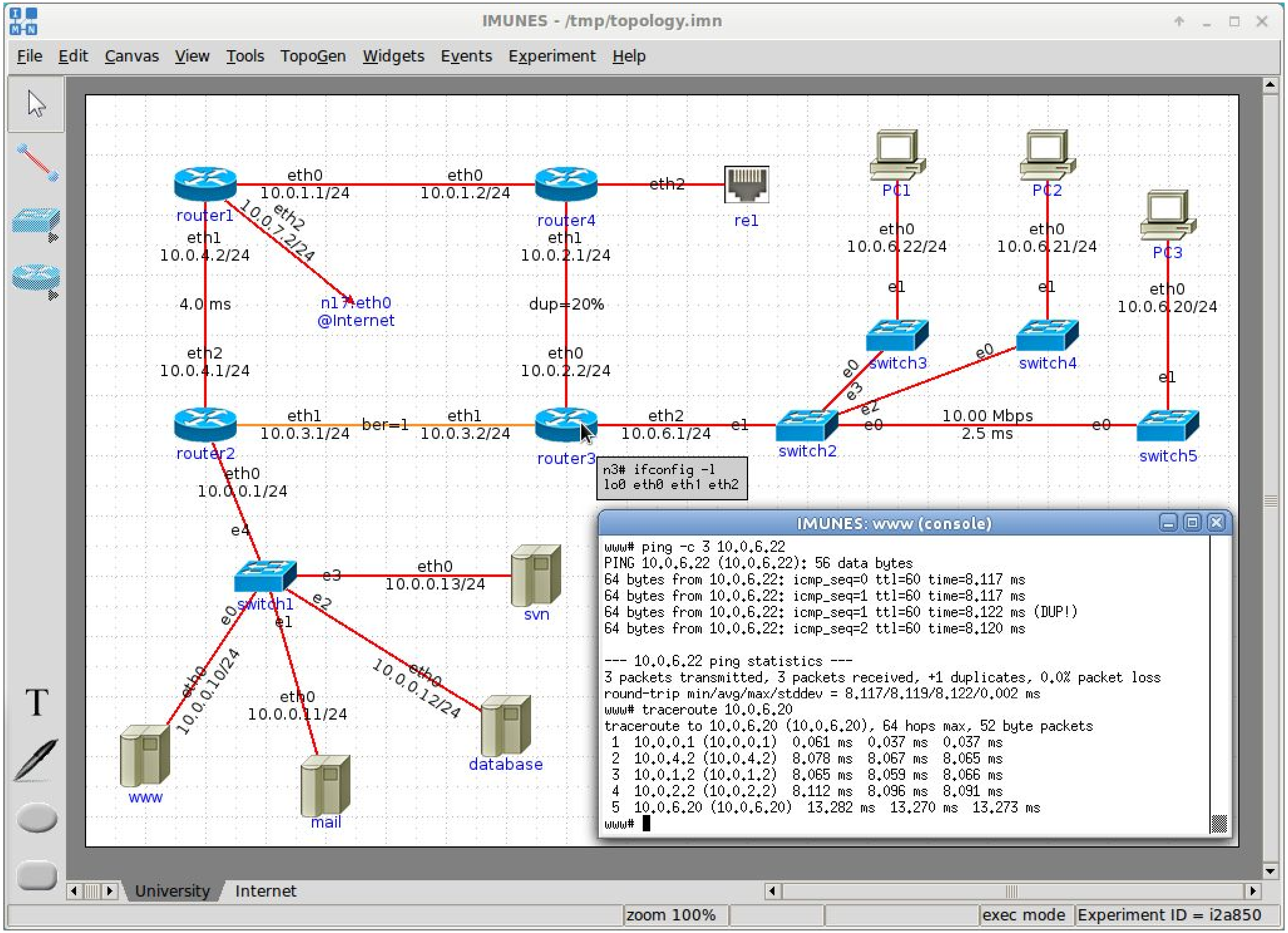
It consists of three interfaces. Two of the interfaces are layer2 (outlined in RED), acting as a switch which segments a production network. The third interface has an IP stack for remote connectivity. This is for both Data Collection and administation.

Interface A: Layer2 interface segmenting production network.

Interface B: Layer 2 interface segmenting Honeynet network.

Intereface C: Layer3 interface VPN connection to collection point.

IMUNES



```
n3# ifconfig -l
lo0 eth0 eth1 eth2
```

```
www# ping -c 3 10.0.6.22
PING 10.0.6.22 (10.0.6.22): 56 data bytes
64 bytes from 10.0.6.22: icmp_seq=0 ttl=60 time=8.117 ms
64 bytes from 10.0.6.22: icmp_seq=1 ttl=60 time=8.117 ms
64 bytes from 10.0.6.22: icmp_seq=1 ttl=60 time=8.122 ms (DUP!)
64 bytes from 10.0.6.22: icmp_seq=2 ttl=60 time=8.120 ms

--- 10.0.6.22 ping statistics ---
3 packets transmitted, 3 packets received, +1 duplicates, 0.0% packet loss
round-trip min/avg/max/stddev = 8.117/8.119/8.122/0.002 ms
www# traceroute 10.0.6.20
traceroute to 10.0.6.20 (10.0.6.20), 64 hops max, 52 byte packets
 1  10.0.0.1 (10.0.0.1)  0.061 ms  0.037 ms  0.037 ms
 2  10.0.4.2 (10.0.4.2)  8.078 ms  8.067 ms  8.065 ms
 3  10.0.1.2 (10.0.1.2)  8.065 ms  8.059 ms  8.066 ms
 4  10.0.2.2 (10.0.2.2)  8.112 ms  8.096 ms  8.091 ms
 5  10.0.6.20 (10.0.6.20)  13.282 ms  13.270 ms  13.273 ms
www# █
```

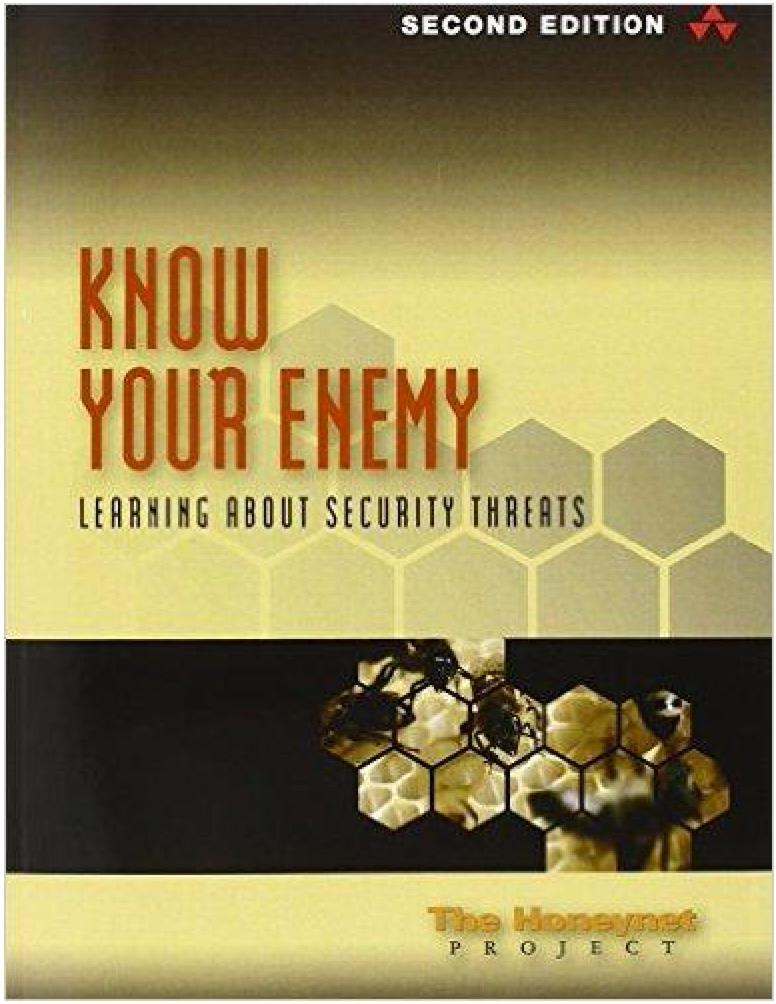
Further reading

Proactive Detection of Security Incidents

Honeypots

2012-11-20

https://www.enisa.europa.eu/activities/cert/support/proactive-detection/proactive-detection-of-security-incidents-II-honeypots


SECOND EDITION
KNOW YOUR ENEMY
LEARNING ABOUT SECURITY THREATS
The Honeynet PROJECT

Why?

Boeing E-3 Sentry

- Designed to passively detect High and Low, Far and close threats.

Early Awareness & Detection System with Reduced False Positives

In a production environment, some events may be suspicious.

Someone successfully connects to a server at unusual time from India:

- it can be your newly appointed offshore IT management service provider performing usual tasks;
- it can be a SysAdmin connecting from his/her vacation place because of an emergency.

… Or one of these guys.

In a honeypot or a honeynet environment, all events are suspicious by nature.

Someone successfully connects to a honeypot from anywhere at any time:

- it can be an intruder performing lateral movements;
- it can be an insider or a too curious authorized user;
- it can be your internal Red Team.

… Or one of these guys.

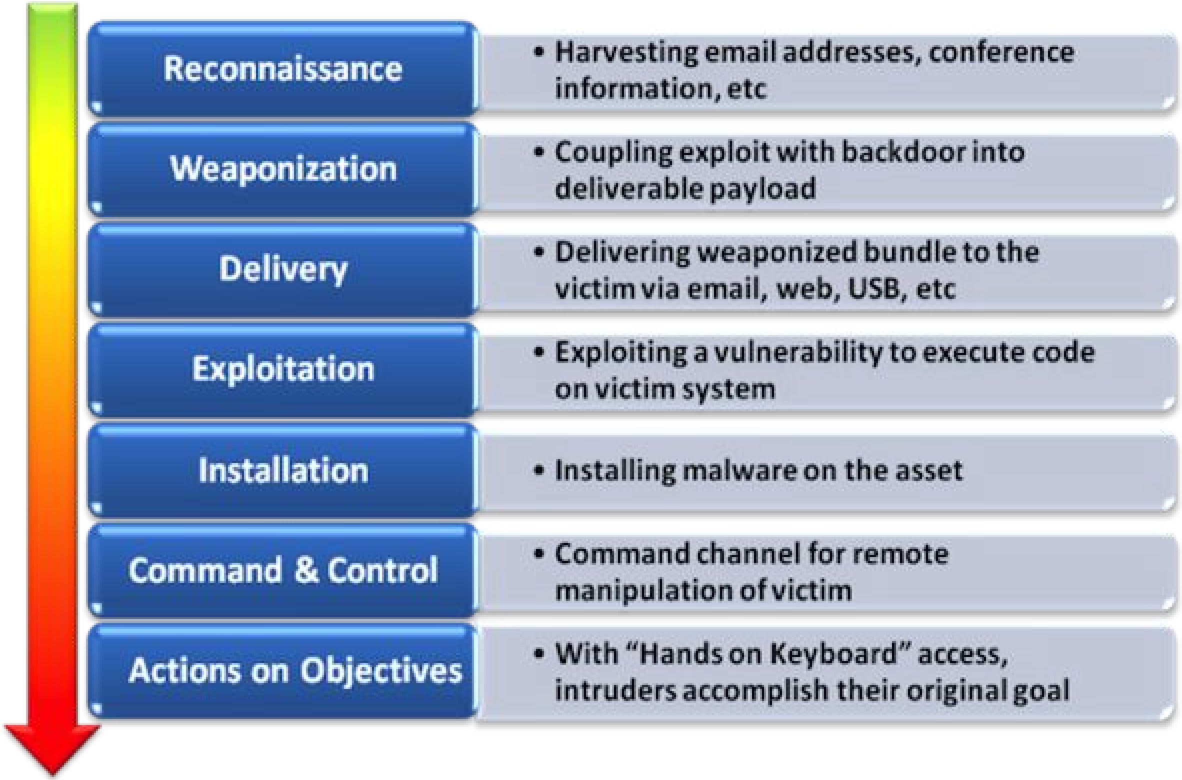
In a production environment, you can not monitor/log/store everything:

- cost & storage constraints
- legal constraints
- Technically complex
- Need to be able to analyze huge volume of data

In a honeypot or honeynet, you must and can monitor/log/store everything:

- network traffic
- uploaded files
- system logs

Honeypots & the Intrusion Kill Chain



Reconnaissance	• Harvesting email addresses, conference information, etc
Weaponization	• Coupling exploit with backdoor into deliverable payload
Delivery	• Delivering weaponized bundle to the victim via email, web, USB, etc
Exploitation	• Exploiting a vulnerability to execute code on victim system
Installation	• Installing malware on the asset
Command & Control	• Command channel for remote manipulation of victim
Actions on Objectives	• With “Hands on Keyboard” access, intruders accomplish their original goal

A honeypot can drastically help detecting adversary’s Reconnaissance actions.

Counter-OSINT:

- A fake LinkedIn profile, Facebook page, email addresses published on corporate website (can be hidden in HTML comments so not visible from usual visitors), fake "leaked credentials" on pastebin, fake DB dumps posted on underground forums, etc. can increase visibility on how the attacker found his/her targets.
- Fake password hash loaded in memory to detect password stealers like Mimikatz.

How?

Critical points

- Monitor/Collect/Store Data
- Allow/Forbid/Restrict access to the Internet
- Do you hide your honeypot or do you make it public (DNS domain, public IP, etc)?

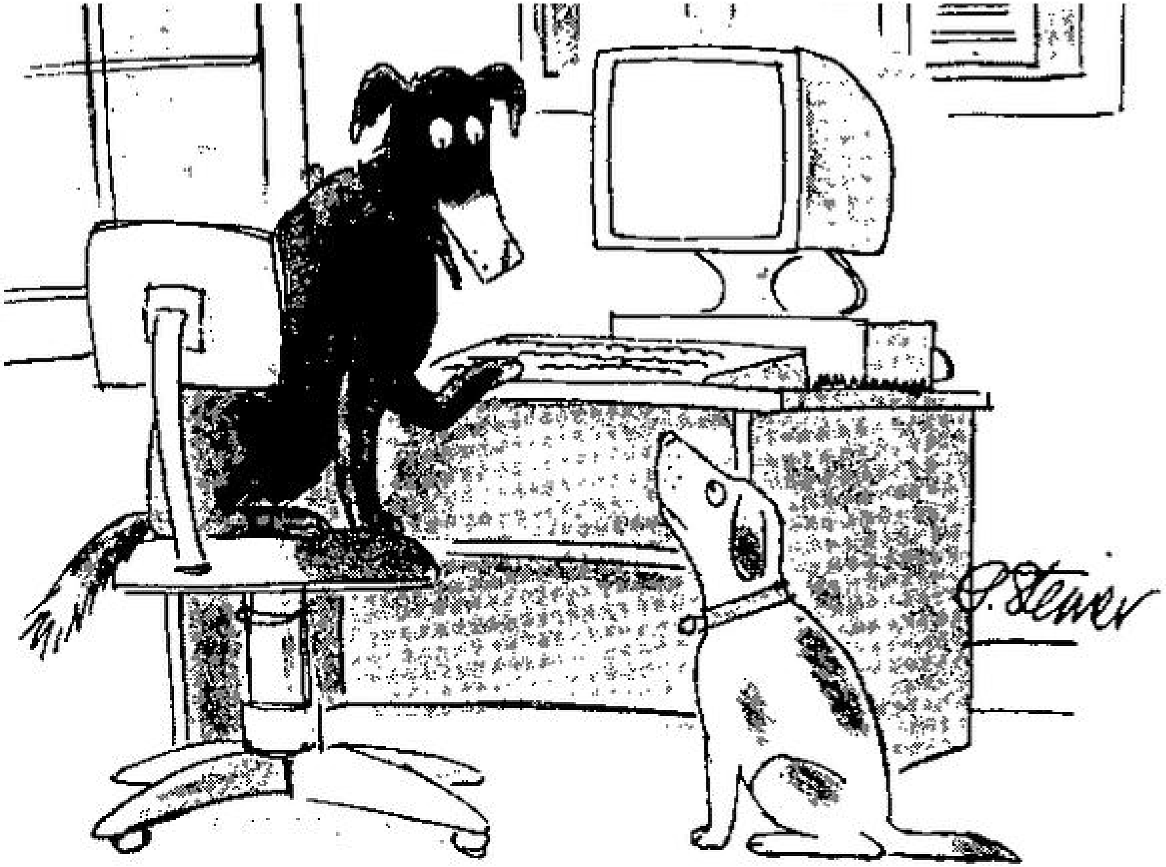
Collecting Data

- You’ll have to answer this question: “How can I monitor an intruder with privileged access (aka: root/administrator|system users rights) without being detected/defeated?

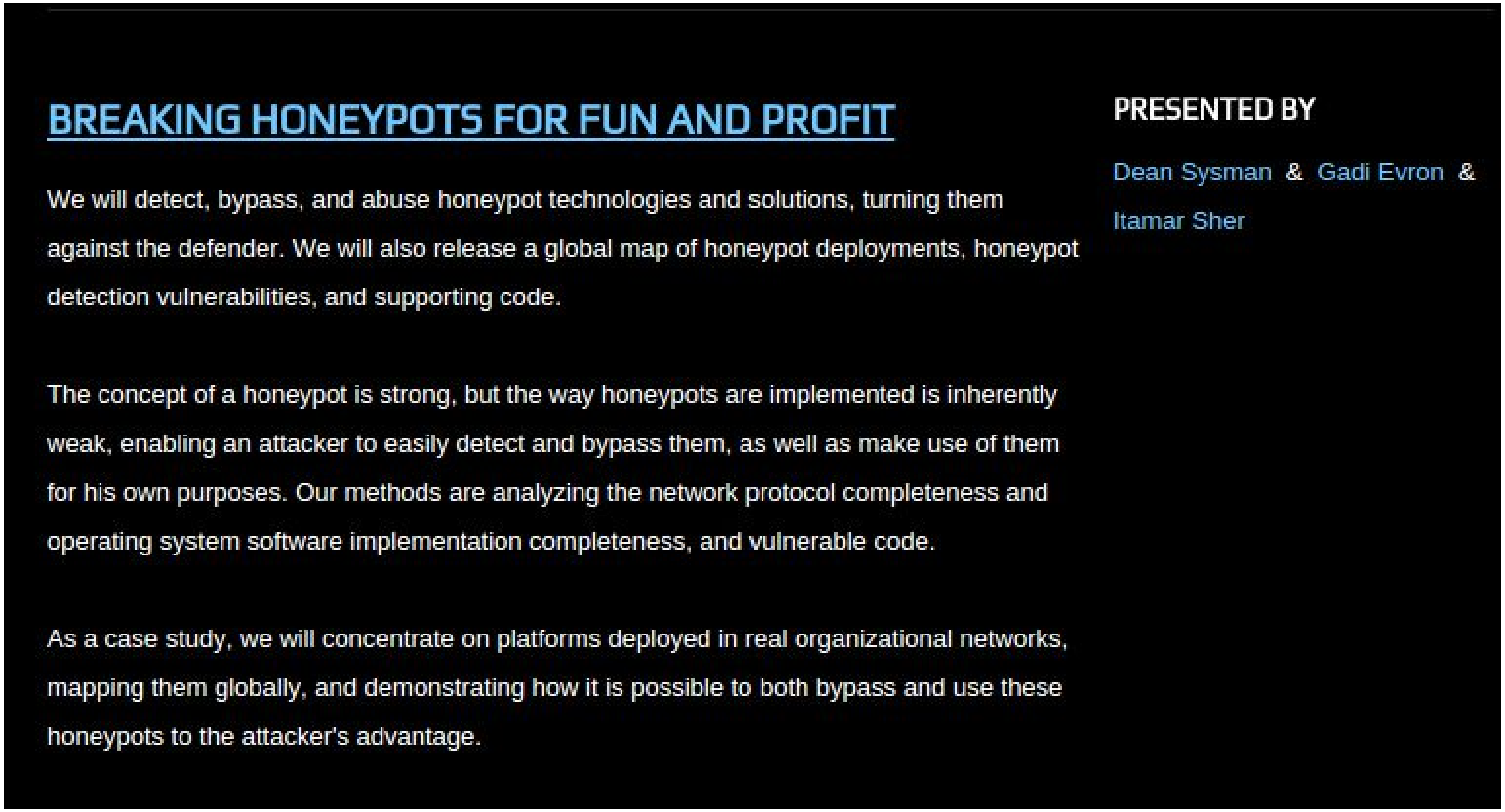
Internet Access

- What kind of Internet access will you grant from the honeypot? If Internet access is too limited, the intruder can find no interest in staying any longer.

Avoid Detection



“On the Internet, nobody knows you're a **pot**”



BREAKING HONEYPOTS FOR FUN AND PROFIT

We will detect, bypass, and abuse honeypot technologies and solutions, turning them against the defender. We will also release a global map of honeypot deployments, honeypot detection vulnerabilities, and supporting code.

The concept of a honeypot is strong, but the way honeypots are implemented is inherently weak, enabling an attacker to easily detect and bypass them, as well as make use of them for his own purposes. Our methods are analyzing the network protocol completeness and operating system software implementation completeness, and vulnerable code.

As a case study, we will concentrate on platforms deployed in real organizational networks, mapping them globally, and demonstrating how it is possible to both bypass and use these honeypots to the attacker's advantage.

PRESENTED BY

Dean Sysman & Gadi Evron & Itamar Sher

Skills

What skills do you need?

- Network Forensics
- System Forensics
- Reverse Engineering
- Data Analysis
- Coding

Honeypots Arsenal

High-Interaction Server-Side Honeypots

- Argos
- HiHAT
- SSH: Bifrozt, DockPot, HonSSH

Low-Interaction Server-Side Honeypots

- General purpose: Dionaea, Honeyd, Honeytrap
- Web Application: Glastopf, GoogleHack Honeypot
- SSH: Kippo/Cowrie
- Scada: ConPot
- VoIP: Atermisa
- Sinkholes: HoneySink
- USB: Ghost USB honeypot

High-Interaction Client-Side Honeypots

- Shelia
- Capture-HPC NG

Low-Interaction Client-Side Honeypots

- Thug
- PhonyeC

Hybrid Honeypots

- HoneySpider
- SURFcert IDS
- SSH: Bifrozt

Honeytokens

- a honeytoken is a piece of data that should not be accessed through normal activity, i.e. does not have any production value, any access must be intentional, which means it is likely to be an unauthorised act. (ENISA)
- http://www1.cs.columbia.edu/~angelos/Papers/2009/DecoyDocumentsSECCOM09.pdf
- http://seclists.org/focus-ids/2003/Feb/95

“OTS” Honeypots

- http://www.honeynet.org/project

Other Tools

- **APKInspector**: static analysis platform for android applications.
- **Cuckoo Sandbox**: automated dynamic analysis sandbox. Powering malwr.com website.
- **Droidbox**: dynamic analysis platform for Android applications

HN/P

KEEP
CALM
AND
TAKE
A BREAK

First steps with a honeypot

Kippo

Kippo is a low-interaction server honeypot emulating the Secure Shell (SSH) service. It stores information about brute-force login attacks against the service and SSH session & actions the attacker launched against the server.

Kippo

According to ENISA:

> “Kippo is extremely useful because, in addition to the detection of simple brute-force attacks against SSH, it also allows you to gather data from terminal session activity of an attacker in the emulated environment and to catch files downloaded by the attacker.”

DETECTION SCOPE	ACCURACY OF EMULATION	QUALITY OF COLLECTED DATA	SCALABILITY AND PERFORMANCE	RELIABILITY	EXTENSIBILITY	EASE OF USE AND SETTING UP	EMBEDDABILITY	SUPPORT	COST	USEFULNESS FOR CERT
SPEC	★★★	★★★	★★	★★★	★★	★★★	★★	★★★	$$	😁

Detection scope		**Rating**		**Cost**		**Usefulness for CERT**	
MULTI	Multi-function	★★★★	Excellent	$	Low	😁	Essential
		★★★	Good	$$	Medium	🙂	Useful
SPEC	Specialised	★★	Fair	$$$	High	☹	Not useful
		★	Poor				

Version tested: 0.5

Date tested: 27 April 2012

Testing time: 24 hours

Website: http://code.google.com/p/kippo/

Cowrie

- Kippo’s developement stopped 2 years ago.
- Cowrie is developed by Michel Oosterhof and is based on Kippo.
- https://github.com/micheloosterhof/cowrie

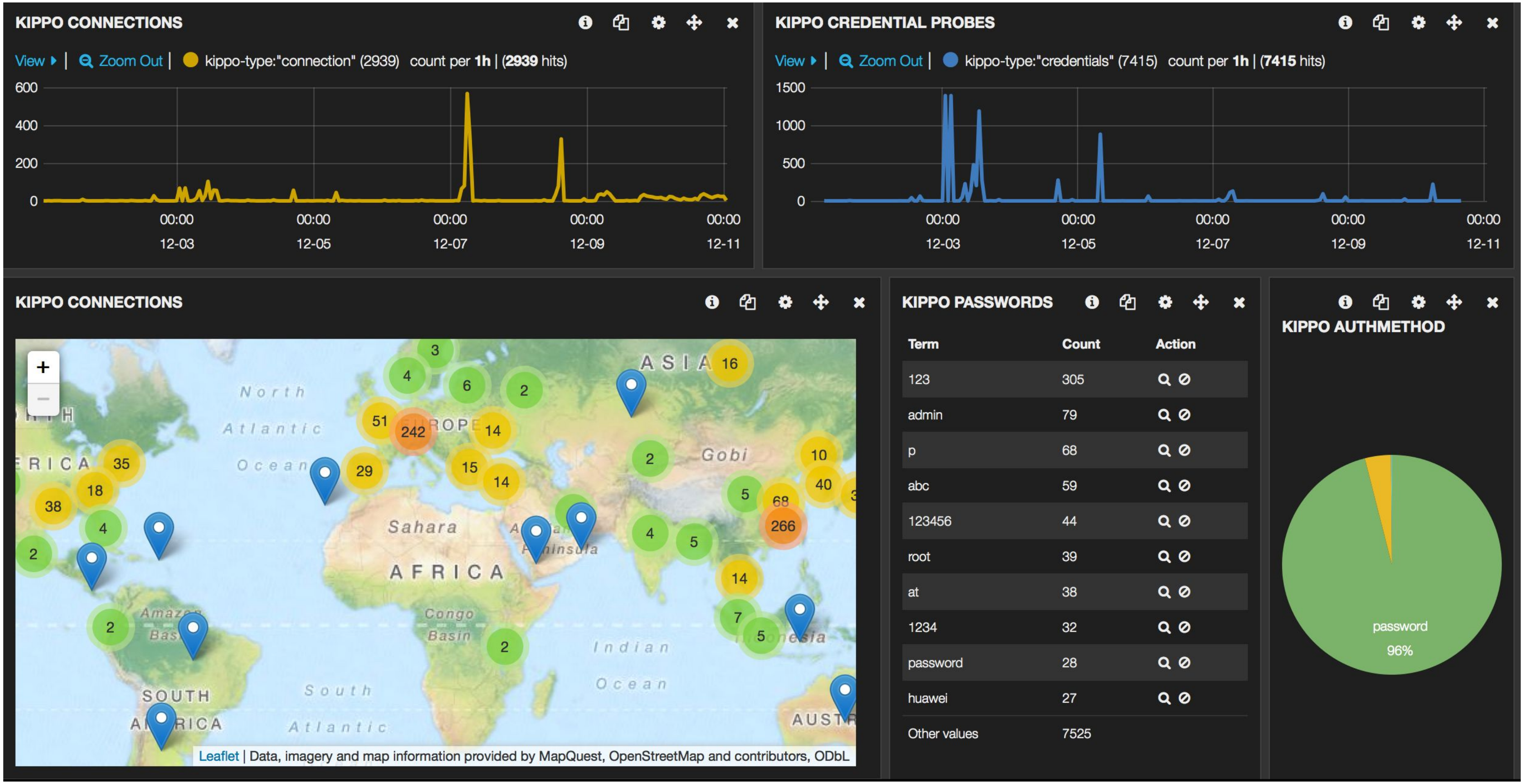
Cowrie Features

- Fake filesystem resembling a Debian 5.0 installation with the ability to add/remove files.
- Possibility of adding fake file contents so the attacker can cat files such as /etc/passwd.
- Cowrie saves files downloaded with wget/curl or uploaded with SFTP and scp for later inspection

Cowrie Features

- SFTP and SCP support for file upload
- Support for SSH exec commands
- Logging of direct-tcp connection attempts (ssh proxying)
- Forward SMTP connections to SMTP Honeypot
- Logging in JSON format for easy processing in log management solutions
- Many, many additional commands

Kibana Dashboards



KIPPO CONNECTIONS

View ▸ | 🔍 Zoom Out | ● kippo-type:"connection" (2939) count per **1h** | (**2939** hits)

KIPPO CREDENTIAL PROBES

View ▸ | 🔍 Zoom Out | ● kippo-type:"credentials" (7415) count per **1h** | (**7415** hits)

KIPPO CONNECTIONS

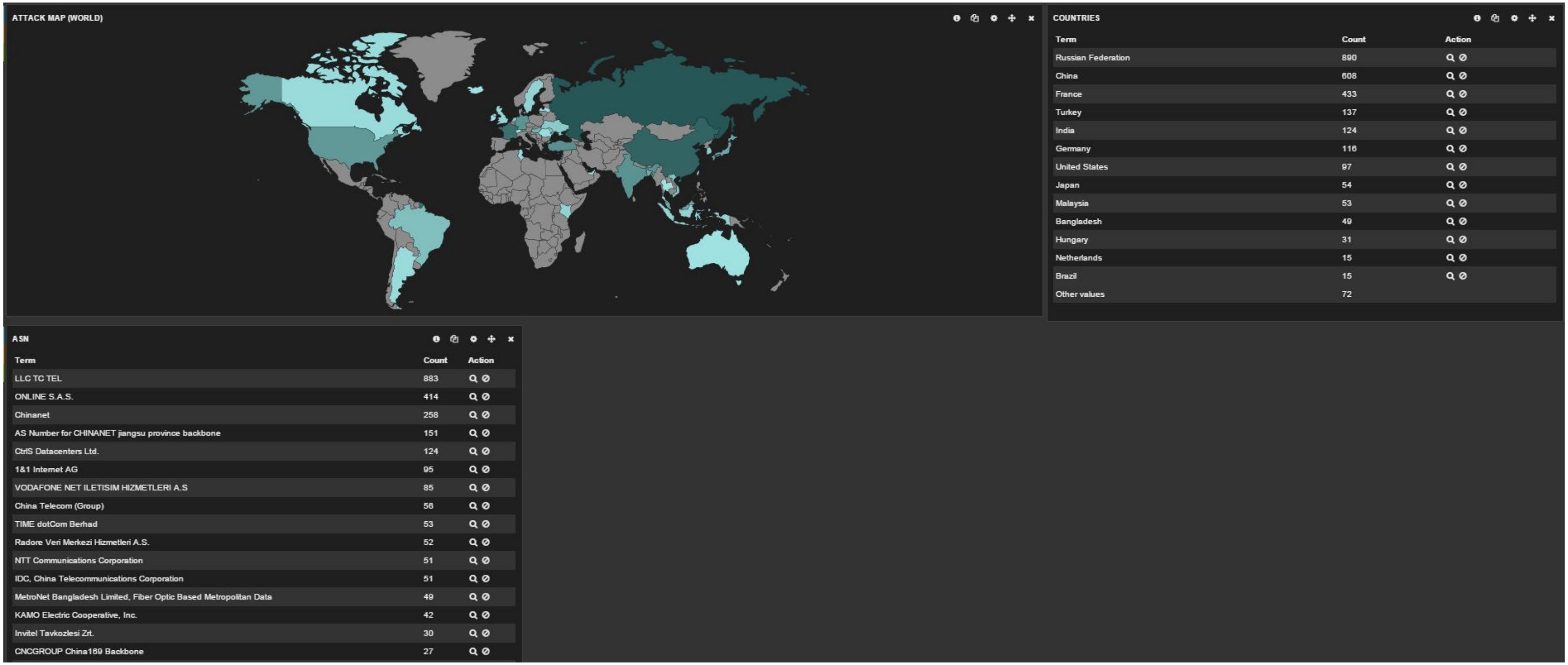
Leaflet | Data, imagery and map information provided by MapQuest, OpenStreetMap and contributors, ODbL

KIPPO PASSWORDS

Term	Count	Action
123	305	🔍 ⊘
admin	79	🔍 ⊘
p	68	🔍 ⊘
abc	59	🔍 ⊘
123456	44	🔍 ⊘
root	39	🔍 ⊘
at	38	🔍 ⊘
1234	32	🔍 ⊘
password	28	🔍 ⊘
huawei	27	🔍 ⊘
Other values	7525	

KIPPO AUTHMETHOD

Kibana Dashboards



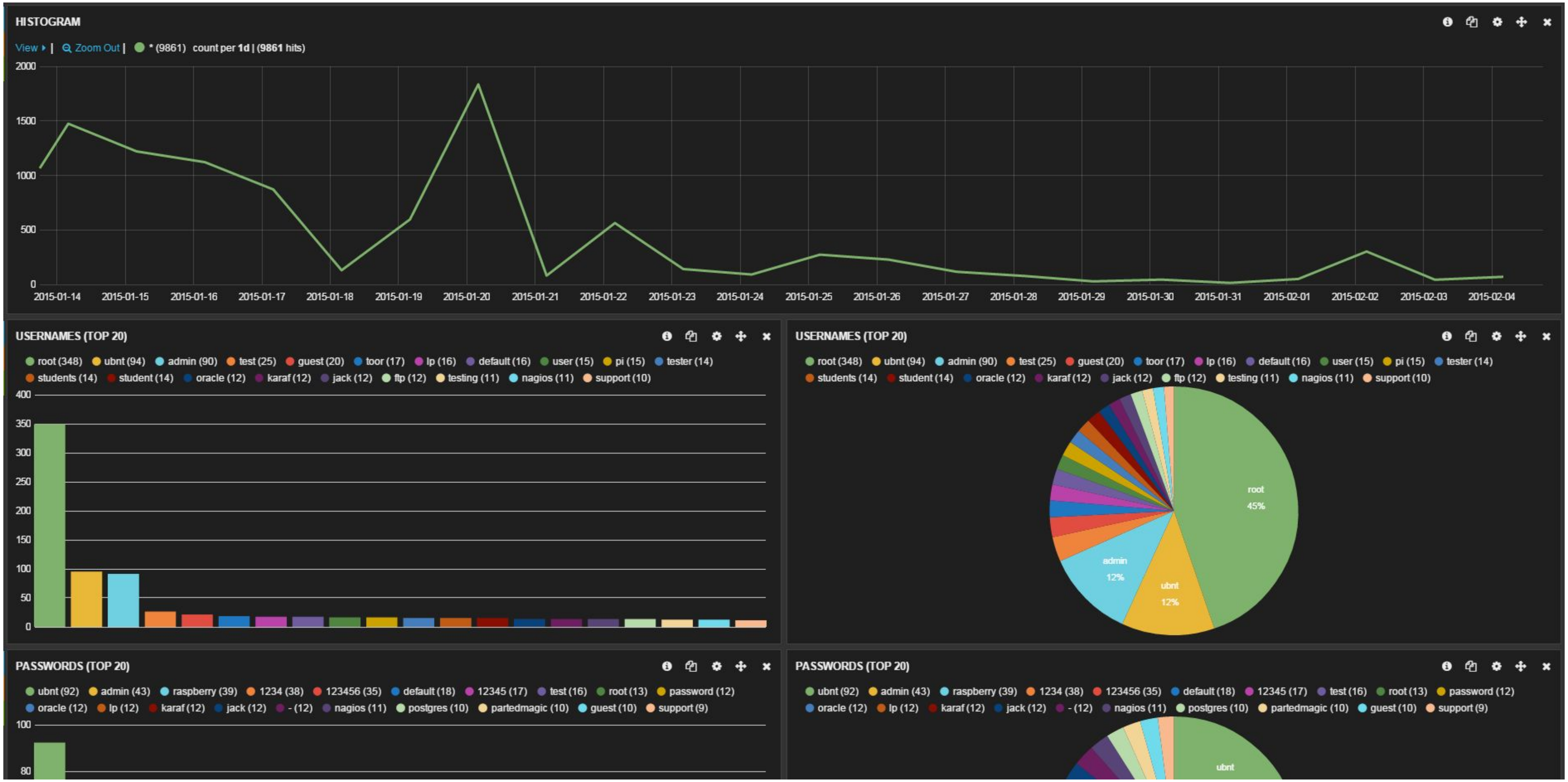
COUNTRIES

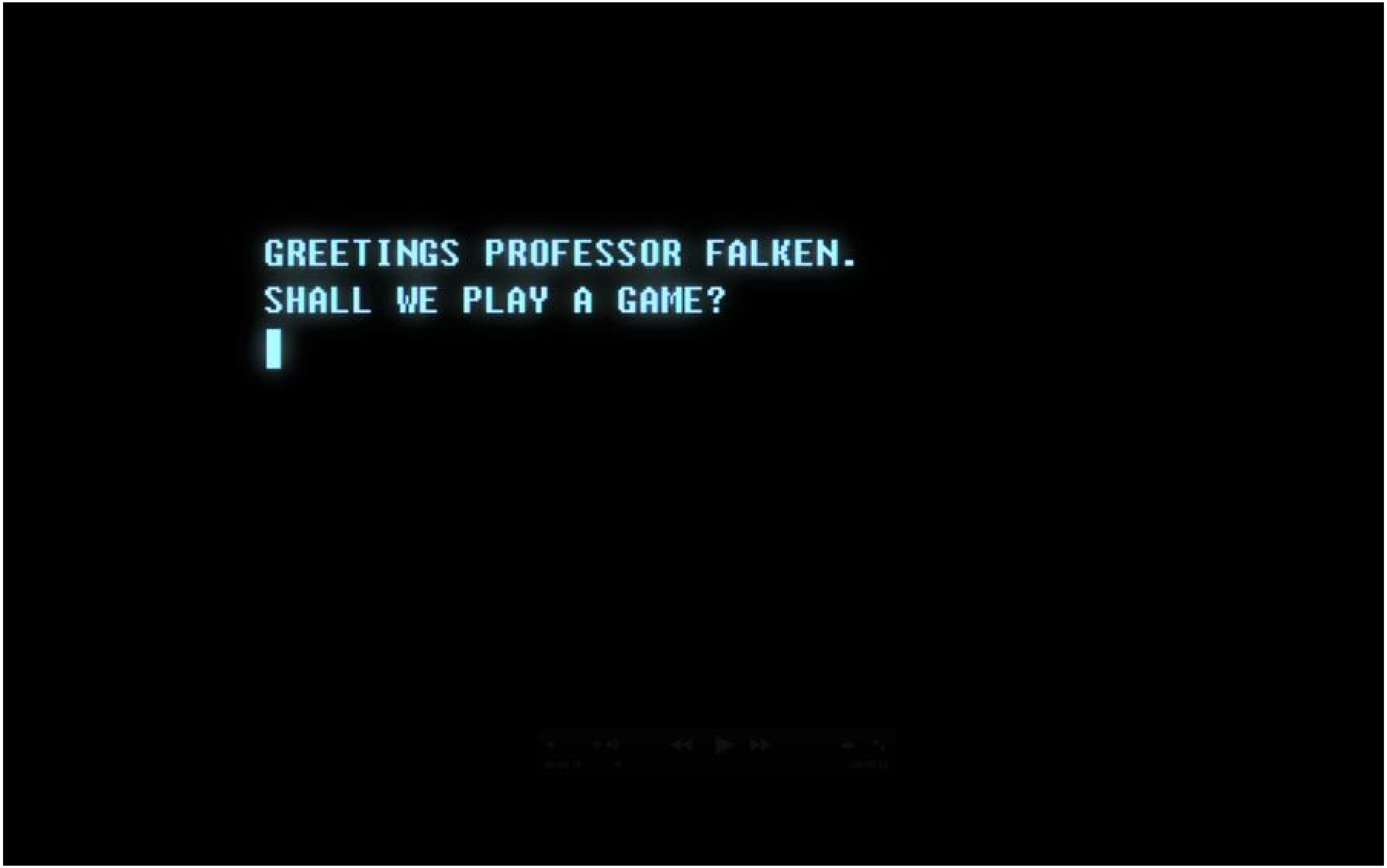
Term	Count	Action
Russian Federation	890	🔍 ⊘
China	608	🔍 ⊘
France	433	🔍 ⊘
Turkey	137	🔍 ⊘
India	124	🔍 ⊘
Germany	116	🔍 ⊘
United States	97	🔍 ⊘
Japan	54	🔍 ⊘
Malaysia	53	🔍 ⊘
Bangladesh	49	🔍 ⊘
Hungary	31	🔍 ⊘
Netherlands	15	🔍 ⊘
Brazil	15	🔍 ⊘
Other values	72	

ASN

Term	Count	Action
LLC TC TEL	883	🔍 ⊘
ONLINE S.A.S.	414	🔍 ⊘
Chinanet	258	🔍 ⊘
AS Number for CHINANET jiangsu province backbone	151	🔍 ⊘
CtrlS Datacenters Ltd.	124	🔍 ⊘
1&1 Internet AG	95	🔍 ⊘
VODAFONE NET ILETISIM HIZMETLERI A.S	85	🔍 ⊘
China Telecom (Group)	56	🔍 ⊘
TIME dotCom Berhad	53	🔍 ⊘
Radore Veri Merkezi Hizmetleri A.S.	52	🔍 ⊘
NTT Communications Corporation	51	🔍 ⊘
IDC, China Telecommunications Corporation	51	🔍 ⊘
MetroNet Bangladesh Limited, Fiber Optic Based Metropolitan Data	49	🔍 ⊘
KAMO Electric Cooperative, Inc.	42	🔍 ⊘
Invitel Tavkozlesi Zrt.	30	🔍 ⊘
CNCGROUP China169 Backbone	27	🔍 ⊘

Kibana Dashboards


HISTOGRAM
View ▸ | Zoom Out | ● * (9861) count per 1d | (9861 hits)
2000
1500
1000
500
0
2015-01-14
2015-01-15
2015-01-16
2015-01-17
2015-01-18
2015-01-19
2015-01-20
2015-01-21
2015-01-22
2015-01-23
2015-01-24
2015-01-25
2015-01-26
2015-01-27
2015-01-28
2015-01-29
2015-01-30
2015-01-31
2015-02-01
2015-02-02
2015-02-03
2015-02-04
USERNAMES (TOP 20)
● root (348) ● ubnt (94) ● admin (90) ● test (25) ● guest (20) ● toor (17) ● lp (16) ● default (16) ● user (15) ● pi (15) ● tester (14)
● students (14) ● student (14) ● oracle (12) ● karaf (12) ● jack (12) ● ftp (12) ● testing (11) ● nagios (11) ● support (10)
400
350
300
250
200
150
100
50
0
USERNAMES (TOP 20)
● root (348) ● ubnt (94) ● admin (90) ● test (25) ● guest (20) ● toor (17) ● lp (16) ● default (16) ● user (15) ● pi (15) ● tester (14)
● students (14) ● student (14) ● oracle (12) ● karaf (12) ● jack (12) ● ftp (12) ● testing (11) ● nagios (11) ● support (10)
root
45%
admin
12%
ubnt
12%
PASSWORDS (TOP 20)
● ubnt (92) ● admin (43) ● raspberry (39) ● 1234 (38) ● 123456 (35) ● default (18) ● 12345 (17) ● test (16) ● root (13) ● password (12)
● oracle (12) ● lp (12) ● karaf (12) ● jack (12) ● - (12) ● nagios (11) ● postgres (10) ● partedmagic (10) ● guest (10) ● support (9)
100
80
PASSWORDS (TOP 20)
● ubnt (92) ● admin (43) ● raspberry (39) ● 1234 (38) ● 123456 (35) ● default (18) ● 12345 (17) ● test (16) ● root (13) ● password (12)
● oracle (12) ● lp (12) ● karaf (12) ● jack (12) ● - (12) ● nagios (11) ● postgres (10) ● partedmagic (10) ● guest (10) ● support (9)
ubnt



```
GREETINGS PROFESSOR FALKEN.
SHALL WE PLAY A GAME?
```

Glastopf

Glastopf

Glastopf is a low-interaction server honeypot for web applications. It is able to emulate vulnerabilities and gather information about incoming attacks. Its working principle is to respond to the attacker in accordance with his expectations, in order to provoke an attack.

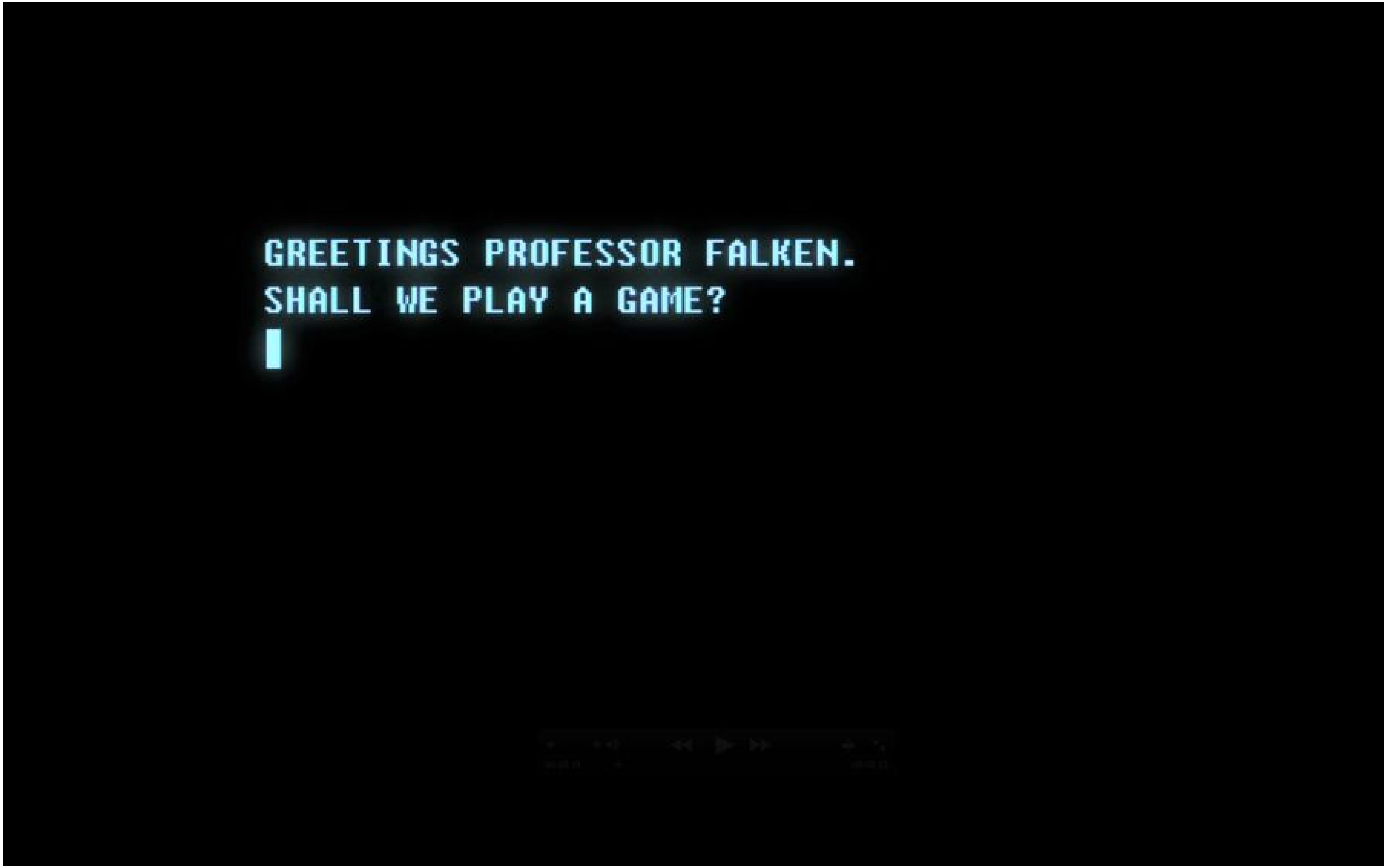
Glastopf was founded by Lukas Rist.
https://github.com/mushorg/glastopf

Glastopf

Glastopf supports multistage attacks. It has a built-in PHP sandbox for code injection emulation. It can be run standalone in its own Python web server or via WSGI. It has modular architecture, which allows it to attract attacks targeting any web application.

DETECTION SCOPE	ACCURACY OF EMULATION	QUALITY OF COLLECTED DATA	SCALABILITY AND PERFORMANCE	RELIABILITY	EXTENSIBILITY	EASE OF USE AND SETTING UP	EMBEDDABILITY	SUPPORT	COST	USEFULNESS FOR CERT
SPEC	★★★★	★★★	★★	★★★	★★★★	★★★	★★★	★★★★	$	😁

Detection scope		Rating		Cost		Usefulness for CERT	
MULTI	Multi-function	★★★★	Excellent	$	Low	😁	Essential
		★★★	Good	$$	Medium	🙂	Useful
SPEC	Specialised	★★	Fair	$$$	High	☹	Not useful
		★	Poor				

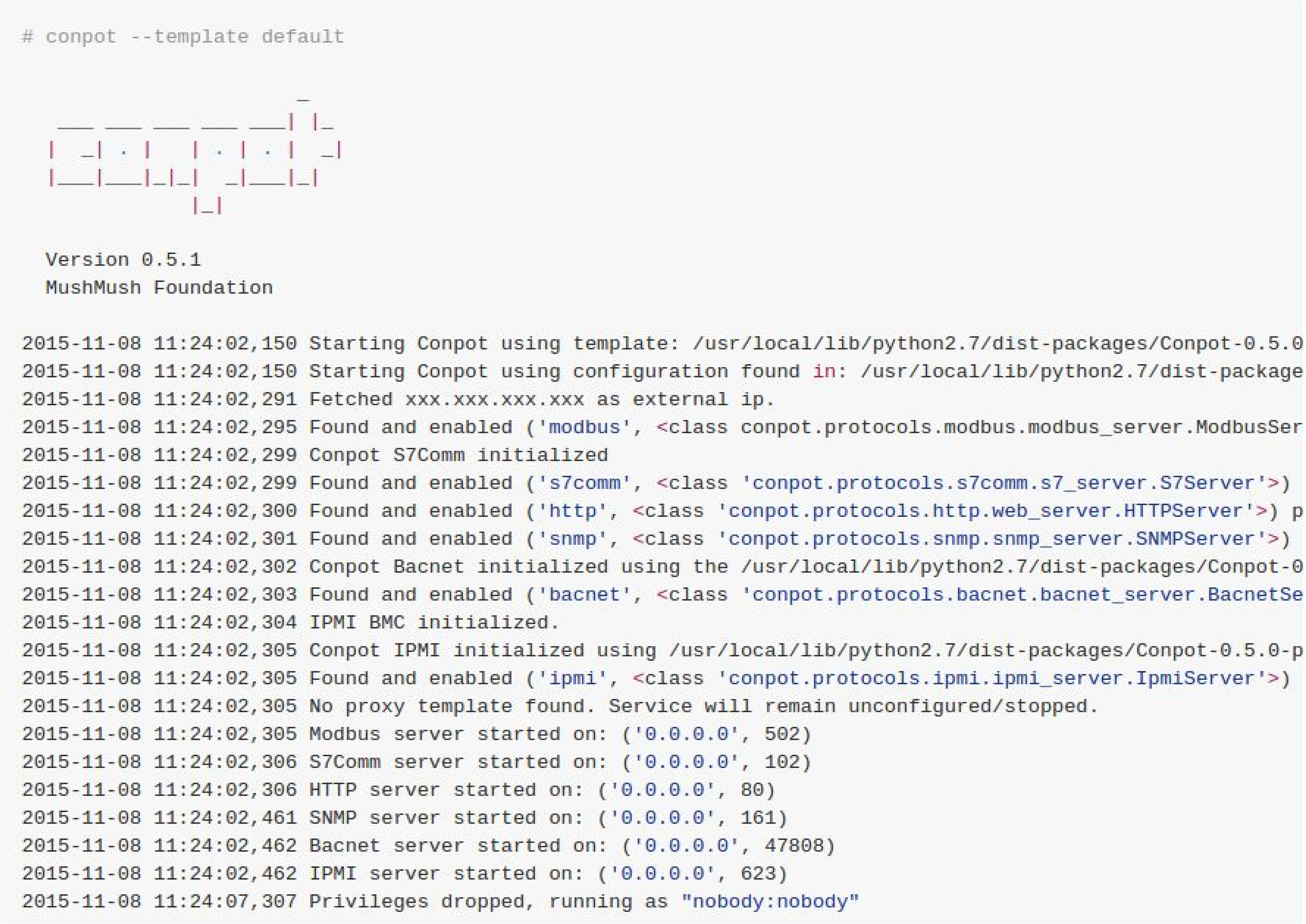


```
GREETINGS PROFESSOR FALKEN.
SHALL WE PLAY A GAME?
```

Want to run a Nuclear Plant at Home?

Conpot

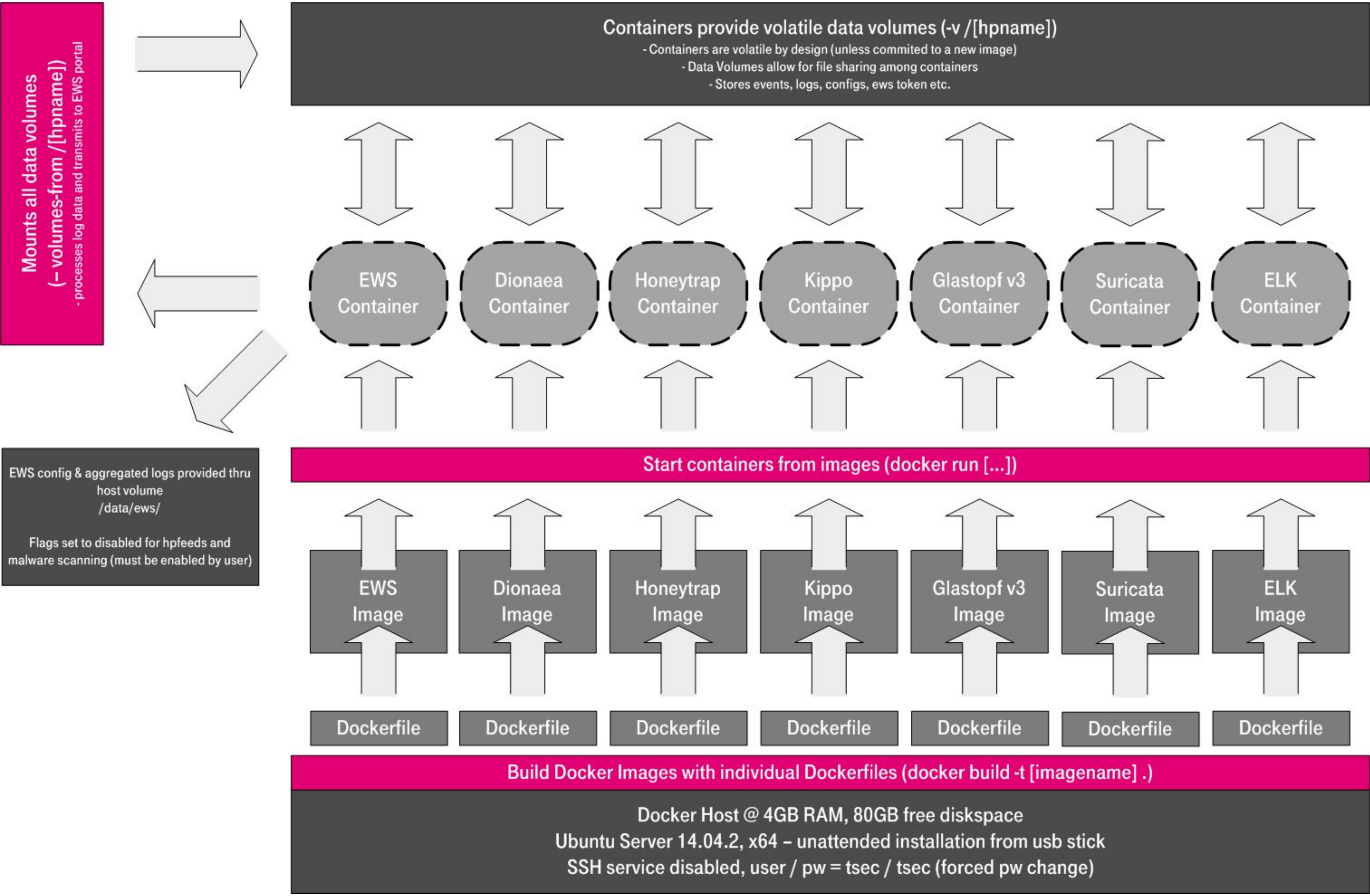
An Industrial Control System Honeypot



```
# conpot --template default


                       _
   ___ ___ ___ ___ ___| |_
  |  _| . |   | . | . |  _|
  |___|___|_|_|  _|___|_|
              |_|

  Version 0.5.1
  MushMush Foundation

2015-11-08 11:24:02,150 Starting Conpot using template: /usr/local/lib/python2.7/dist-packages/Conpot-0.5.0
2015-11-08 11:24:02,150 Starting Conpot using configuration found in: /usr/local/lib/python2.7/dist-package
2015-11-08 11:24:02,291 Fetched xxx.xxx.xxx.xxx as external ip.
2015-11-08 11:24:02,295 Found and enabled ('modbus', <class conpot.protocols.modbus.modbus_server.ModbusSer
2015-11-08 11:24:02,299 Conpot S7Comm initialized
2015-11-08 11:24:02,299 Found and enabled ('s7comm', <class 'conpot.protocols.s7comm.s7_server.S7Server'>)
2015-11-08 11:24:02,300 Found and enabled ('http', <class 'conpot.protocols.http.web_server.HTTPServer'>) p
2015-11-08 11:24:02,301 Found and enabled ('snmp', <class 'conpot.protocols.snmp.snmp_server.SNMPServer'>)
2015-11-08 11:24:02,302 Conpot Bacnet initialized using the /usr/local/lib/python2.7/dist-packages/Conpot-0
2015-11-08 11:24:02,303 Found and enabled ('bacnet', <class 'conpot.protocols.bacnet.bacnet_server.BacnetSe
2015-11-08 11:24:02,304 IPMI BMC initialized.
2015-11-08 11:24:02,305 Conpot IPMI initialized using /usr/local/lib/python2.7/dist-packages/Conpot-0.5.0-p
2015-11-08 11:24:02,305 Found and enabled ('ipmi', <class 'conpot.protocols.ipmi.ipmi_server.IpmiServer'>)
2015-11-08 11:24:02,305 No proxy template found. Service will remain unconfigured/stopped.
2015-11-08 11:24:02,305 Modbus server started on: ('0.0.0.0', 502)
2015-11-08 11:24:02,306 S7Comm server started on: ('0.0.0.0', 102)
2015-11-08 11:24:02,306 HTTP server started on: ('0.0.0.0', 80)
2015-11-08 11:24:02,461 SNMP server started on: ('0.0.0.0', 161)
2015-11-08 11:24:02,462 Bacnet server started on: ('0.0.0.0', 47808)
2015-11-08 11:24:02,462 IPMI server started on: ('0.0.0.0', 623)
2015-11-08 11:24:07,307 Privileges dropped, running as "nobody:nobody"
```

Still don’t know which one to run?

So run ‘em all!



Mounts all data volumes
(--volumes-from /[hpname])
- processes log data and transmits to EWS portal

Containers provide volatile data volumes (-v /[hpname])
- Containers are volatile by design (unless commited to a new image)
- Data Volumes allow for file sharing among containers
- Stores events, logs, configs, ews token etc.

EWS Container

Dionaea Container

Honeytrap Container

Kippo Container

Glastopf v3 Container

Suricata Container

ELK Container

EWS config & aggregated logs provided thru host volume
/data/ews/

Flags set to disabled for hpfeeds and malware scanning (must be enabled by user)

Start containers from images (docker run [...])

EWS Image

Dionaea Image

Honeytrap Image

Kippo Image

Glastopf v3 Image

Suricata Image

ELK Image

Dockerfile

Dockerfile

Dockerfile

Dockerfile

Dockerfile

Dockerfile

Dockerfile

Build Docker Images with individual Dockerfiles (docker build -t [imagename] .)

Docker Host @ 4GB RAM, 80GB free diskspace
Ubuntu Server 14.04.2, x64 – unattended installation from usb stick
SSH service disabled, user / pw = tsec / tsec (forced pw change)

Thank You