





Welcome to

## 10. Honeypots

### Communication and Network Security 2022

Henrik Kramselund he/him han/ham hkj@zencurity.com @kramse  

Slides are available as PDF, kramse@Github  
10-Honeypots.tex in the repo security-courses

# Goals for today



Today's goals:

- Talk about investigating network data, using Zeek
- We already used logstash and Kibana
- Spend time in Zeek data using the Filebeat import

Trying to make today less heavy with information, same tomorrow with the Email subject.

# Plan for today



## Subjects

- History of honeypots
- Why use them research, production
- Types of honeypots low vs high interaction
- Honey nets
- Discuss visualisation and investigating incidents

## Exercises

- Run SSH honeypot and try brute-force it
- Use Filebeat and investigate the options available from Zeek data

# Reading Summary



ANSM chapter 11,12 - 54 pages

11. Anomaly-Based Detection with Statistical Data

12. Using Canary Honeypots for Detection

# 11. Anomaly-Based Detection with Statistical Data



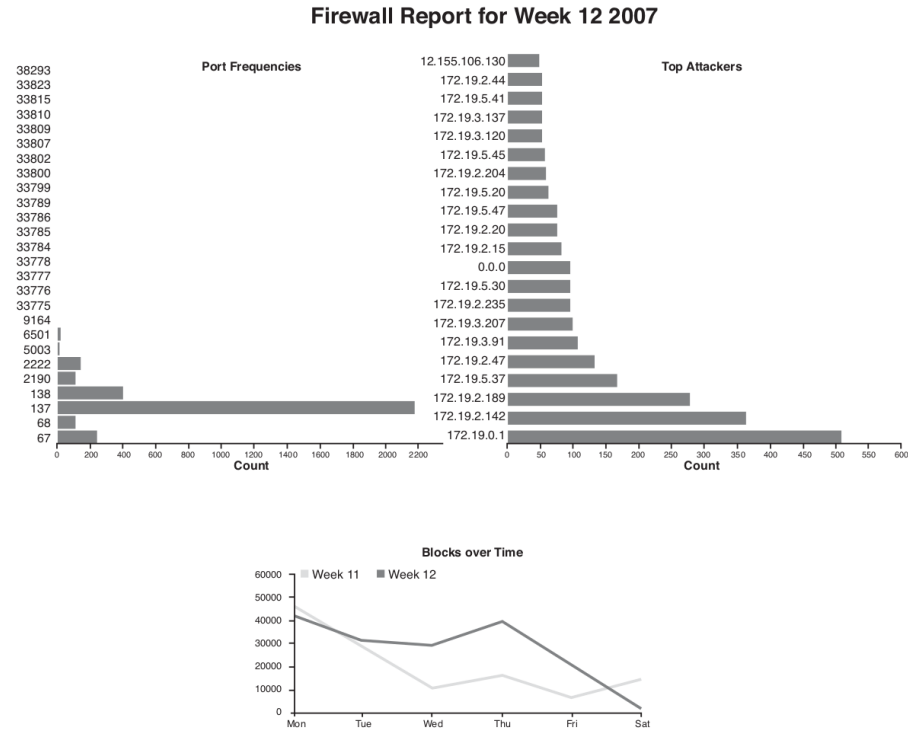
Good advice found in the book:

- Top Talkers with SiLK
- Service Discovery with SiLK
- Furthering Detection with Statistics
- Visualizing Statistics with Gnuplot
- Visualizing Statistics with Google Charts
- Visualizing Statistics with Afterglow

Newer and other tools exist, but the process is the same.

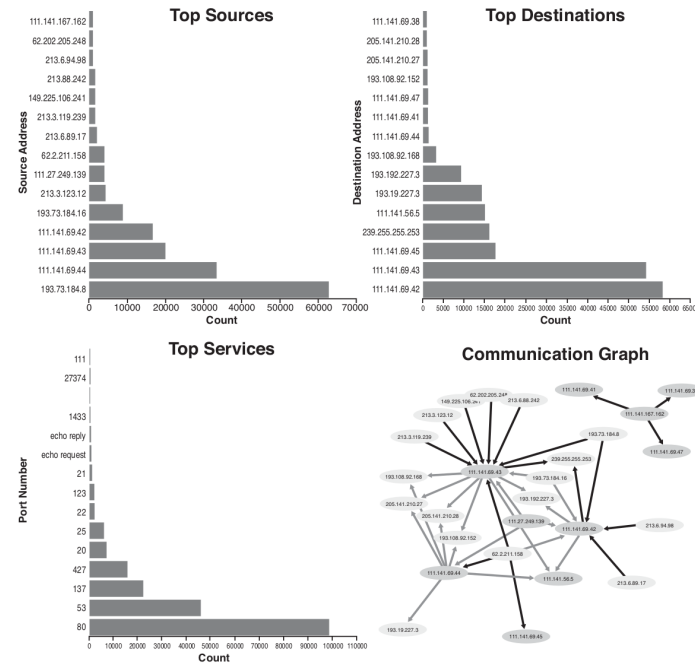
Source: Applied Network Security Monitoring Collection, Detection, and Analysis, 2014 Chris Sanders ISBN: 9780124172081

# Applied Security Visualization examples



Source: Firewall Report in *Applied security visualization*, Rafael Marty, 2009

# Applied Security Visualization examples



Source: Network Flow Data in *Applied security visualization*, Rafael Marty, 2009

# Honeypot Definition



In computer terminology, a **honeypot** is a computer security mechanism set to detect, deflect, or, in some manner, counteract attempts at unauthorized use of information systems. Generally, a honeypot consists of data (for example, in a network site) that appears to be a legitimate part of the site, but is actually isolated and monitored, and that seems to contain information or a resource of value to attackers, who are then blocked.

Source: [https://en.wikipedia.org/wiki/Honeypot\\_\(computing\)](https://en.wikipedia.org/wiki/Honeypot_(computing))

also used as Honeynet - monitored network infrastructure

En honeypot består typisk af:

- Et eller flere sårbare systemer
- Et eller flere systemer der logger trafik til og fra honeypot systemerne

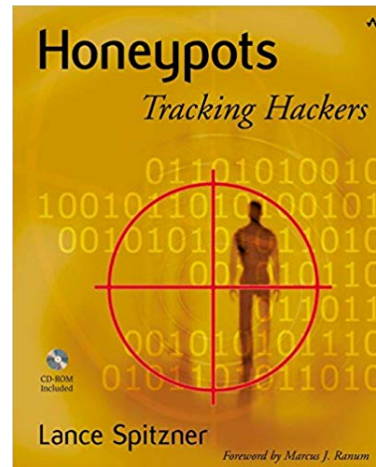
Meningen med en honeypot er at den bliver angrebet og brudt ind i, se også Canary Tokens



# History of honeypots



## An Evening with Berferd



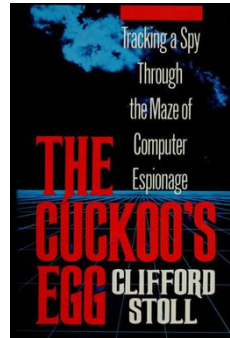
Artikel om en hacker der lokkes, vurderes, overvåges

Et tidligt eksempel på en honeypot

Senere kom The HoneyNet Project <http://www.honeynet.org>

Billede er: *Honeypots: Tracking Hackers* af Lance Spitzner, 2003

## Cuckoo's Egg 1986 A real spy story



*Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*

Stoll brugte også lignende lavede interessante filer som hackeren hentede - over modem

*During his time at working for KGB, Hess is estimated to have broken into 400 U.S. military computers*

Source: [https://en.wikipedia.org/wiki/Markus\\_Hess](https://en.wikipedia.org/wiki/Markus_Hess)

# ANSM 12. Using Canary Honey pots for Detection



## Canary Honey pots

### Types of Honey pots

### Canary Honey pot Architecture

- Phase One: Identify Devices and Services to be Mimicked
- Phase Two: Determine Canary Honey pot Placement
- Phase Three: Develop Alerting and Logging

### Honey pot Platforms

- Honeyd
- Kippo SSH Honey pot
- Tom's Honey pot
- Honeydocs

Source: Applied Network Security Monitoring Collection, Detection, and Analysis, 2014 Chris Sanders ISBN: 9780124172081

# Honeypots - ressourcekrævende?



"There are 69 separate departments at Georgia Tech with between 30,000-35,000 networked computers installed on campus."...  
"In the six months that we have been running the Georgia Tech Honeynet **we have detected 16 compromised Georgia Tech systems on networks** other than our Honeynet. These compromises include automated worm type exploits as well as individual systems that have been targeted and compromised by hackers."

## *The Use of Honeynets to Detect Exploited Systems Across Large Enterprise Networks*

Honeypots og IDS systemer kan være ressourcekrævende, men en kombination kan være mere effektiv i visse tilfælde

Kilde: <https://staff.washington.edu/dittrich/pnw-honeynet/reading/gatech-honeynet.pdf>

# Honeypot High interaction and low interaction

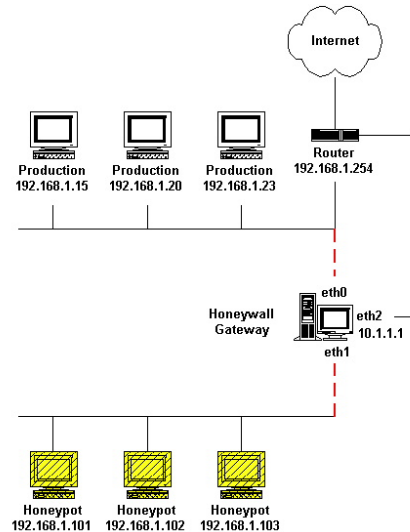


**High-interaction** honeypots imitate the activities of the production systems that host a variety of services and, therefore, an attacker may be allowed a lot of services to waste their time. By employing virtual machines, multiple honeypots can be hosted on a single physical machine. Therefore, even if the honeypot is compromised, it can be restored more quickly. In general, high-interaction honeypots provide more security by being difficult to detect, but they are expensive to maintain. If virtual machines are not available, one physical computer must be maintained for each honeypot, which can be exorbitantly expensive. Example: Honeynet.

**Low-interaction** honeypots simulate only the services frequently requested by attackers. Since they consume relatively few resources, multiple virtual machines can easily be hosted on one physical system, the virtual systems have a short response time, and less code is required, reducing the complexity of the virtual system's security. Example: Honeyd.

Source: [https://en.wikipedia.org/wiki/Honeypot\\_\(computing\)](https://en.wikipedia.org/wiki/Honeypot_(computing))

# Honeynets - Why use them research, production



Creating a network architecture with multiple systems become a honeynet.

- Lessons Learned from <http://old.honeynet.org/papers/edu/>
- Out of all of this were a variety of lessons learned things to do and NOT to do. Hopefully this short list can help you avoid some common mistakes.



- Start Small - If you are going to install a honeynet within your enterprise, start small. Begin initially with two machines (in order to detect sweep scans of your honeynet) with operating systems that you are familiar with installed behind the reverse firewall.
- Maintain good relations with your enterprise administrators. THIS IS CRITICAL! Inform your network administrators of the types of exploits that you are seeing. In some cases, they will already be aware of these exploits, but in other cases, you will have been the first person to notice them.
- Focus on attacks and exploits originating from within your enterprise network. These are the attacks that can do the most damage to your enterprise. Inform your enterprise administrators immediately of these types of attacks since they indicate machines that have already been compromised within the enterprise.
- Don't publish the IP address range of the honeynet. There is no need to do this. Hackers and worms are constantly scanning across the Internet for machines to exploit. Your honeynet will be found and attacked.
- Don't underestimate the amount of time required to analyze the data collected from the honeynet. This data must be analyzed every day. You will be collecting lots of information and it must be analyzed to provide any benefit.
- Powerful machines are not necessary to establish the honeynet. The Georgia Tech Honeynet did not use state of the art machines and it functioned as intended. Everything we needed to establish our honeynet was already available on campus.

Source: *Know Your Enemy: Honeynets in Universities Deploying a Honeynet at an Academic Institution*



# Honeypot vs NIDS



## NIDS

- + See all traffic
- — see and need to process ALL TRAFFIC
- + Known and understood by management

## Honeypot

- + See only attack traffic
- + Few false positives
- + Require less ressources

## Selecting honeypot



We will work with a SSH honeypot, since our servers used in the labs are Debian

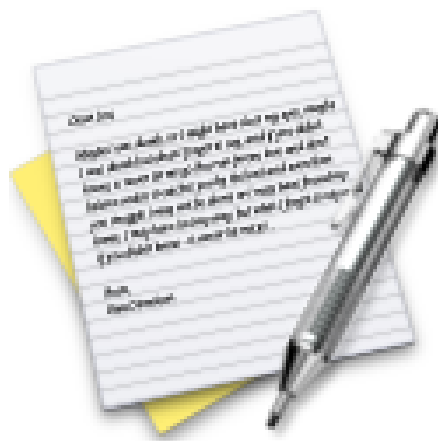
Searching for ssh honeypot show an example: Kippo, <https://github.com/desaster/kippo>, and this has a more recent fork: <https://github.com/cowrie/cowrie>

Very common - an open source tool exist, and reusing existing projects save time!

Maybe even try to get graphs from it using AfterGlow!

<https://xn--blgg-hra.no/2017/01/how-to-produce-afterglow-diagrams-from-cowrie/>

# Exercise



Now lets do the exercise

## Fun with SSH honeypots 30min

which is number **66** in the exercise PDF.

# Security visualisation



We have talked about Kibana, but there are lots of other tools:

- graphviz, tulip, cytoscape, and gephi
- afterglow <http://afterglow.sourceforge.net/>  
<https://xn--blgg-hra.no/2017/01/how-to-produce-afterglow-diagrams-from-cowrie/>
- treemap
- mondrian, ggobi

More inspiration can be found on sites like: <https://secviz.org/>

A picture or graph often show more than just a table of data

# SecViz - Security Visualization



- Lets visit <https://secviz.org/>
- Look into the Gallery

# Exercise

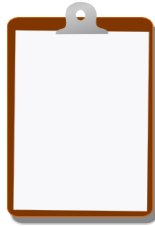


Now lets do the exercise

## Integrating Zeek IDS with the Elastic Stack 30min

which is number **67** in the exercise PDF.

## For Next Time



Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools