Honeypots & Honeynets Overview

Adli Wahid

Security Specialist, APNIC.net

adli@apnic.net

Contents

1. Objectives
2. Definition of Honeypot & Honeynets
3. Benefits & Risk consideration
4. Example of Honeypot tools
5. The Honeynet Project

Credits: David Watson (Honeynet Project) for the some of the contents of this slide
david@honeynet.org.uk

Objectives

1. Understand the the concept of honeypots / honeynets and how they are deployed
2. Understand the value of honeypots and honeynets to security teams
3. Familiarize with different types of honeypots

Know Your Enemy

How can we defend against an enemy, when we don’t even know who the enemy is?

(Lance Spitzner 1999)

Know Your Enemy (2)

To learn the tools, tactics and motives involved in computer and network attacks, and share the lessons learned

(Mission Statement, The Honeynet Project)

These days you may be familiar with the term 'Threat Intelligence'

Honeypots and Honeynets

- A honeypot is an information system resource whose value lies in the unauthorized or illicit use of that resource
- Honeypot systems have no production value, so any activity going to or from a honeypot is likely a probe, attack or compromise
- A honeynet is simply a network of honeypots
- Information gathering and early warning are the primary benefits to most organisations

Honeypot and Honeynet Types

- Low-Interaction (LI)
  - – Emulates services, applications and OS's
  - – Easier to deploy/maintain, low risk, but only limited information
- High-Interaction (HI)
  - – Real services, applicatios and OS's
  - – Capture extensive information, but higher risk and time intensive to maintain

Honeypot and Honeynet Types

- Server Honeypots
  - Listen for incoming network connections
  - Analyse attacks targeting host's users, services and operating systems

- Client Honeypots
  - Reach out and interact with remote potentially malicious resources
  - Have to be instructed where to go to find evil
  - Analyse attacks targeting clients and users

Honeypot and Honeynet Pros / Cons

Pros

- Simple Concept
- Collect small data sets of high value
- Few False Positives
- Catch new attacks
- Low False Negatives
- Can beat encryption
- Minimal hardware
- Real time alerting

Cons

- Potentially complex
- Need data analysis
- Only a microscope
- Detection by attackers
- Risk from compromises
- Legal concerns
- False negatives
- Potentially live 24/7
- Operationally intensive

Implementing Honeypot

Recap

Evilness

Badness

Noise

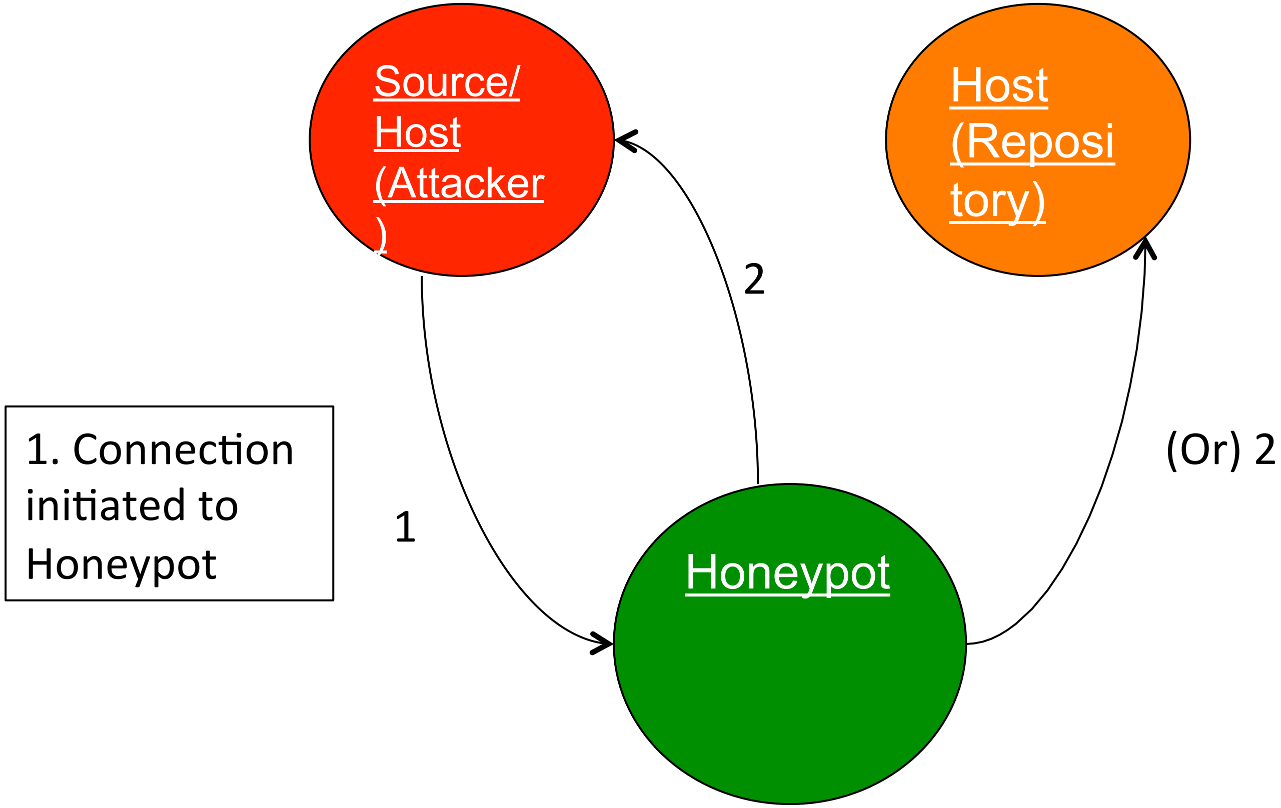
Malware

Honeypots: Computer resource(s) to be probed and/or attacked

Why would you want to do this?

- By right, you should not expect any real activity or traffic to/from/in your honeypot
- Detect anomalous activities in your network or system?
  - Infected / Compromised computers
  - Misconfiguration
- Learn about attacks in the wild (research)
  - Attackers and attacker techniques
  - Information Sharing opportunities
  - Improve overall Security

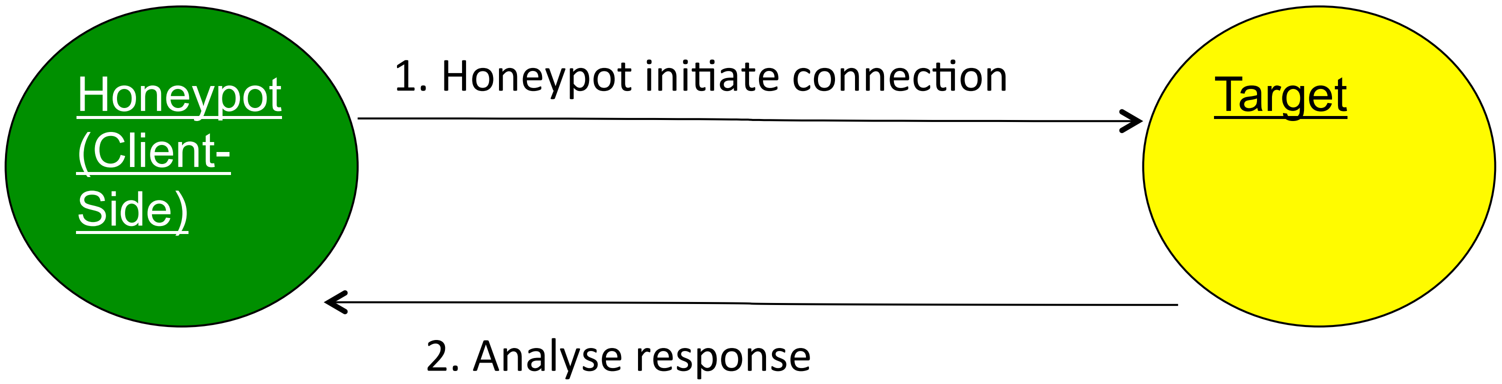
Example 'Network-based Attack' Pattern



What can you learn?

- Hosts that are trying to connect / scan you
  - Potentially already compromised or infected
- The payload used after successfully gaining access to the honeypot system
- Scripts, binaries/executables etc

Example of Client-based Honeypot



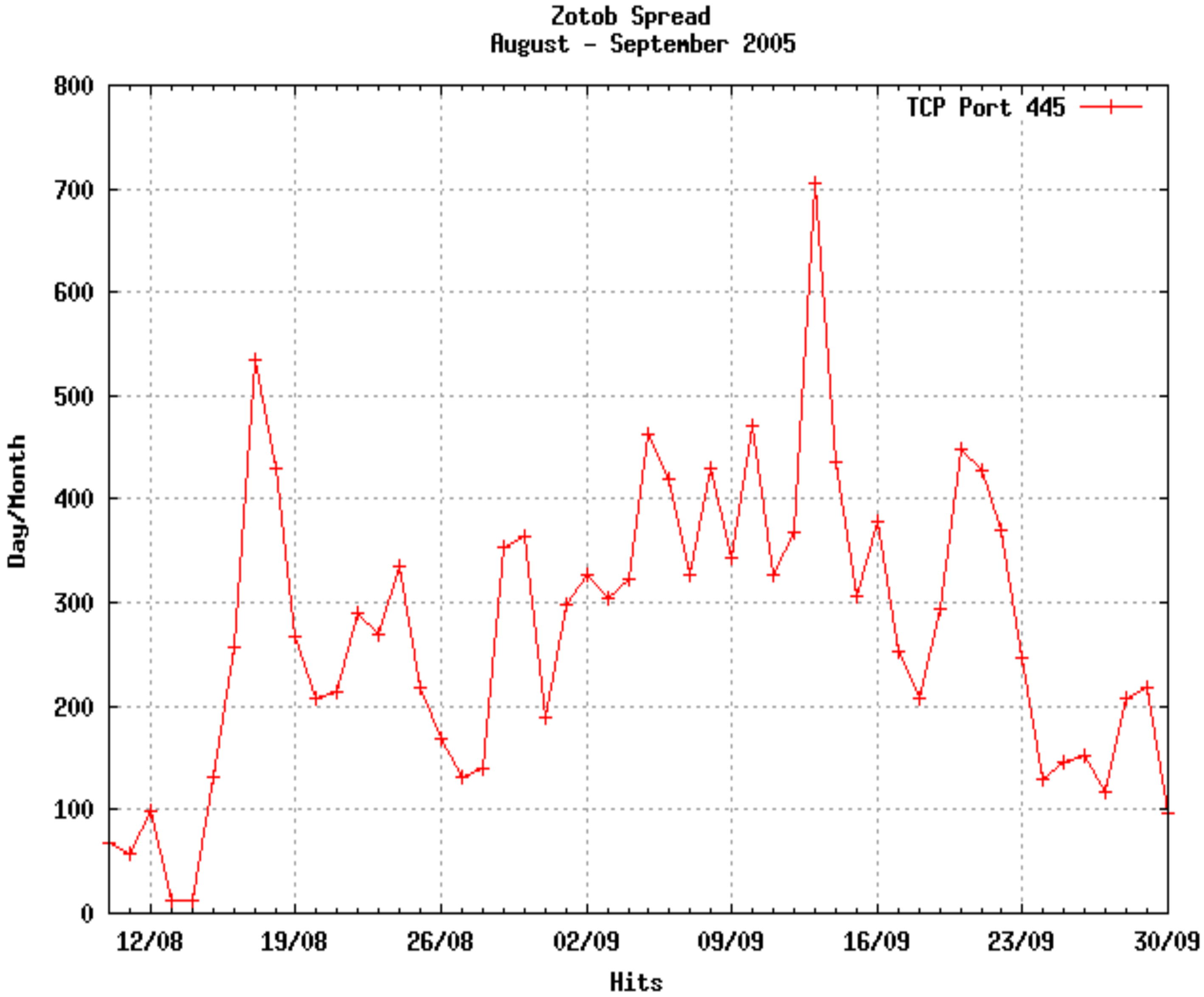
What you can learn?

- (0-days) or attacks on the Client Application (i.e. Web Browser)
- Learn about hosts / computers that are hosting malicious websites
  - <Iframes>
  - Javascript
  - Flash
  - PDF etc

Exercise 1

Let us discuss

- What is the difference between IDS and Honeypot?
- What are some of the Use Cases for deploying a honeypot or honeynet?
- What is the mininum that is required for setting up a honeypot?
- Look at the previous 2 scenarios, what are the intelligence that we can obtain from implementing 2 types of honeypots?


Zotob Spread
August - September 2005
TCP Port 445
Day/Month
Hits
800
700
600
500
400
300
200
100
0
12/08
19/08
26/08
02/09
09/09
16/09
23/09
30/09

Deploying Honeypots with Publically Available Tools

High Interaction Honeypot

- Think about your goals and objectives first
- Possible scenario
  - Setup a real system and make give it an IP address (so it is reachable to something)
  - i.e. Install a Windows, Linux, Unix server)
- Challenging to control & manage
  - What if attacker use system to launch attack to other systems
  - Keeping the computer in a usable state

Some Examples

- Dionaea (Malware)
- Kippo - SSH honeypot
- Glastopf – Web Honeypot
- Ghost – USB Honeypot
- Thug – Client Honeypot

Dionaea

- 2nd Generation low interaction honeypot
  - Python, runs on *NIX
  - IPv6 Support
- Goals
  - Detect both known and unknown attacks
  - Better protocol awareness
  - Vulnerability modules in scripting language
  - Shell code detection using LibEmu
- Check out http://dionaea.carnivore.it
- Learn about attacks, malware and many more

Kippo

- Emulate SSH server
  - Allow 'attacker' to log-in using credentials (username and password)
  - Environment allow limited commands – i.e. ping, who, and wget
  - Record activities (keylog) of attackers and their activities

Glastopf Web Honeypot

- Minimalistic web server written in Python
- Scans incoming HTTP requests strings
- Checks for remote file inclusion (RFI), local file inclusion (LFI) and SQL injection
- Signatures and dynamic attack detection
- Attempt to download attack payloads
- Search keyword indexing to draw attackers
- MySQL DB plus web console
- Integration with botnet monitoring & sandbox
- Check out Glastopf.org

Ghost

- USB Honeypot
- Runs on Windows
- Many malware spread across systems using thumbrive (and bypass network containment stragegies)
  - i.e. Stuxnet, Conficker
- Trick malware into thinking that a USB Thumbrive has been inserted
- Captures malware written on USB
- More: https://code.google.com/p/ghost-usb-honeypot

Thug

- Low Interaction Client-based honeypot to emulate web browser
  - Browser Personalities (i.e. IE)
  - Discovering Exploit Kits, Malicious Websites
- **Python vulnerability modules:** activeX controls, core browser functions, browser plugins
- **Logging:** flat file, MITRE MAEC format, mongoDB, HPFeeds events + files
- **Testing:** successfully identifies, emulates and logs IE WinXP infections and downloads served PDFs, jars, etc from Blackhole & other attack kits
- More information
  - http://www.honeynet.org/node/827

Tools and Projects

- Cuckoo Sandbox
- Visualization
- The Honeynet Project
  - HPFeed

Cuckoo Sandbox

- Automated Malware Analysis System
  - Why not just use Anti-Virus?
- Analyze Windows executables, DLL files, PDF documetns, Office documents, PHP Scripts, Python Scripts and Internet URLs
- Windows guest VMs in Virtual Box Linux
- Windows hooking / driver plus python modules for extracting and analysing sample executions

Cuckoo Sandbox (2)

- Trace of relevant win32 API calls performed
- Dump network traffic generated (pcap)
- Creation of screenshots taken during analysis
- Dump of files created, deleted and downloaded by the malware during analysis
- Extract trace of assembly instructions executed by malware process
- http://cuckoobox.org
- http://www.malwr.com

Visualization

- Many of the tools do not really have a GUI
- Reporting / Presentation is key
- Many visualization tools
  - PicViz
  - Afterglow
  - Gnuplot
  - Splunk
  - Plug-ins or front-end for many of the existing tools

The Honeynet Project

- The platform for those interested in running, building and learning from honeypots
  - http://www.honeynet.org
- Many Chapters from around the world
- Initiative for information sharing
  - HP Feeds
  - http://hpfeeds.honeycloud.net

Consider!

- Installing and playing with Honeypots to learn about security
- Joining the Honeynet Project as a chapter
- Sharing your experience and knowledge
- Happy Honeypotting!

<Demos>

- Kippo
- Dionaea

Questions?

- Thank You!
- Email adli@apnic.net