Honeypots

Overview

Introduction

Filtering network traffic

Examination of Datagram Fields



Introduction

Problem

Much like the problem for IDS/IPS

- Human attackers (hackers) and intruding software cost
- Threat intelligence is needed to prevent issues
- Ethically “stuck” on defensive end
- Almost everyone is a potential target
- Environment/technology/users keep changing

Exercise

1. Read section 2 in "How to Mock a Bear: Honeypot, Honeynet, Honeywall & Honeytoken: A Survey" (on DLE)
2. Read section II.B (page 3) in "A Survey on Honeypot Software and Data Analysis" (on DLE)

What is a honeypot?

* a resource that is designed to be attacked and compromised

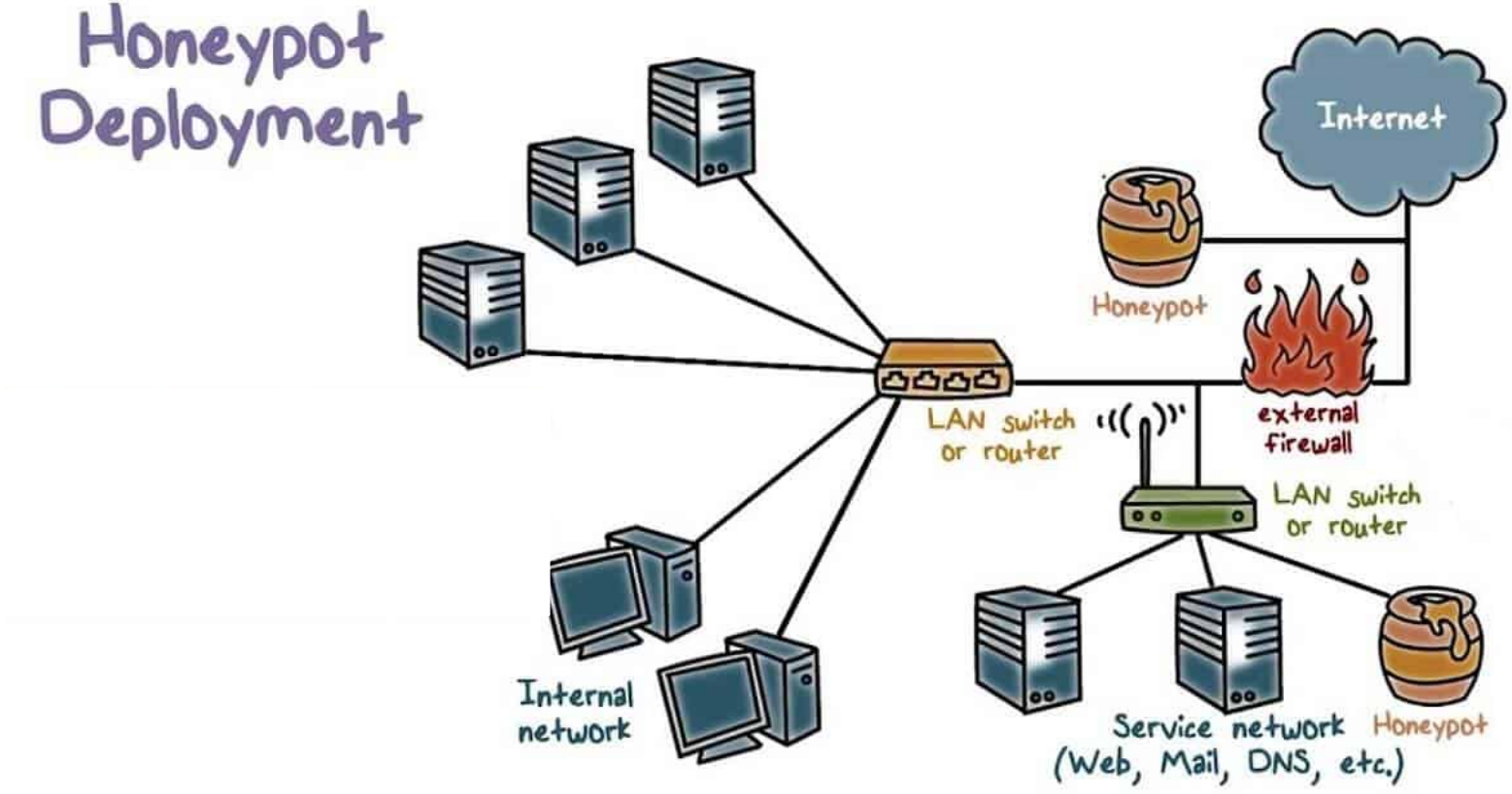
* expected to be probed, attacked, and exploited

* a single host, network device or daemon luring potential attackers to dis- tract them from valuable network resources

* Honeypots do not fix or prevent anything

* Honeypots are the “ultimate” information gatherers for SA

Architecture example



Types of Honeypots - Use

Research honeypot: This type of trap is used by developers, system administrators, and blue team managers working in institutions such as universities, colleges, schools and other related associations.

Production honeypot: This is used by private and public institutions, companies and corporations to investigate the behaviour and techniques of hackers seeking to attack networks on the Internet.

Will discuss levels of realism later, but research honeypots tend to be more detailed, but production honeypots tend to be easier to deploy

Types of Honeypots - context

Server honeypots: More often associated with production than research, placed inside the production network with other production servers by an organization to improve their overall state of security.

Client Honeypots are active security devices in search of malicious servers that attack clients. The client honeypot poses as a client and interacts with the server to examine whether an attack has occurred. Often the focus of client honeypots is on web browsers, but any client that interacts with servers can be part of a client honeypot (for example ftp, ssh, email, etc.).

Types of Honeypots - build

There is a difference between **physical** and **virtual honeypots**. A **physical honeypot** is a real machine on the network with its own IP address. A **virtual honeypot** is simulated by another machine that responds to network traffic sent to the **virtual honeypot**.

A virtual honeypot can emulated multiple systems, but can be more easily detected as a honeypot.

Pros

- Protect network through deception
- Early detection and prevention
- Better prepare system admins
- Better understanding of attacker actions & motives
- Protect real systems (hardening)

Cons

- Modern cyber criminals have more resources, and are often more organised and daring
- No matter what, as systems get more complicated, more errors and misconfigurations are common
- New attacks happen, more each year
- Organisations sometimes focus to much on prevention and not enough on detection (or even mitigation)
- Honeypots for detection are limited, as they often emulate virtual hosts (and emulated hosts can be detected with fingerprinting, timing, etc)
- Honeypots don't have a centralized deploying system – can be difficult to find one to suit needs

Levels of interaction

Low, mid, high (the more an attacker can interact with honeypot, the more information can be gained) but then more time to set it up

Low-interaction

Simple to install

Provides a few fake (emulated services)

No real operating system that attacker can operate on

-minimizes risks

-limits information

Hacker recon

If running honeypot (192.168.1.145) on other side of IDS

But can listen to network (netcat) listen on these so capture scans

Nc –l –p 80 //http

Nc –l –p 23 //telnet

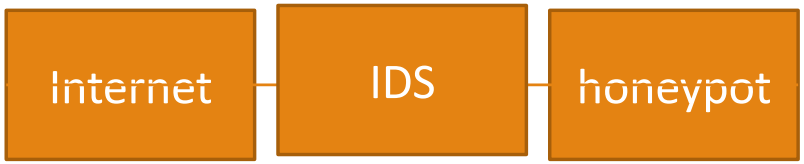
Hacker wants to know things like:

What OS running?

What ports open?

This lets them run specific exploits:

Nmap –sS –p 1-100 100.x.x.100/30 -O



flags
-SS TCP SYN stealth scan
-p scan ports in this range
-O fingerprints OS

What hacker sees

Nmap –sS –p 1-100 100.x.x.100/30 -O

- 192.168.1.145
  - Linux kernel [version]
    - Port 80 (http) - faked
    - Port 23 (telnet) - faked

Low risk but also minimal interaction – minimal information, no services hacked, and only able to listen, no interaction possible

However, this attempted connection is captured by IDS and logged

Mid-interaction

Provides more for attackers to interact with

- Services are emulated (like low-level)

- Unlike low-level, has more scripts for increased interaction, gives attacker more options in actions

Developing more complicated honeypot is more complex and time consuming. And if need several types, can be challenging. This also takes more expertise.

More complex if have multiple fake systems in a honeypot, which can also “communicate” with each other

High-Interaction Honeypot

Actual Operating system (OS) in place for attacker

Higher risk – not a phantom emulated system, can be used as a waypoint to attack the rest of the networks

With today's cutting-edge dynamic deception methods, a high interaction honeypot can adapt to each incident, making it far less likely that the attacker will realize they are engaging with a decoy.

Of course, the biggest downside to a high interaction honeypot is the time and effort it takes to build the decoy system at the start, and then to maintain the monitoring of it long-term

Honeypot examples

Spam honeypot: Also known as spam trap, this honeypot is specifically created to catch spammers before they hit legitimate email boxes. These often have open relays in order to get attacked, and work closely with RBL lists to block malicious traffic.

Malware honeypot: This type of honeypot is created to simulate vulnerable apps, APIs and systems for the purpose of getting malware attacks. The data that is then collected will later be used for malware pattern reconnaissance, to aid in creating effective malware detectors.

Honeypot examples

Database honeypot: Databases are a common target of web attackers, and by setting up a database honeypot you can watch and learn different attack techniques such as SQL injection, privilege abuse, SQL services exploitation and much more.

Spider honeypot: This type of honeypot works by creating false web pages and links that are only accessible by web-crawlers, not by humans. Once the crawler accesses the honeypot, it's detected along with its headers for later analysis, usually to help with blocking malicious bots and ad-network crawlers.

Types of security honeypots

- SSH
- HTTP
- WordPress
- Database
- Email
- IoT

And more specialised ones.

Visualisation – T-pot - an open-source honeypot framework



https://www.youtube.com/watch?v=G-_OabDowFU

Honeypot hunters

Fingerprinting, timing, etc, can make honeypots easier to detect, and the effort of making all honeypots high-interactive level is not feasible for most. This has led to a decrease in use, but:

Still interesting pieces to apply elsewhere

Still has its uses in specific cases

Could become more popular in the future if IoT, if some of the problem was emulating a complex device, if devices get less complex it may be easier to fool attackers.

Advantages of honeypots

Independent from Workload: Honeypots only need to process traffic directed at them or from them. This means, that they are independent from the workload which the production systems experience.

Data Collection: the data collected in a honeypot/net is not polluted with noise from other normal network activities and the ones setting up the honeypot can choose what data is high value, which makes the resulting dataset concentrated with relevant data. This makes data sets smaller and data analysis less complex.

Zero-Day-Exploit Data: If designed well, it can gather data for known exploit as well as gather data for new, unknown exploits.

Advantages of honeypots

Reduced False Positives and Negatives: Any activity with server-honeypots is an anomaly, which is by definition an attack. Client-honeypots verify attacks by detecting system state changes. These procedures result in reduced false positives and false negatives.

Flexibility: Honeypots are a very flexible concept as can be seen by vast amount of different honeypot software. This means, that well-adjusted honeypot tools can be used for specific tasks.

Disadvantages of honeypots

Limited Field of View: Server-honeypots have one common problem: they are worthless if no one attacks them. As long as attackers do not send any packets to the honeypot, it will be unaware of any unauthorized activity on production systems.

Getting Fingerprinted: Low-interaction honeypots emulate services, that means that their services might behave different than the real services, which can be used for fingerprinting honeypots and consequently detect them.

Risk to the Environment: If honeypots get exploited, they can introduce a risk into the user's environment. As discussed, the higher the interaction level, the higher the possible misuse.

What questions do you have?

If you owned a set of honeypots in a network, what kind questions do you have about possible attacks? This determines what data needs to be collected.

Honeypot Questions & Data

Do common attacks “origins” exist?

- IP Address / IP-Prefex
- Domain Name, URL (and URL details)
- Geographic location (e.g., country)
- Attacker UserID / Email and other user identifiers
- OS of attacker

Honeypot Questions & Data

What is the target of the attack

- IP
- Port and Transport Protocols
- Services
- Software clients, plugins
- Vulnerability (e.g., CVE)
- OS

Honeypot Questions & Data

Is the attacker human or script/machine?

- Keystrokes (errors, mistypes, speed)
- Time of day
- Speed and volume of attacks
- Complexity of actions

Honeypot Questions & Data

What are attack frequencies

- Number of attempts per time unit, and their origins
- Time between attacks
- Number of attack origins and targets
- Received packets/data per time unit
- Number/duration of time in a honeypot

Honeypot Questions & Data

Are there any new questions we need to answer with IoT etc?

- Tying back to situational awareness
- Self learning honeypots – monitor local network and dynamically deploys honeypots based on network makeup, what about the next level of ML and AI?
- Honey farm - centralized collection of honeypots and analysis tools
- Next generation deception technology / security
- Emulation on a larger scale (e.g., medical devices, hospital, ATMs)
- market size exceeding $1 billion by 2020 [Technavio]
- Market Research Media estimates the cumulative deception technology market value at $12 billion (2017–2022)

Honeytoken, honeynet

- A *honeynet* is a network of multiple honey- pots
- *Honeytokens* are just data in the form of files, entries within files, or special strings

Types of honeytokens

Fake email Addresses

Phishing, malicious attachments, malicious links to software and websites, email can also be used to observe / catch criminal activity.

Fake email accounts may be set up using the names of individuals not actually connected to your enterprise and left inactive but in plain sight on your organization's mail server, or at a company location which has a public-facing web server.

Since the fake addresses should in theory never be used, any received mail likely came from someone who got access to an internal mail server, email address list, or compromised public web server.

Types of honeytokens

Fake Database Data

Inserting fake records with enticing names or suggested content into an existing database is another method. These bogus data entries are the honeytokens which may lure hackers into stealing them, or malicious insiders into giving them away.

Either way, clues can be gained into how the attackers were able to exploit weaknesses or loopholes in your systems or network administration.

Types of honeytokens

Embedded Links That "Phone Home"

An alternative is to create honeytokens that exist as real data files (documents, programs, etc.) in your organization's file system, but with concealed and embedded links in them, fitted with a "phone home" switch.

Again, this method is likely to fail if an attacker takes precautions to isolate their system from the internet while opening the file.

Types of honeytokens

Web beacons

A web beacon consists of an internet link to a small object embedded within a file, such as a single-pixel transparent graphic. Being so small, this kind of honeytoken is unlikely to be spotted by the naked eye. And when a document with a beacon on it is opened, the web beacon will phone home details on the thief's system and location on the internet.

As with fake executables and embedded links, this method relies on an attacker not firewalling their machine against outgoing traffic to the internet, or via external ports.

Types of honeytokens

Fake Executable Files

A somewhat more extreme approach is to present honeytokens in the form of software programs or applications. These fake executable files are typically created with a "phone home" switch that activates whenever they're run. The switch relays details such as the hacker's IP address, names associated with their system, and so on, back to the organization that the files were stolen from.

The method is extreme because it's essentially a "hack back", on the part of the organization that set the trap. And it's one that may result in damage to the attacker's system, or violations to privacy and cyber-security laws in their home jurisdiction. In addition, the tokens only work if the attacker's machine is unprotected – e.g., by having its external ports to the internet blocked, while the program is run.

Types of honeytokens

Browser Cookies

Setting up browser cookies as honeytokens get around the blocked ports problem, and enables the people setting the trap to use methods similar to the ones that Google and other platforms employ to glean information about people who visit them.

This method has the most success by relying on human error – and the fact that hackers often act in groups or networks, not all of whose members may be careful about hiding their online activities (clearing browser caches, etc.).

Types of honeytokens

“Canary Traps”

These honey traps are constructed on the basis of a “canary” being like a whistle-blower, or snitch – someone within or associated with an enterprise, who “sings”, or gives out information that they shouldn’t. The associated honeytokens typically consist of some kind of tracer or marker that can tie an exposed piece of data to the individual who made it public.

Hollywood’s Screen Actors Guild (SAG) has had some notable success with this strategy, in exposing members of its organization who leak copies of movies submitted for Oscar consideration, to outsiders. Each copy of a film sent out to its membership bears a unique marker, associated with the recipient – which is replicated if they make copies for other people.

Types of honeytokens

Amazon Web Services (AWS) Keys

The huge and popular cloud platform Amazon Web Services (AWS) uses digitally signed keys to unlock various parts of its access management infrastructure. These may be placed on desktops, in text files, in GitHub repositories, and in numerous other places – and they're potentially valuable to cyber-criminals, for a number of reasons.

At the "jackpot" end of the spectrum, AWS keys could potentially be used to control a targeted organization's infrastructure. Lower down the success scale, the keys can grant hackers access to useful information, which might suggest other pathways or methods of gaining access to a corporate network.

To find out how far an AWS key can take them, the hackers have to test it out – and since Amazon Web Services keys have a built-in logging mechanism, they can be set up as honey tokens by security teams who monitor how they're being used. This honey trap approach has been brought to light by a successful test, reported in recent days.