Honeypots

Mathias Gibbens Harsha vardhan Rajendran

April 22, 2012

Outline

1. Introduction
2. History
3. Types of honeypots
4. Deception techniques using Honeypots
5. Honeyd
6. Service-specific honeypots
7. Deployment strategies
8. Pros / Cons
9. Real life uses
10. Improvements
11. Conclusion

Introduction

1. What is a honeypot?

Introduction

1. What is a honeypot?
2. What are the uses for a honeypot?

Introduction

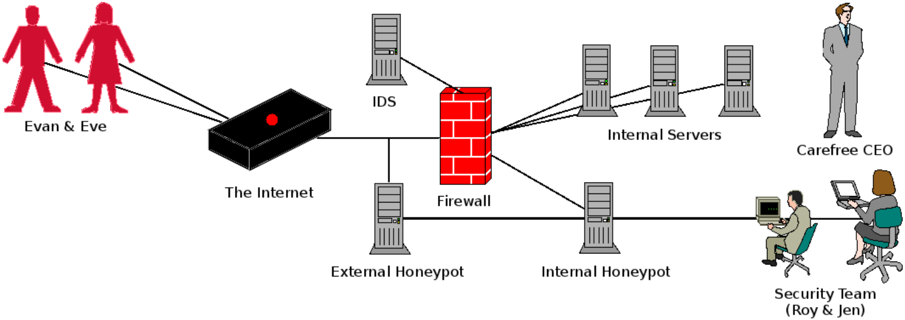


Figure: The key characters in our drama.

Introduction

1. Example of a logged attack: `http://goo.gl/phnI3`

History

1. Origin of the name

History

1. Origin of the name
2. Early manual entrapment by the Military

History

1. Origin of the name
2. Early manual entrapment by the Military
3. Cheswick at AT&T Bell
   "*I wanted to watch the cracker's keystrokes, to trace him, learn his techniques, and warn his victims. The best solution was to lure him to a sacrificial machine and tap the connection. ... Though the Jail was an interesting and educational exercise, it was not worth the effort. It is too hard to get it right, and never quite secure. A better arrangement involves a throwaway machine with real security holes, and a monitoring machine on the same Ethernet to capture the bytes.*"

History

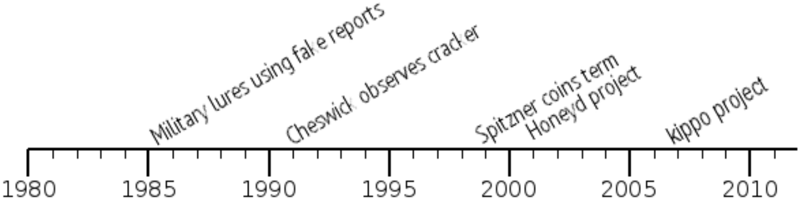


Figure: Honeypot development milestones.

Types of honeypots

1. There are many ways to classify honeypots

Types of honeypots

1. There are many ways to classify honeypots
2. The most common is by the amount of interaction provided to the malicious user: *high*, *medium*, or *low*

Types of honeypots

1. There are many ways to classify honeypots
2. The most common is by the amount of interaction provided to the malicious user: *high*, *medium*, or *low*
3. Other ways are by looking at the data collected and whether or not more than one honeypot is being used

Types of honeypots

Interactive

1. **Low-interaction** Emulates a single service; must be simple

Types of honeypots

Interactive

1. **Low-interaction** Emulates a single service; must be simple
2. **Medium-interaction** Emulates a group of services that could be expected on a server

Types of honeypots

Interactive

1. **Low-interaction** Emulates a single service; must be simple
2. **Medium-interaction** Emulates a group of services that could be expected on a server
3. **High-interaction** Full OS is presented to attacker; most useful, but also most risky

Types of honeypots

Type of data collected

1. Various types of data can be collected:

Types of honeypots

Type of data collected

1. Various types of data can be collected:
2. Events

Types of honeypots

Type of data collected

1. Various types of data can be collected:
2. Events
3. Attacks

Types of honeypots

Type of data collected

1. Various types of data can be collected:
2. Events
3. Attacks
4. Intrusions

Types of honeypots

System configuration

1. Stand alone

Types of honeypots

System configuration

1. Stand alone
2. Honeyfarm presenting a unified appearance to attacker

Uses of honeypots

1. Production environments to provide information and warning

Uses of honeypots

1. Production environments to provide information and warning
2. Security research trying to keep a step ahead of new attacks

Uses of honeypots

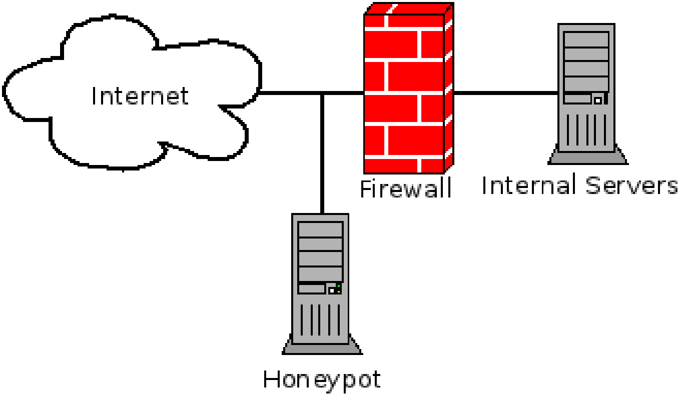


Figure: A example of an exposed honeypot.

Honeypots as mobile code throttlers

1. Principle: Infected machines make more connections than regular ones

Honeypots as mobile code throttlers

1. Principle: Infected machines make more connections than regular ones
2. Sacrifice a few machines for the common good

Honeypots as mobile code throttlers

1. Principle: Infected machines make more connections than regular ones
2. Sacrifice a few machines for the common good
3. Prevents a virus from spreading across the network, but cannot save the system

Honeypots as mobile code throttlers

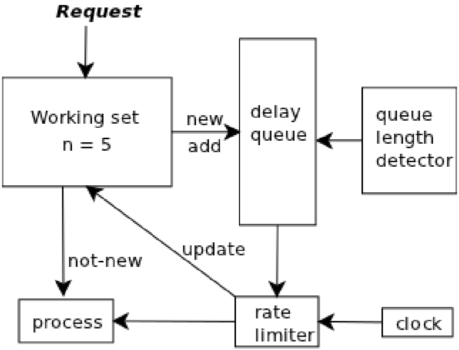


Figure: Virus throttling

Honeytokens (cost-effective honeypots)

1. Reiterate Honeypot definition: an information system resource whose value lies in the unauthorized or illicit use of that resource.

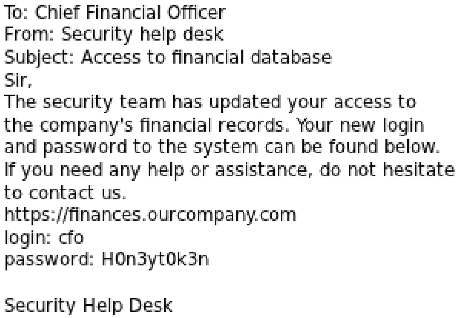
Honeytokens (cost-effective honeypots)

1. Reiterate Honeypot definition: an information system resource whose value lies in the unauthorized or illicit use of that resource.
2. Honeytoken is a Honeypot which is not a computer, but a digital entity.

Honeytokens (cost-effective honeypots)

1. Reiterate Honeypot definition: an information system resource whose value lies in the unauthorized or illicit use of that resource.
2. Honeytoken is a Honeypot which is not a computer, but a digital entity.
3. Hospital DB example

Honeytokens (cost-effective honeypots)



To: Chief Financial Officer
From: Security help desk
Subject: Access to financial database
Sir,
The security team has updated your access to the company's financial records. Your new login and password to the system can be found below. If you need any help or assistance, do not hesitate to contact us.
https://finances.ourcompany.com
login: cfo
password: H0n3yt0k3n

Security Help Desk

Figure: Honeytoken

Honeyd - Introduction

1. Honeyd - Low interaction virtual honeypot

Honeyd - Introduction

1. Honeyd - Low interaction virtual honeypot
2. Deception through simulation of network stack

Honeyd - Architecture

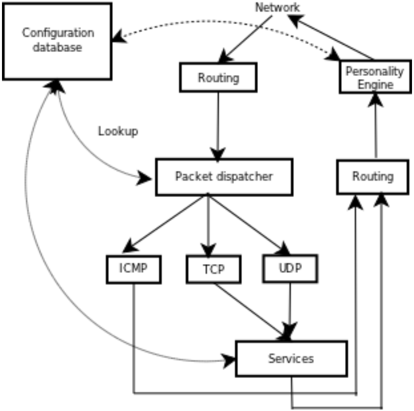


Figure: Honeyd architecture.

Service-specific honeypots

1. Simpler honeypots running for a specific service

Service-specific honeypots

1. Simpler honeypots running for a specific service
2. SSH honeypot (kippo)

Service-specific honeypots

1. Simpler honeypots running for a specific service
2. SSH honeypot (kippo)
3. Logs interactions for later analysis

Service-specific honeypots

1. Simpler honeypots running for a specific service
2. SSH honeypot (kippo)
3. Logs interactions for later analysis
4. Fairly safe to run on a computer, even if not dedicated

Service-specific honeypots

1. Simpler honeypots running for a specific service
2. SSH honeypot (kippo)
3. Logs interactions for later analysis
4. Fairly safe to run on a computer, even if not dedicated
5. This idea can be applied to other services as well

Deployment strategies

1. Sacrificial lamb

Deployment strategies

1. Sacrificial lamb
2. Deception ports on production systems

Deployment strategies

1. Sacrificial lamb
2. Deception ports on production systems
3. Proximity decoys

Deployment strategies

1. Sacrificial lamb
2. Deception ports on production systems
3. Proximity decoys
4. Redirection shield

Deployment strategies

1. Sacrificial lamb
2. Deception ports on production systems
3. Proximity decoys
4. Redirection shield
5. Minefield

Deployment strategies

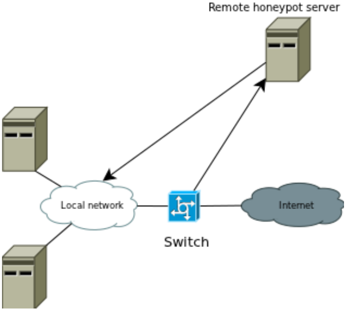


Figure: Redirection shield.

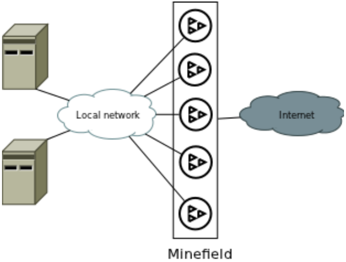


Figure: Minefield.

Honeypot Pros

1. Shield real servers from attacks

Honeypot Pros

1. Shield real servers from attacks
2. Gather information about current attack strategies

Honeypot Pros

1. Shield real servers from attacks
2. Gather information about current attack strategies
3. Limit risk to real data

Honeypot Cons

1. At best, just a copy of the real target

Honeypot Cons

1. At best, just a copy of the real target
2. Potentially prone to the same weaknesses as their copy

Honeypot Cons

1. At best, just a copy of the real target
2. Potentially prone to the same weaknesses as their copy
3. Additional time required to develop and maintain, in addition to real servers

Real life uses

1. Honeypots can play a vital role in a layered security setup

Real life uses

1. Honeypots can play a vital role in a layered security setup
2. At Utah State University as part of protecting their SSH servers: "*[Honeypots] make it easy to automate blocking SSH attackers, with virtually no chance of false positives.*"

Improvements

1. There is a constant battle between security researchers and hackers

Improvements

1. There is a constant battle between security researchers and hackers
2. Honeypots need to be updated to emulate newer servers and fix implementation bugs

Conclusion

1. Honeypots can be very useful as part of a comprehensive security setup

Conclusion

1. Honeypots can be very useful as part of a comprehensive security setup
2. Let us see the interactions of malicious users without their being aware

Conclusion

1. Honeypots can be very useful as part of a comprehensive security setup
2. Let us see the interactions of malicious users without their being aware
3. Versatile: many possible uses

Conclusion

Questions?