CSE508 Network Security

4/6/2016

Honeypots and Decoys

Michalis Polychronakis

Stony Brook University

Monitoring Unused Address Space

Dark space: chunk of unused, but routable IP address space

Background radiation: non-productive traffic

- Backscatter packets from flooding DoS attacks
- Port scanning activity
- Blind attack packets
- Benign traffic (broadcast packets, misconfigurations, …)

Why waste it?

- Use it for ***network telescopes*** and ***honeypots***

Network Telescopes

E.g., a whole /16 or /24 subnet – the larger, the better

- Multiple non-adjacent smaller chunks are also good

Observe arriving traffic using passive monitoring

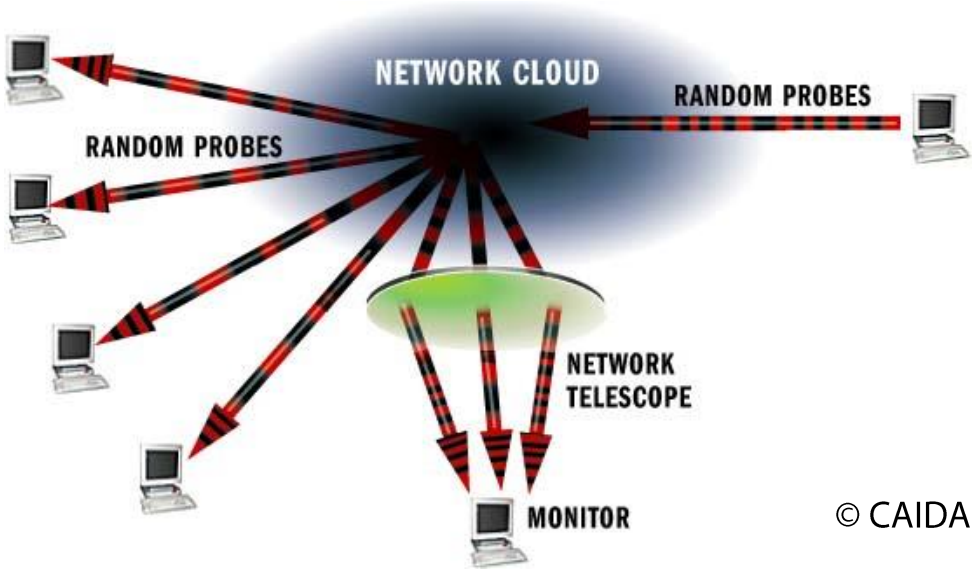
- Tcpdump, NetFlow, …

Active Responders

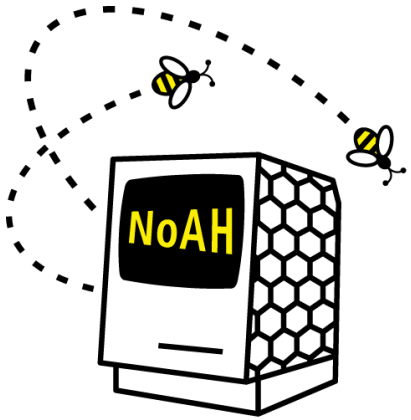
- Reply with SYN/ACK to elicit more traffic

Good only for global-scale events

- Blind to targeted attacks and wary adversaries



Honeypots



Computer traps

- Lure attackers, then spy on them
- Misdirect them and waste their resources

Two main categories

- **Low interaction:** scripts emulating real services
- **High interaction:** fully-blown systems (typically VMs)

Implemented using either virtual or physical machines

- Convenience vs. realism

Provide insight to adversaries' tools and tactics

- But cannot directly protect against them
- Just waste their time

Easily detectable: once noticed, can be avoided

- Or even worse: used for misdirection
- Dynamic malware analysis sandboxes/VMs face similar challenges

Honeyd (2004, Low-interaction Honeypot Example)

Framework for virtual honeypots and network services

- Simulate arbitrary TCP/UDP services on unallocated IP addresses
- Scalable design (hundreds of networks/thousands of hosts)

Operating system personalities

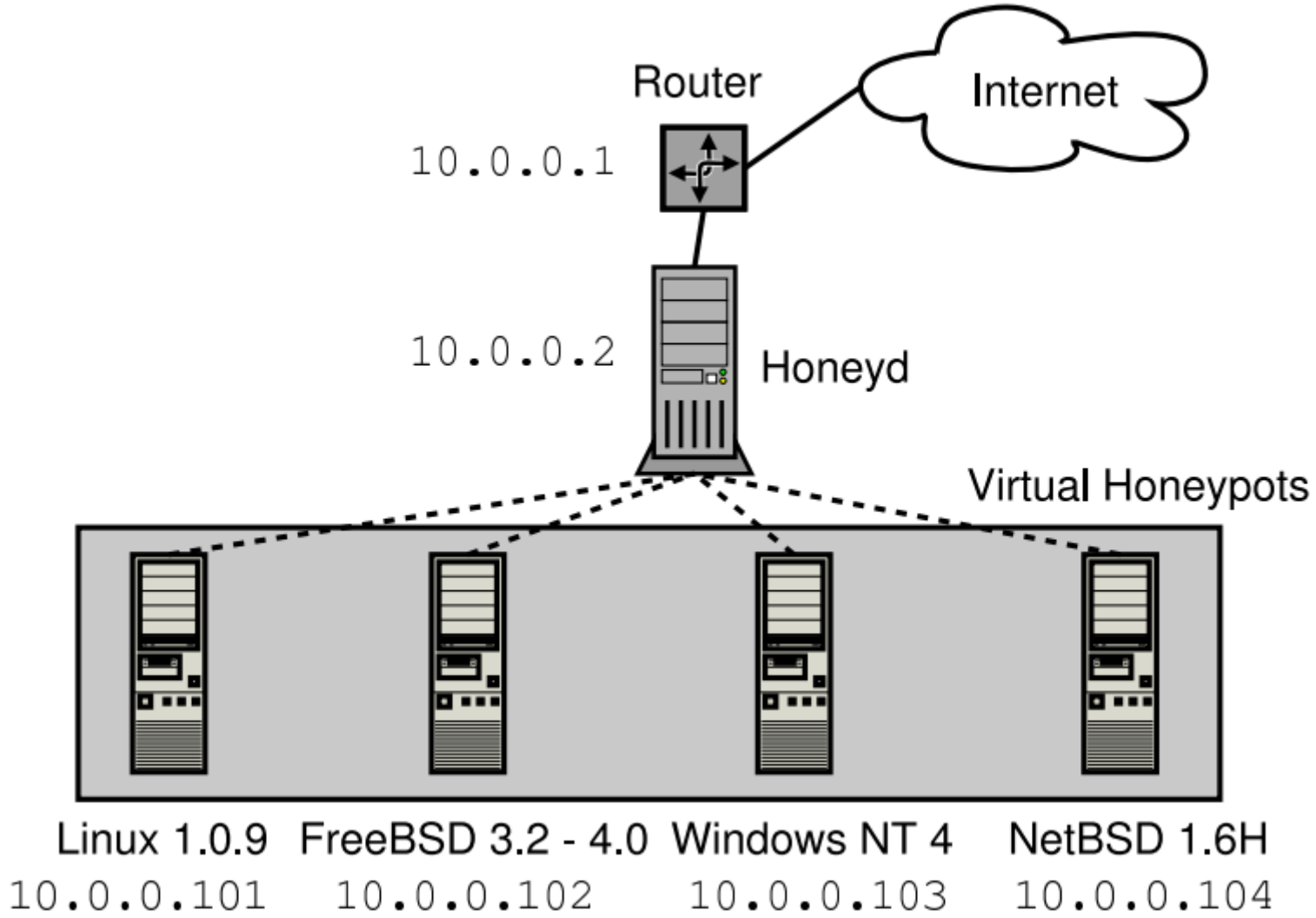
- Simulate only network stack behavior instead of every OS aspect
- Different operating system have different network stack behaviors: trick OS fingerprinting (e.g., using `nmap` or `xprobe`)

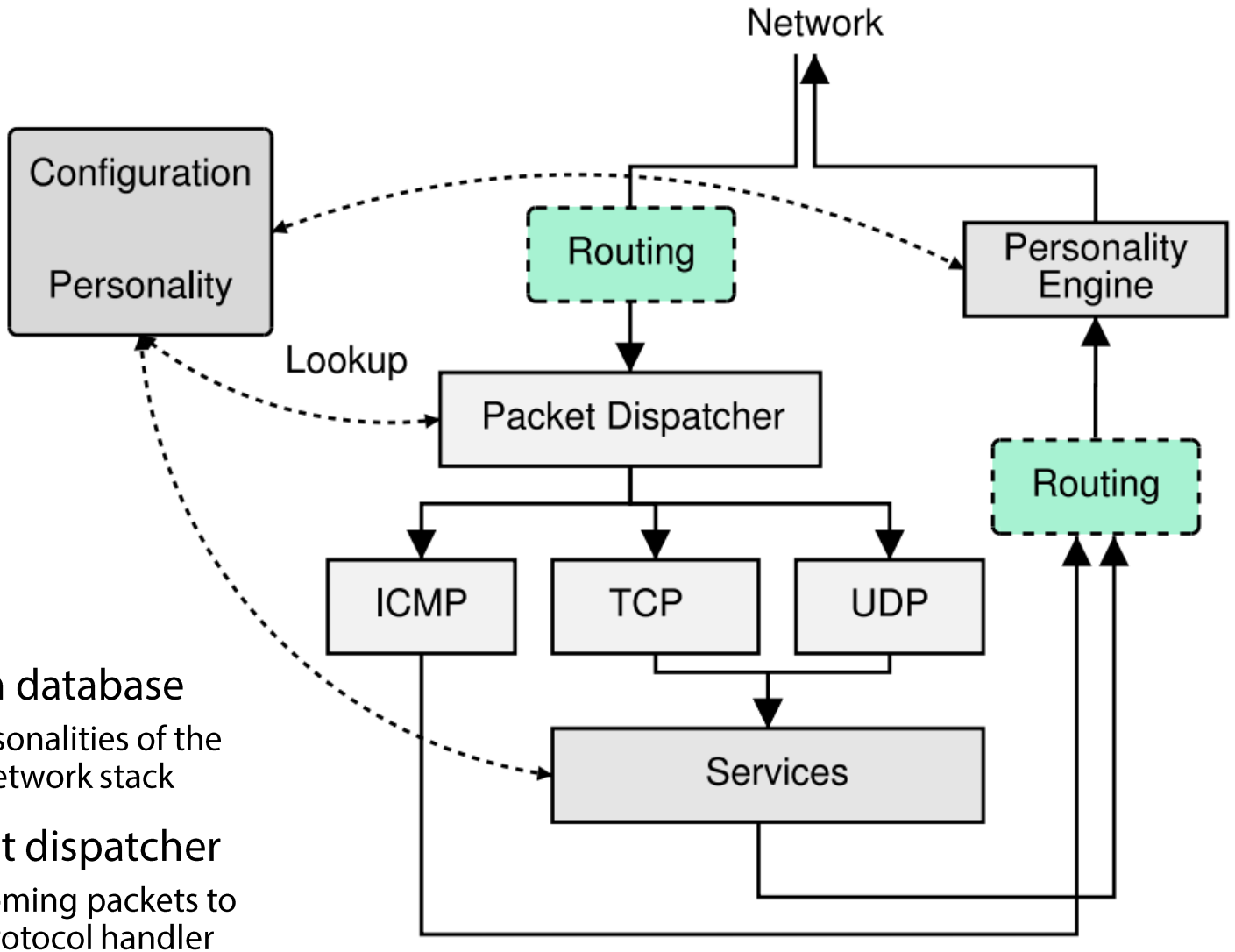
Arbitrary routing topologies

- Latency, packet loss, bandwidth

Supports integration of real systems

- Services can be proxied and redirected


Router
Internet
10.0.0.1
10.0.0.2
Honeyd
Virtual Honeypots
Linux 1.0.9
FreeBSD 3.2 - 4.0
Windows NT 4
NetBSD 1.6H
10.0.0.101
10.0.0.102
10.0.0.103
10.0.0.104



Configuration database

Store the personalities of the configured network stack

Central packet dispatcher

Dispatch Incoming packets to the correct protocol handler

Protocol handlers

Emulate different services

Argos (2006, High-interaction Honeypot Example)

Based on Qemu: open source system emulator

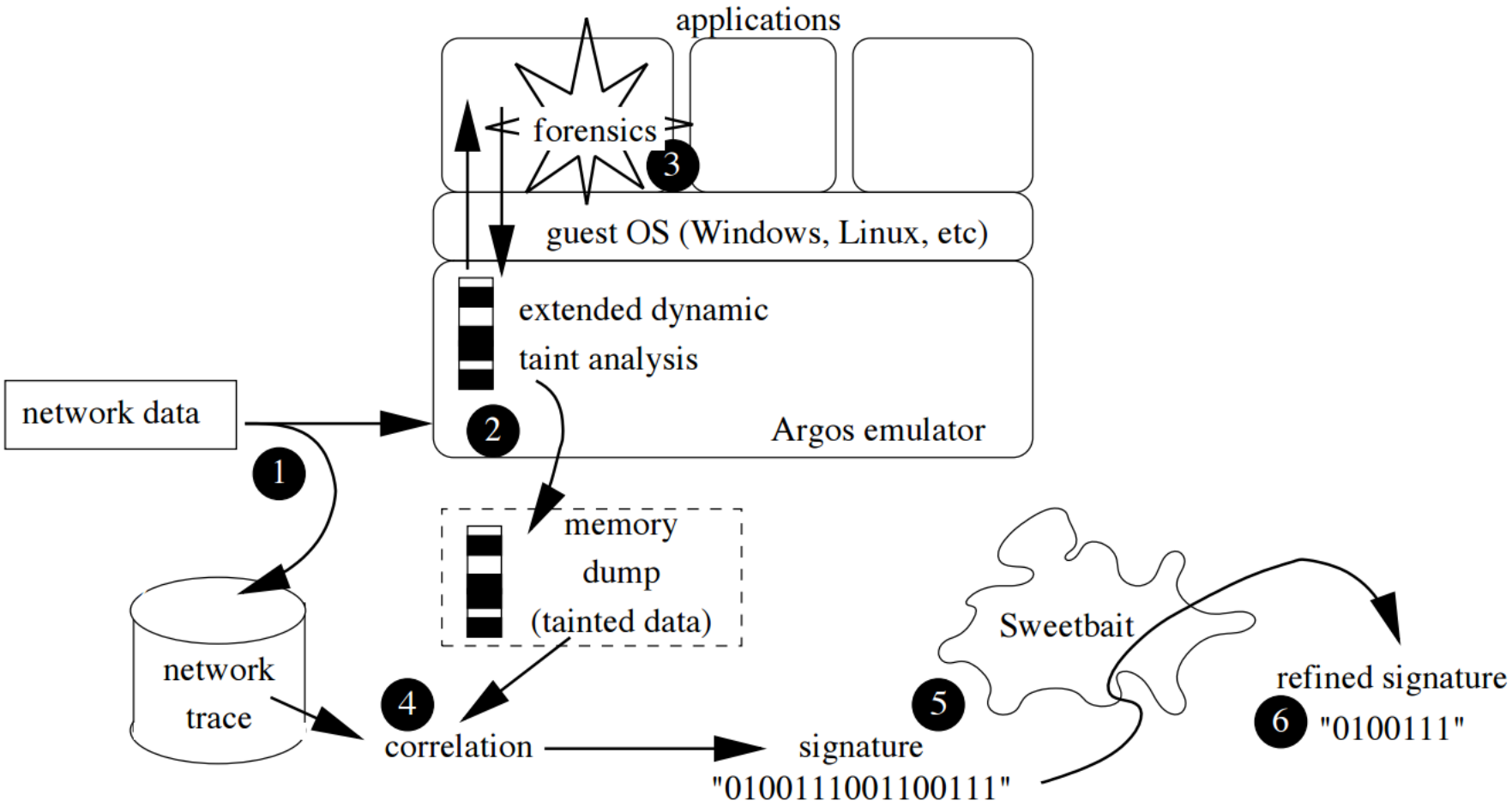
- Windows XP guest (or any other OS)

Zero-day attack detection

- Dynamic taint analysis
- Trace the usage of network input
- Alert and block execution when network-derived data end up being used as a program counter value

Detected attack analysis

- Relevant registers
- Memory footprint


applications
forensics
3
guest OS (Windows, Linux, etc)
extended dynamic
taint analysis
network data
2
Argos emulator
1
memory
dump
(tainted data)
network
trace
4
correlation
signature
"0100111001100111"
Sweetbait
5
refined signature
6
"0100111"

Client Honeypots (Honeyclients, Honeymonkeys, …)

Honeypots running fake services are good only for server-side attacks

- These days attackers go after client-side applications

Client honeypots pose as virtual clients and visit potentially malicious servers

- Using (potentially vulnerable) web browsers, document viewers, other client-side applications…
- Usually coupled with a crawler or other source of target URLs
- Underlying system is heavily instrumented to detect infection side-effects

May emulate user behavior

- Example: click on "download" buttons that may be part of social engineering attempts

Deception

Blur attackers' perception about what is real

- Detect suspicious (if not malicious) activity
- ***Honeypots:*** nobody should connect to them
- ***Honeyfiles:*** nobody should access them
- ***Honeytokens:*** nobody should use them

Examples

- Fake passwords: detect password DB leak
- Fake credit card numbers: detect credit card DB leak
- Fake email address: detect mailing list leak
- Fake document files: detect unauthorized access

Can help detect data exfiltration and insider threats

Deployment is not always trivial

- Should not affect usability
- Detection triggers should not reveal the decoys