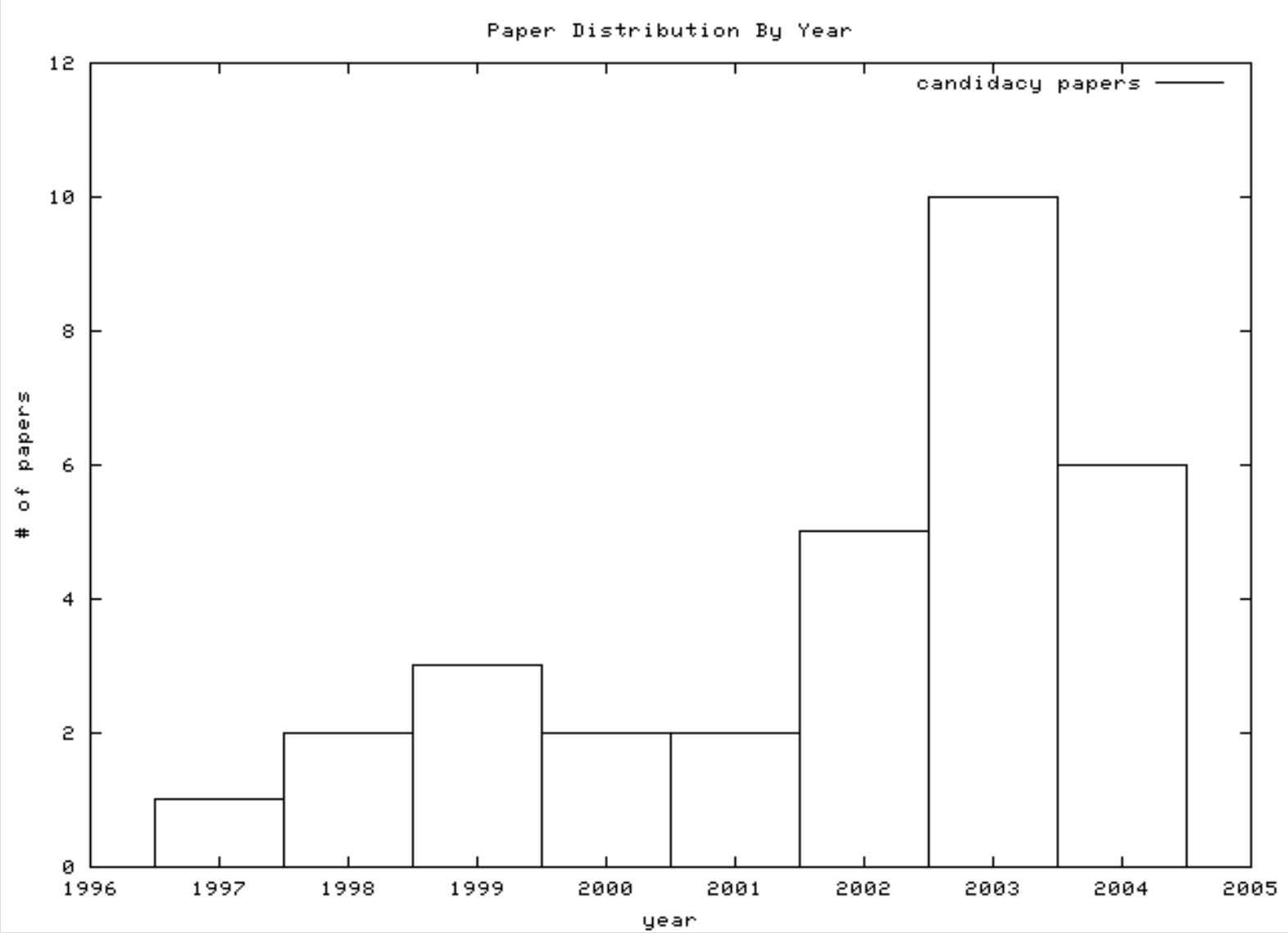
Host and Network Defense Systems for Intrusion Reaction

Michael Locasto
PhD Candidacy Exam
Network Security Lab
Columbia University

30 November 2004

The Literature



Defining Intrusion Reaction

- "...careful, rational, and automatic selection of an appropriate response to the threat or event of system penetration or subversion."
  - ROAR: Recognize, Orient, Adapt, Respond
- In general, notion of automatic response capability, largely unsupervised

Topics Not Covered

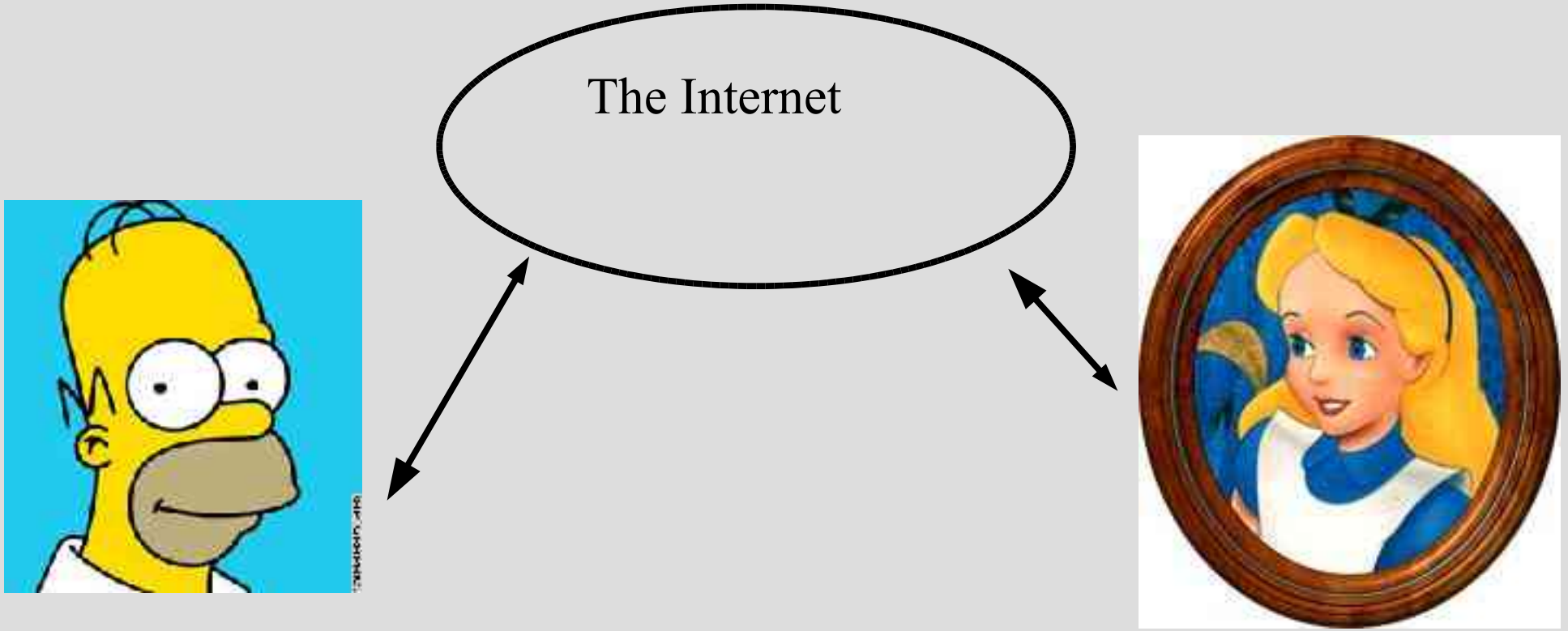
- Intrusion Detection [Wang2004]
- Fault Tolerance
  - Availability
  - Survivable Computing

- The essence of the talk is about tools and mechanisms that enable a system to nullify the effects of an attack

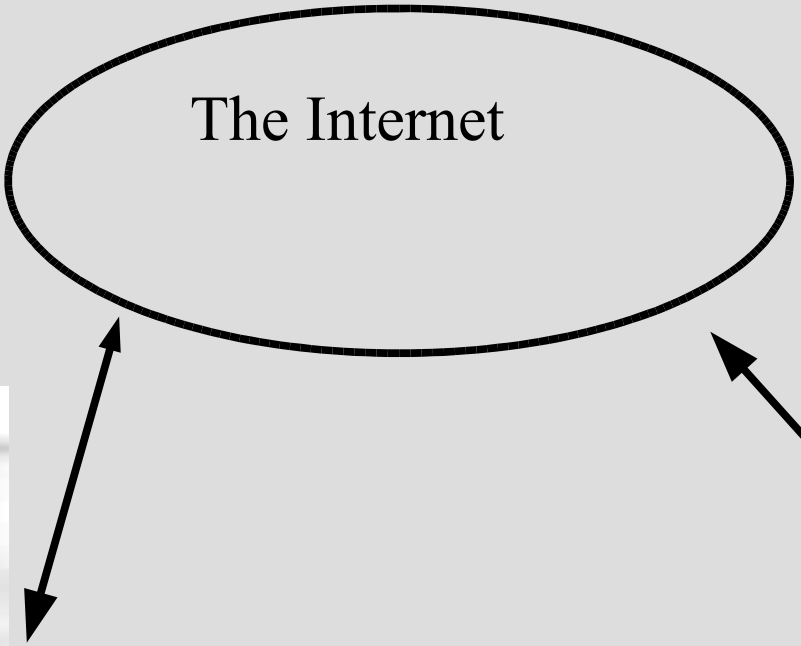
What's Ahead

- Overview of IR
  - Systems
  - Issues
- An examination of mounting a response
  - Where
  - How
- Open Problems
  - Hard Issues
  - Future Directions
- Questions

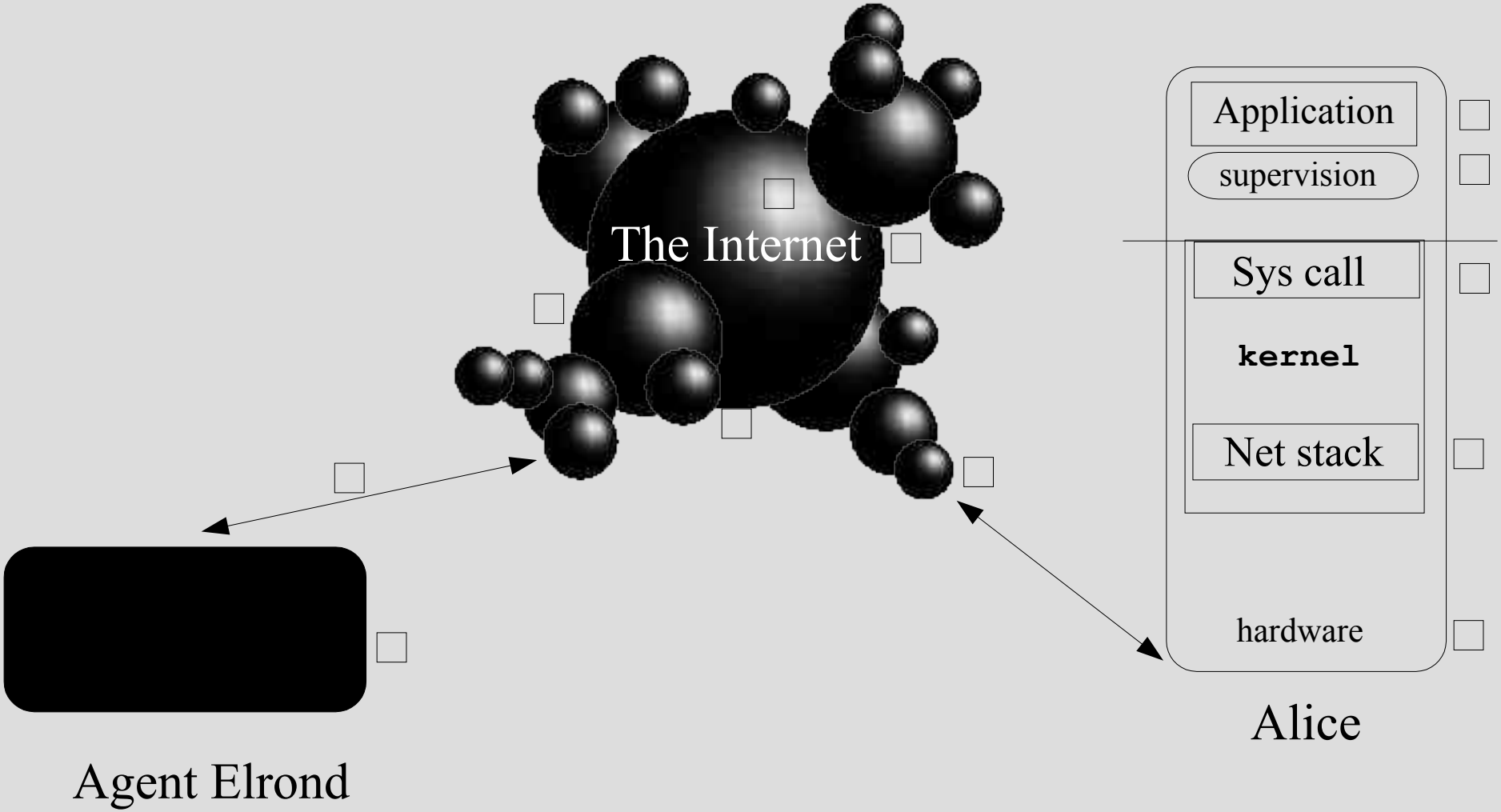
Alice and Homer



Alice and Agent Elrond



Attacker and Target



Recommendations

- Caution!
  - Automation only goes so far [Overill1998]
  - Technical issues (range of reactions, FP)
  - Legal issues
  - Ethical issues
- Where should we aim? [LaPadula1998]
  - Automate as much as possible
  - Areas:
    - Correlation
    - Forensics
    - Repair and damage control

Caution and Disclaimer

- Not very many of the following are actually deployed
  - Barriers to deployment are primarily social in nature
- All the systems have downsides
  - Performance
  - Realistic identification of attacks (detection)
  - Unrealistic deployment environment
- What's out there?
  - Sys call interposition
  - Network quarantine

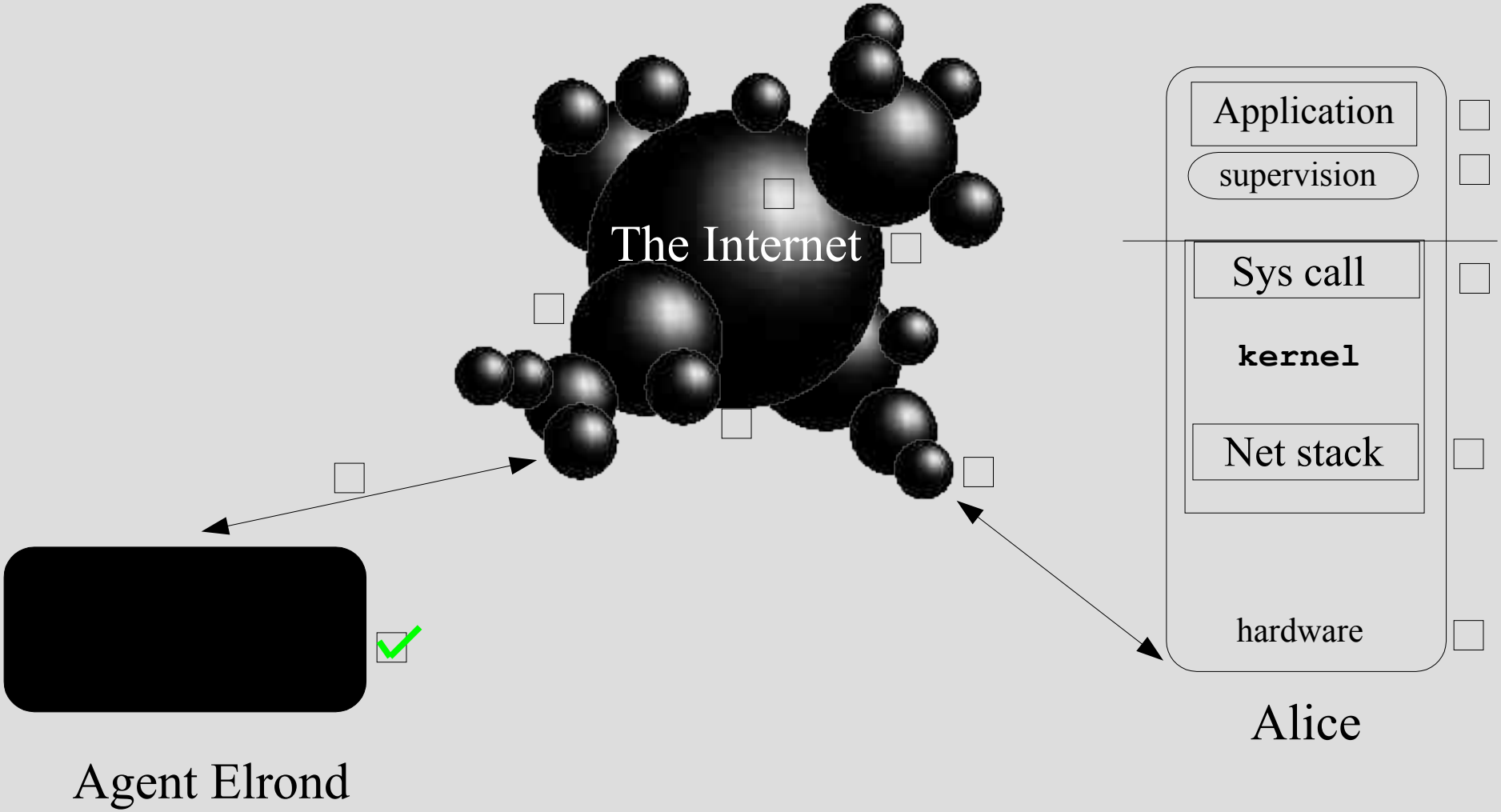
Frameworks

- Survivable Architectures
  - Artificial Immune System [Hofmeyr2000]
  - SABER [Keromytis2003]
  - APOD [Atighetchi2003]
  - HACQIT [Reynolds2003]
  - Crash Only [Candea2003]
- Worm Defense
  - Automated Defense System [Scandariato2004]
  - Worm Vaccine [Sirdiroglou2003]
  - Hybrid Quarantine Defense [Porras2004]
- Creating Diversity [Forrest1997]
  - Node Coloring [O'Donnell2004]

ARTIS [Hofmeyr2000]

- The Immunology Analogy
  - Epidemic models of worm infection
  - Much work by Forrest et al.
  - Fault injection [Ghosh1999]
- Model response system after immune system [Hofmeyr2000]
  - Distinguish self from nonself (anomaly detection) [Wang2004]
  - Isolate and remove pathogens
  - Maintain a homeostasis [Somayaji2002, Somayaji2000]

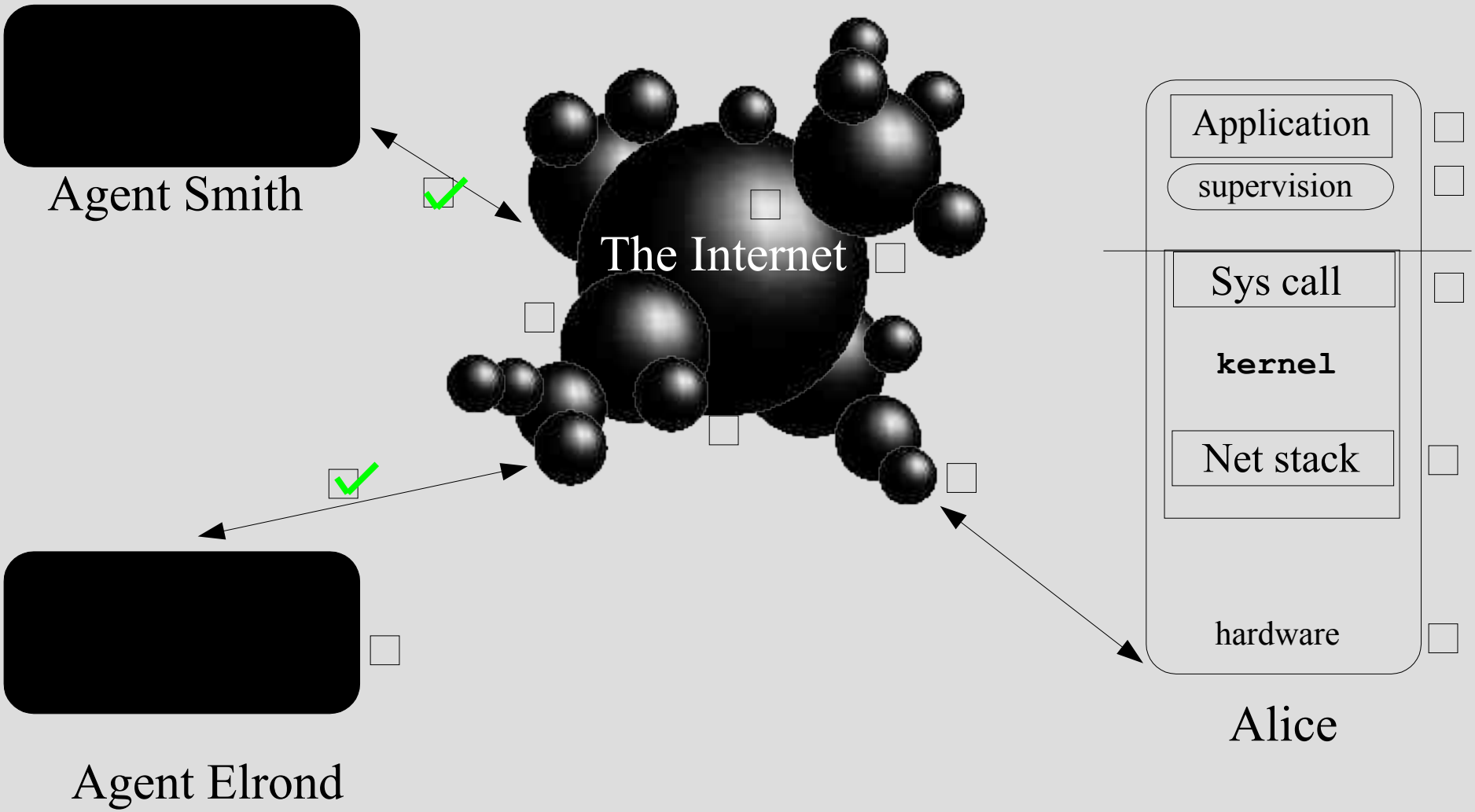
At the attacker [Bruschi2001]



Disable the Attacker

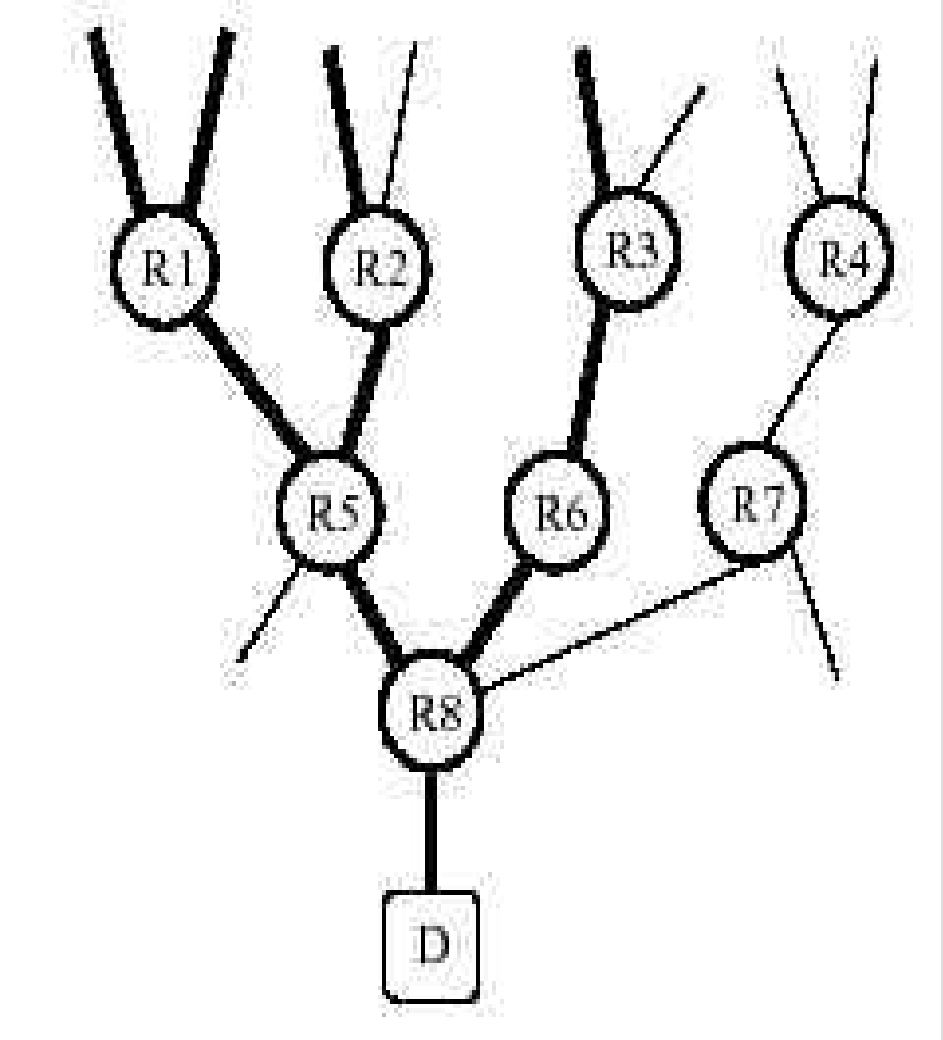
- Not as exotic as it sounds
  - AngeL uses system call interposition to
    - Identify outgoing network attacks
    - Identify local privilege escalation (detect sh code)
    - Stop local DoS (fork() bomb, etc.) by terminating process
- Rate limiting network connections
- “Outgoing” firewalls (proactive)
- Shutting off services (proactive)

At the Attacker's ISP [Ioannidis2002]

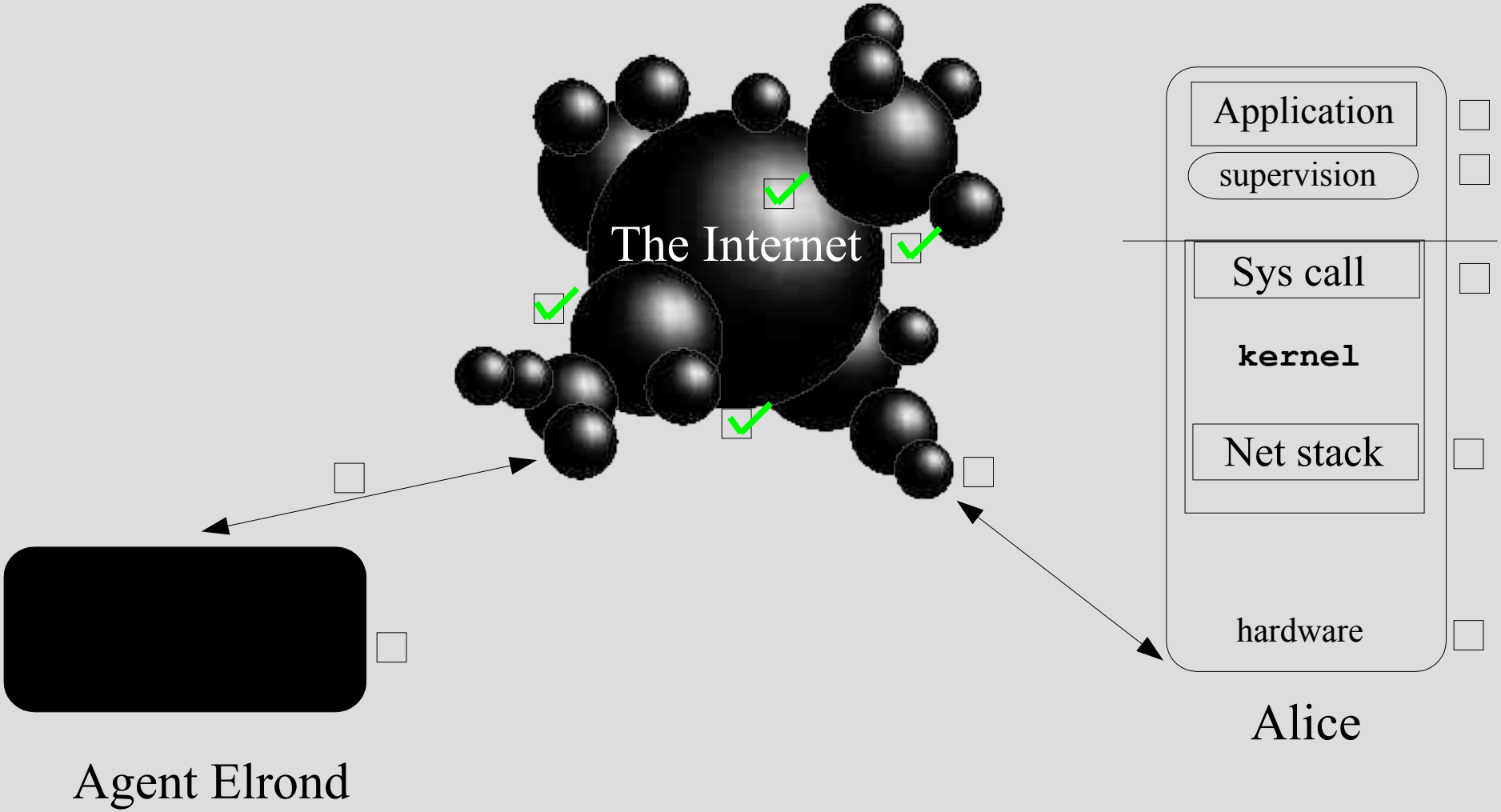


Hey! you! Get off of my cloud [Ioannidis2002]

- Pushback: attempt to recursively control network congestion
- “High Bandwidth Aggregates”
- Distinguish?
  - Flash crowds
  - DDoS
- Default: threshold on dst IP addr



Collaboration Across the Network



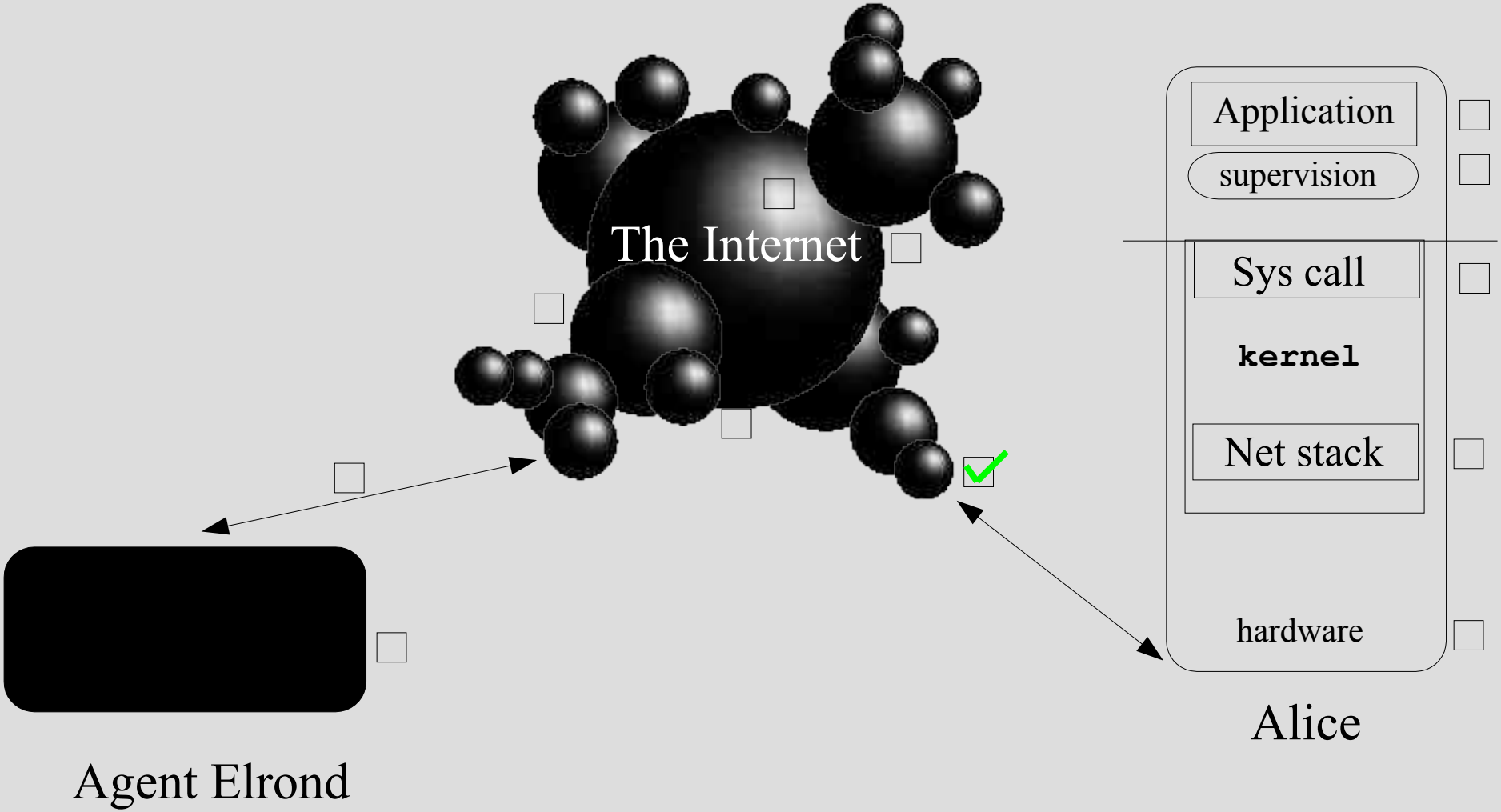
Tracing-based Active Intrusion Response [Wang2001]

- TBAIR triggered by IDS to identify “root cause” of intrusion
  - Injects watermark into reply
  - Internet-wide collaborating group of routers
- Recommended action is to “contain and isolate” intruders near their source
  - Similar to IDIP [Rowe1999]
  - Watermark notifies routers along the attack chain
  - One interesting idea is to move back a honeypot

Hybrid Quarantine Defense [Porras2004]

- Analysis paper of two worm defenses
  - Rate limiting (RL)
  - Friends-based communication (LA)
- Rate limiting is employed to allow the friend's protocol time to spread the word

The Local Network



Worm Vaccine [Sidiroglou2003]

- Key idea is to “catch” exploit in instrumented sandbox and generate a patch
- Test patch against regression suite
- Repair Heuristics
  - stack buffer -> heap, increase size, pmalloc
  - re-order fields in heap object
  - randomization techniques [Forrest1997]
  - content filtering
  - slice off

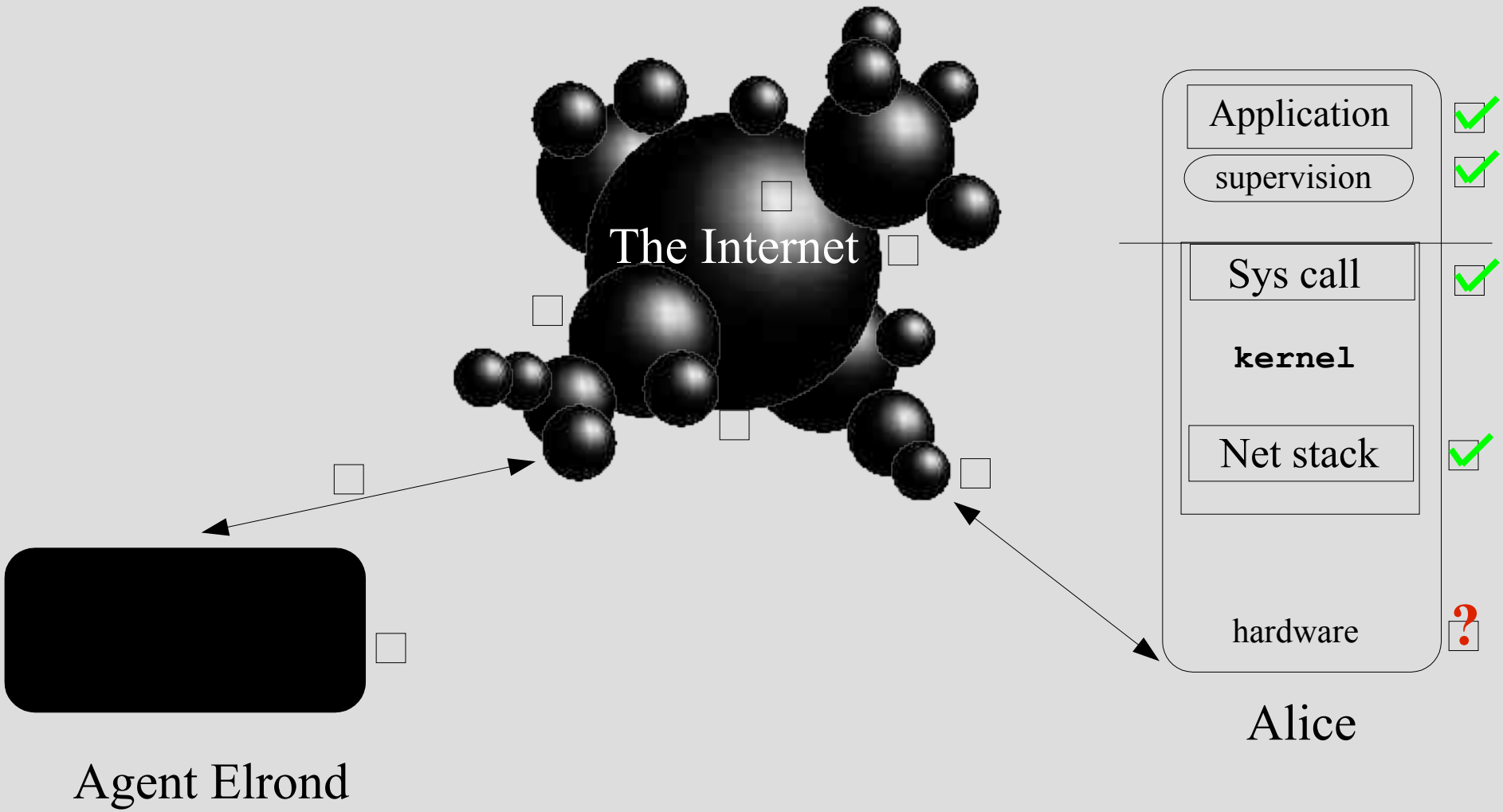
HACQIT [Reynolds2002]

- Not a general purpose architecture
  - Authenticated users (VPN)
  - Protected enclaves of COTS software
- Techniques
  - "Diverse process pairs" (hardware test analogy)
    - Test return results (e.g., HTTP status codes)
  - Attack learning via generate-and-test on failover
    - Attempts to generalize signature
    - Feedback loop to content-based blocking
  - Continual repair [Reynolds2003]
    - Monitor system logs, process & file creation

APOD [Atighetchi2003]

- Discusses network-based reaction mechanisms of the APOD project from BBN
- Mainly aimed at slowing an attacker down
  - Localized behavior becomes coordinated on a larger scale
- Techniques
  - Adjust firewall rules based on matched ID sigs [Rowe1999]
  - Rate-limit DoS traffic
  - Monitor ARP table & idle TCP connections
  - Port and Address hopping

At the Target Host



At the Host

- Countering Attacks
  - pH [Somayaji2000,Somayaji2002]
  - Program Shepherding [Kiriansky2002]
  - Systrace [Provos2003]
  - Failure oblivious computing [Rinard2004]
  - Shield [Wang2004]
  - ARB [Balepin2003]
- Cleaning up attacks
  - Data structure repair [Demsky2003]
  - Continual Repair for Windows [Reynolds2003]
  - Dynamic Access Control [Naldurg2003]

Shield [Wang2004]

- Exploit-generic, vulnerability-specific filter in the network stack
- Motivation
  - Patches are disruptive, not deployed quickly
  - Exploit signatures occur after the fact
- Mimic execution of application protocol
  - Monitor state and attack conditions
  - Drop or truncate bad traffic

Systrace [Provos2003]

- System call interposition w/ AD
  - Somehow the “right” level
  - Policy created during training
- Advantages of systrace
  - Addresses race conditions
  - Normalizes sys call args
- Disadvantages
  - SCI has large overhead
  - FP, need good training
- Response is encoded in return value of system call (error virtualization)

pH [Somayaji2000,Somayaji2002]

- Key idea is to enable the host to dynamically adjust back to 'normal'
- Anomaly detection on sequences of system calls
- Primary response is to “slow down” system call
  - Scaled exponential delay based on number of recent anomalies
  - Claim that attacker network connections time out, other tasks take unbearably long
  - Can disallow execve() call

Program Shepherding [Kiriansky2002]

- Control techinques
  - Restricted code origins
    - Don't jump into middle of lib
    - Return only to address after caller
  - Restricted control transfers
    - Requires verifying every branch instruction
    - Results are cached to amortize overhead
  - Uncircumventable sandboxing
    - Code must be entered/exited via SB checks
- Problems
  - Policy encoded in system, tradeoff between freedom and subversion

Failure Oblivious Computing [Rinard2004]

- Compiler mechanism that enables a piece of code to dynamically deal with the errors that lead to vulnerabilities
- Bury errors by
  - Manufacturing values for invalid read
  - Truncating input for OOB write
- Large test suite of software
- Problems
  - Truncated writes use error virtualization
  - Manufactured reads may cause infinite loops

Data Structure Repair [Demsky2003]

- Specify data structure constraints and properties (like assert)
- Tool to repair data structures
  - Standalone or (via program request, on error)
- Test cases
  - Ext2 (1.5 seconds)
  - Air traffic control software
- Key idea: disjunctive normal form, tries to satisfy all basic propositions of 1 conj
  - Each prop has at least 2 actions bound to make T or F

Conclusions

- Presented the relatively unexplored concept of *intrusion reaction*
  - Supplies an automated response to attack
- Various places in the network and on the host that a response can be intelligently mounted

Open Problems

- Designing robust methods that will work tomorrow, not just today
- Consider context
- The “Grand Unification Problem”
  - What is the Ideal Reaction System?
  - No cohesive way to protect the wide array of future, current, and legacy systems
- Detection is still a hard problem
- Researching the impact of reaction systems

Thanks

- To the committee
- To my external reviewers

Questions & Discussion

Appendix

Backup Slides

Network System Comparisons

System Name	Zero-day Attack Prevention	Non-Collaborative	Perf Impact	Encryption Problem
APOD	No	Yes	--	No
Hybrid Q. Defense	Yes (worms rate)	No	--	Sort of
TBAIR	No (depends on IDS)	No	Yes	Yes
Worm Vaccine	Yes (known attack class)	Yes	Sort of	No
Pushback	Yes (depends on aggregate)	Sort of	Potentially	Soft of
HACQIT	Yes (depends on sig gen)	Yes	--	No

Host Defense System Comparison

System name	AD	Effect of FP	Needs policy?	Perf Penalty
Systrace	Yes	Significant	No	Significant
PH	Yes	Significant	Yes	Significant
Program shepherding	No	Unknown	Yes, policy is hard coded	Amortized
Failure oblivious computing	No	Unknown	No	Little overhead

Downsides

- False positives are a showstopper for reactive systems
- Runtime overhead involved in checking policy
- Missing context, need flexible policy
- Unsafe/unpredicted error paths via “error virtualization”

FAQ

- What is the major problem for failure-oblivious?
  - Unexpected results propagating through program
- What is major problem for pushback?
  - Requires cooperation from neighbor ISPs
- What is the performance of Pushback?
  - Fairly good, implemented on FreeBSD, but has potential performance issues where policy decision may become a bigger factor in machines that have hardware routing
- What is shortcoming of Shepherding?
  - Policy encoded in system, not flexible, and is missing "context" information that can cause it to be bypassed

FAQ (2)

- What is problem with Systrace?
  - Sys call interposition is expensive and if training isn't good enough, too many FP
- What is the problem with pH?
- What is rational for FOC?
  - Short error propagation distances
- Why not just crash and reboot?
  - Too many stakeholders relying on the continued liveness of a server
- Why is the host most promising place to embed reaction mechanisms?
  - Network traffic volume is high, plus crypto problem

FAQ (3)

- Does Shepherding prevent an overwrite?
  - No, it doesn't prevent injection, it only prevents the last state: control transfer

Changing Access Control [Naldurg2003]

- A more fine-grained approach to isolation than traditional network connectivity cut off
  - Capability lists (CL's) for subject
  - Access Lists (AL's/ACL's) for object
- Two major issues:
  - How expensive is a particular rights representation to reconfigure?
  - How are trust relationships maintained across access control changes?
    - Enforce atomic updates and add *authorization guards*

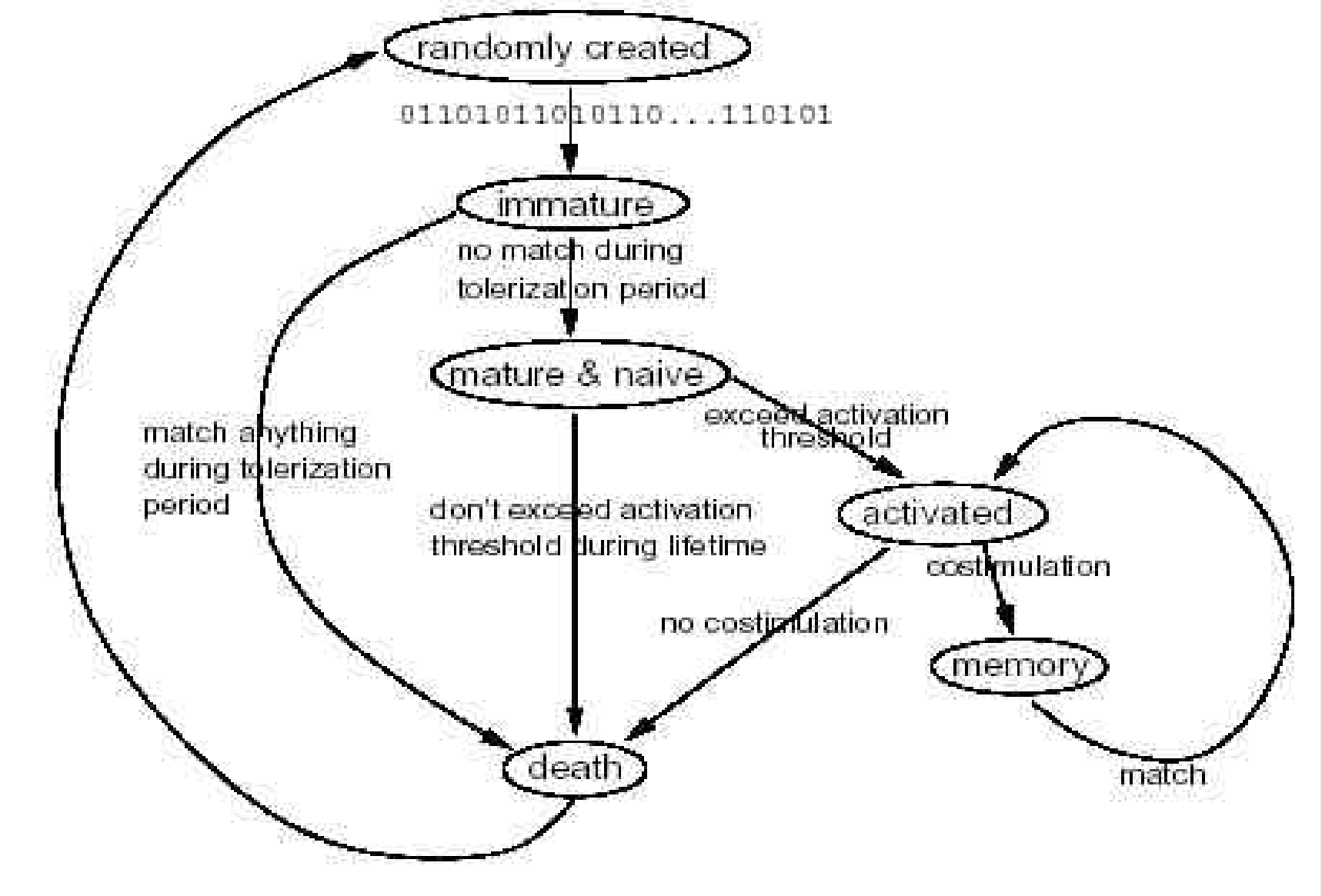
Specification-Based ID [Balepin2003]

- A static resource map drives each response
- System map reflects dependencies of “typed” resources
- Each node in the “system map” carries:
  - Activation condition
  - cost/weight
  - List of responses
- For full list of responses, see paper

ARTIS

- Breakdown:
  - Removal
  - Aging (stealthy attackers)
  - No centralized control, highly complex
  - Discrimmination: can reject healthy tissue from even very close genetic matches, gives identity
- Response
  - Effector functions
  - Antibodies are one particular effector, y shaped with tail isotypes

ARTIS detector training



Fault Injection [Ghosh1999]

Figure 1. Overview of the fault injection security tool.

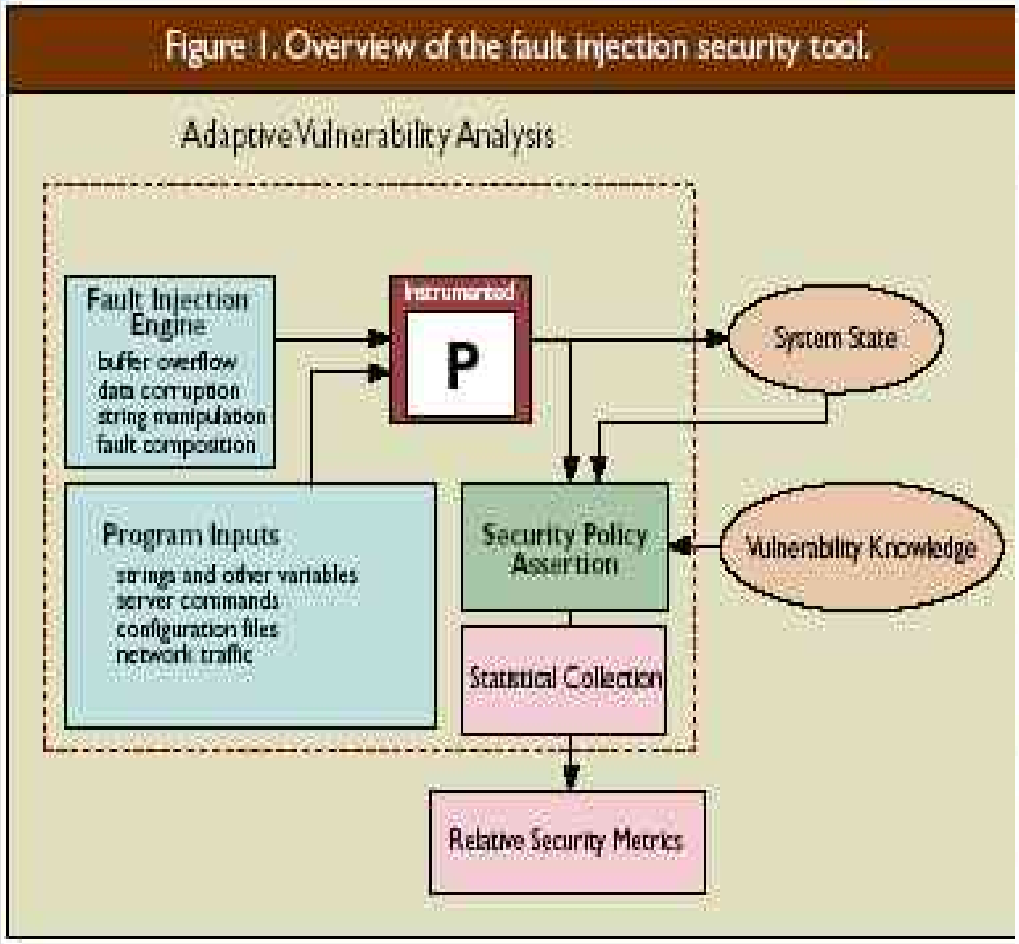


Table 1. Results from fault injection analysis of network daemons.

Program	Instrumented Locations	Successful Simple Corruptions	Successful Buffer Overruns	Function Coverage
Samba v1.9.17p3	1261	2	15	15.5%
NCSA http v1.5.2a	463	27	3	40.4%
wu-ftpd v2.4	476	1	3	38.67%
pop3d v1.305h	73	2	1	63.64%
kfingerd v0.07	146	2	5	38.1%

Crash Only Software

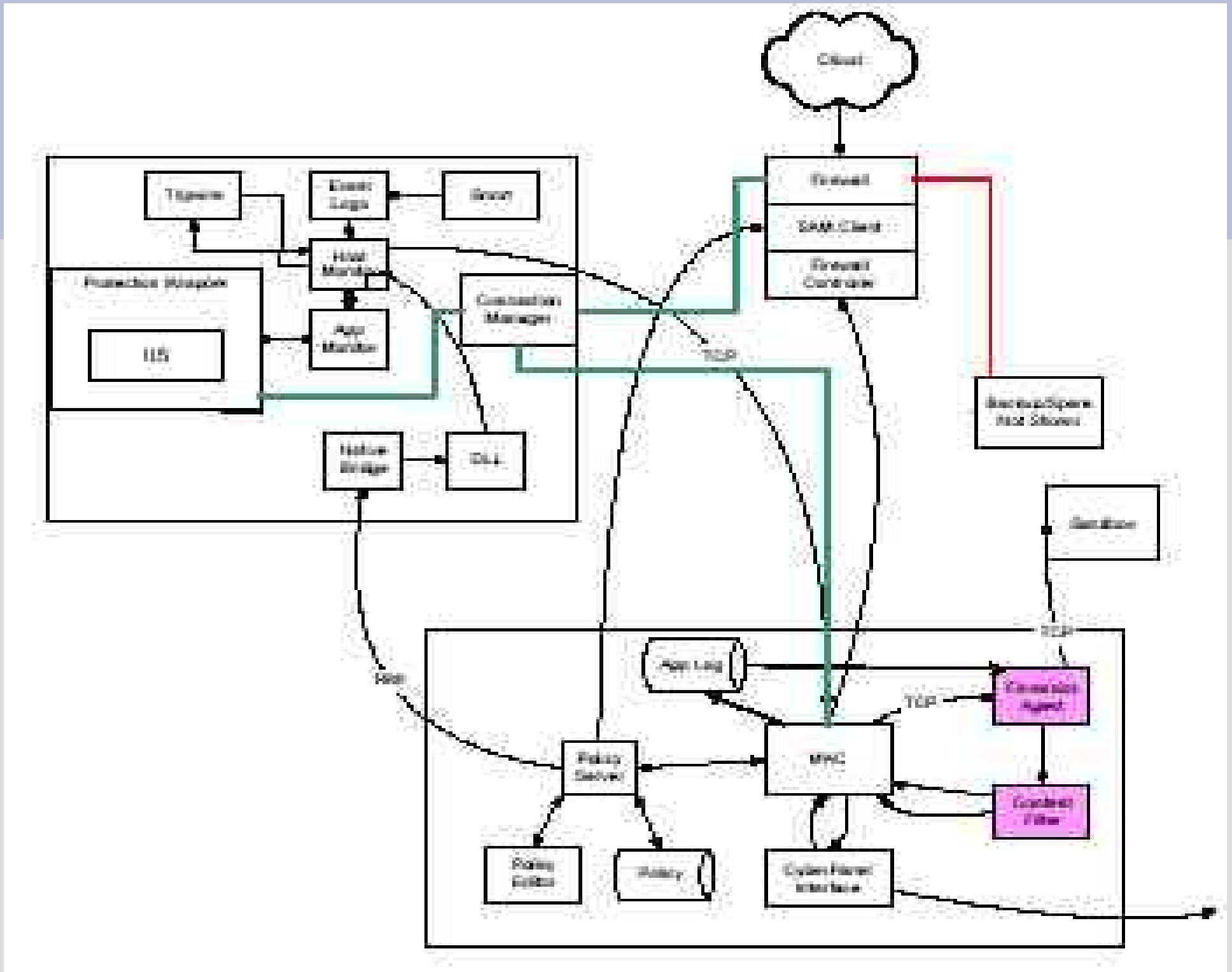
- Focused mainly on failure or resource exhaustion, not necessarily malicious adversary
- Recursive Micro-reboots
- Synchronized state stores
- crash=stop, start=recover
- External power kill switch

Building Diverse Computer Systems [Forrest 1997]

- Advocate a number of compile time, link, load time randomizations of software, including random stack padding, code block relocation, adding or deleting non-functional code, unique names for system files, permute args of functions

Feedback Control

- Elaborate set of mathmatical relationships for anomaly detection to see if the kernel is deviating from normal behavior. Recovery actions are to kill the process misbehaving.


Cloud
Firewall
SAM Client
Firewall Controller
Triggers
Event Logs
Server
HAM Monitor
Protection Wrapper
IIS
App Monitor
Connection Manager
TCP
Backup/Spare Host Servers
Native Bridge
DLL
Database
TCP
App Log
TCP
Connection Agent
Policy Server
MMC
Content Filter
Policy Editor
Policy
Custom Parser Interface

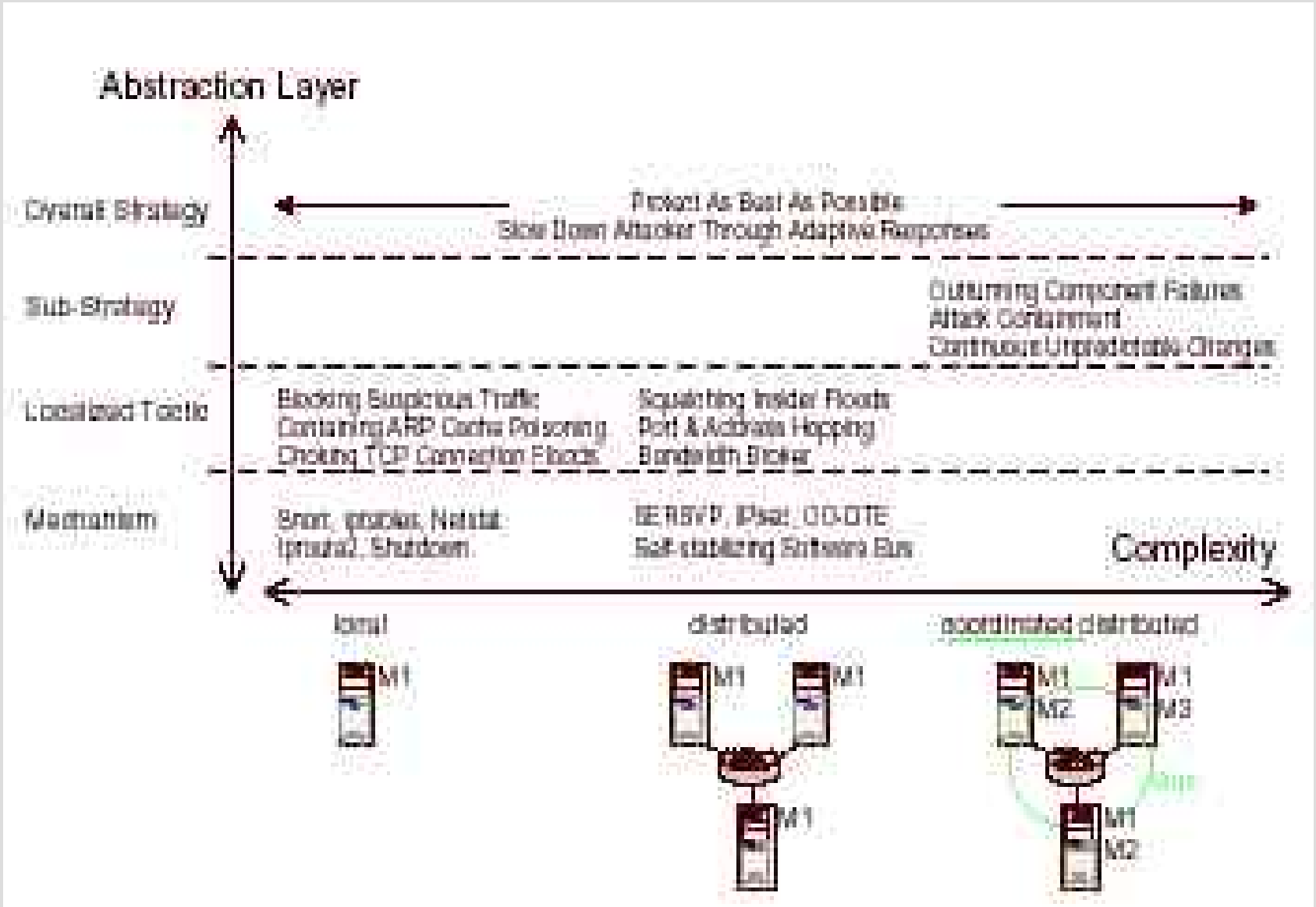
APOD Attacks

- Choking TCP Conn Floods
  - Threshold of open-but-idle TCP conns
  - Reject from src, but if attacker switches srcs or spoofs src...but spoofing is harder
- Blocking Suspicious Traffic
  - Use Snort to detect scan, blacklist src IP
  - Fails when src is spoofed, but can whitelist
- Containing ARP Poison
  - MAC binding, but admin overhead
- Squelching Insider Floods
  - Deploy gateways to rate limit on “mean”

APOD (cont.)

- High-level defense strategies
  - “Outrunning Component Failures”
  - “Attack Containment”
  - “Continuous Unpredictable Changes”
- Most local methods have glaring weaknesses
- Top-level strategy requires secure communication bus between components

APOD Defense Strategy Map



PAYL [Wang2004]

- An anomaly detector
  - A valuable trigger for reactive systems
  - Not a reactive system, but a building block for reactive systems to both detect and feedback information (as suggested by [Reynolds2002])
- Specialized to site or node
  - Calc freq distrib of payload and diff from training model for new traffic
  - Computes Model(contentlength,port)
    - Mahalanobis Distance (avg. weighted by variance)
    - Cluster if training data is sparse
  - Z-string serves as signature

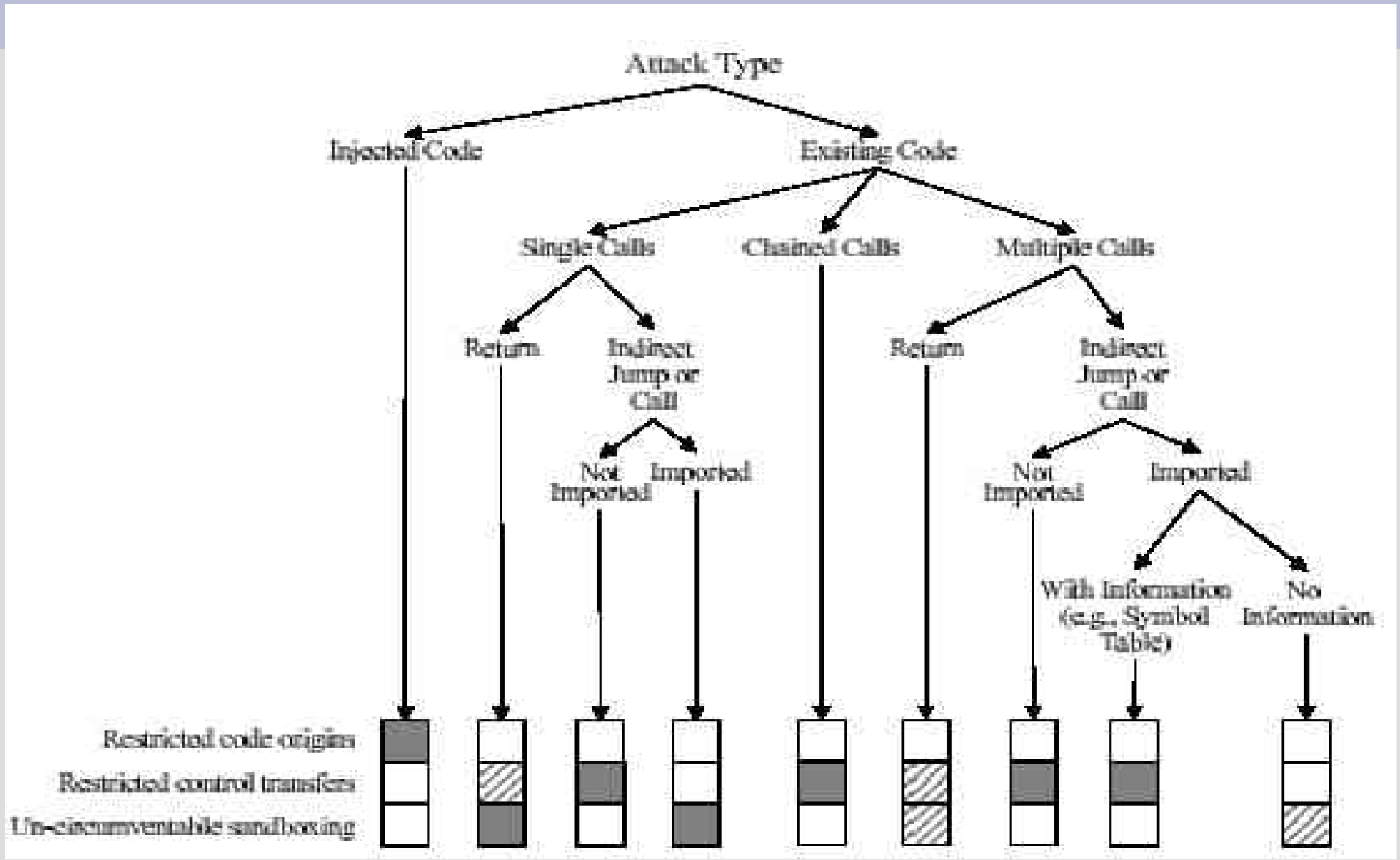
Systrace Policy Example

```
Policy:  /usr/sbin/named, Emulation:  native
     native-__sysctl: permit
     native-accept: permit
     native-bind: sockaddr match "inet-*:53" then permit
     native-break: permit
     native-chdir: filename eq "/" then permit
     native-chdir: filename eq "/namedb" then permit
     native-chroot: filename eq "/var/named" then permit
     native-close: permit
     native-connect: sockaddr eq "/dev/log" then permit
                  ...
```

pH

- They note that a user-level policy daemon could help give context to certain decisions
- Some training needs to be done

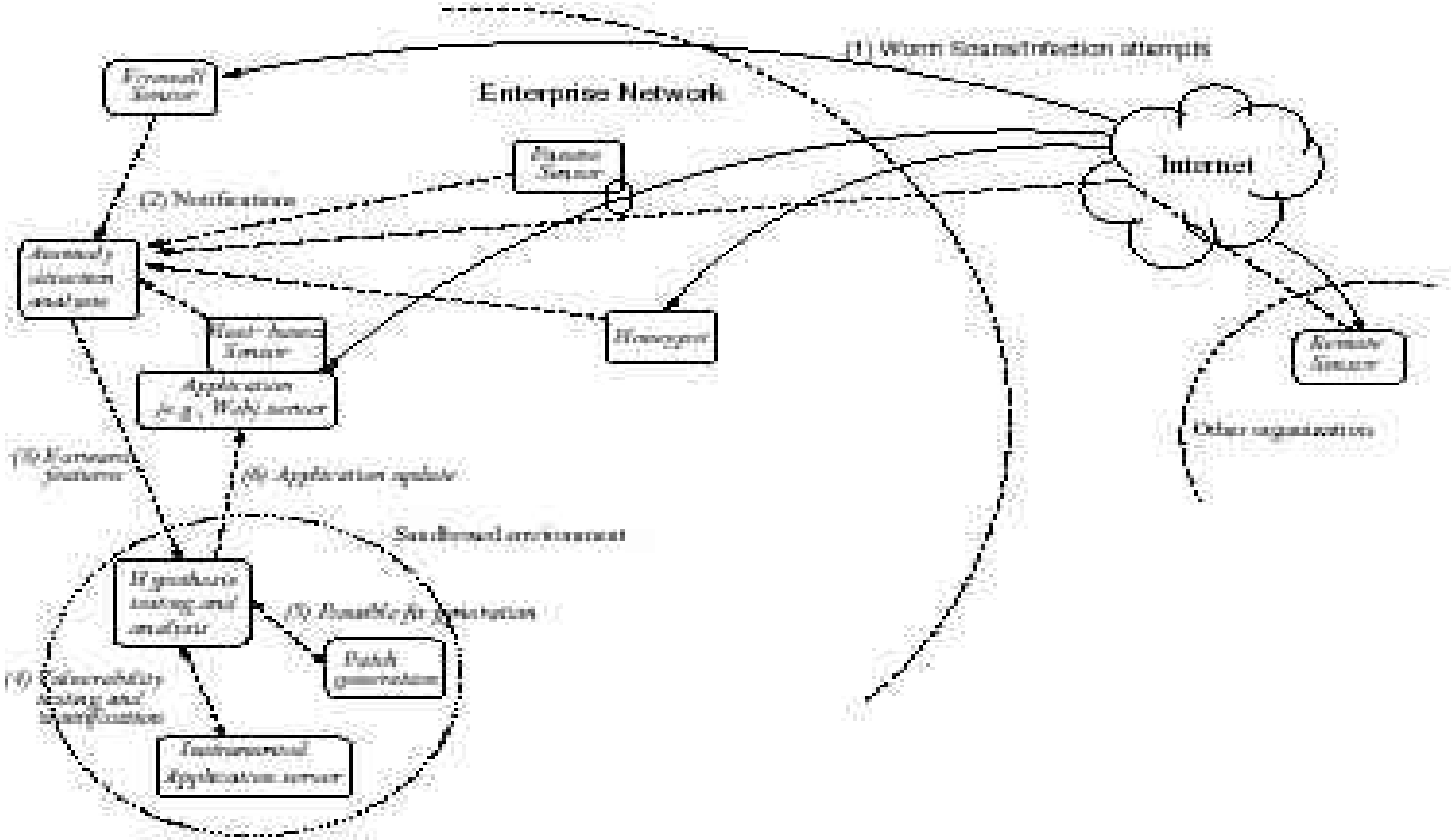
Program Shepherding



Program Shepherding

- Many ways to inject code and take advantage of already present code, so prevent final step
- Code origins (from disk, dynamically generated, modified)
- Current implementation has policy hard-coded into it
- Although stringing together existing code is an unlikely attack, glibc is imported into most programs, and the “system()” call and “execve()” call are available

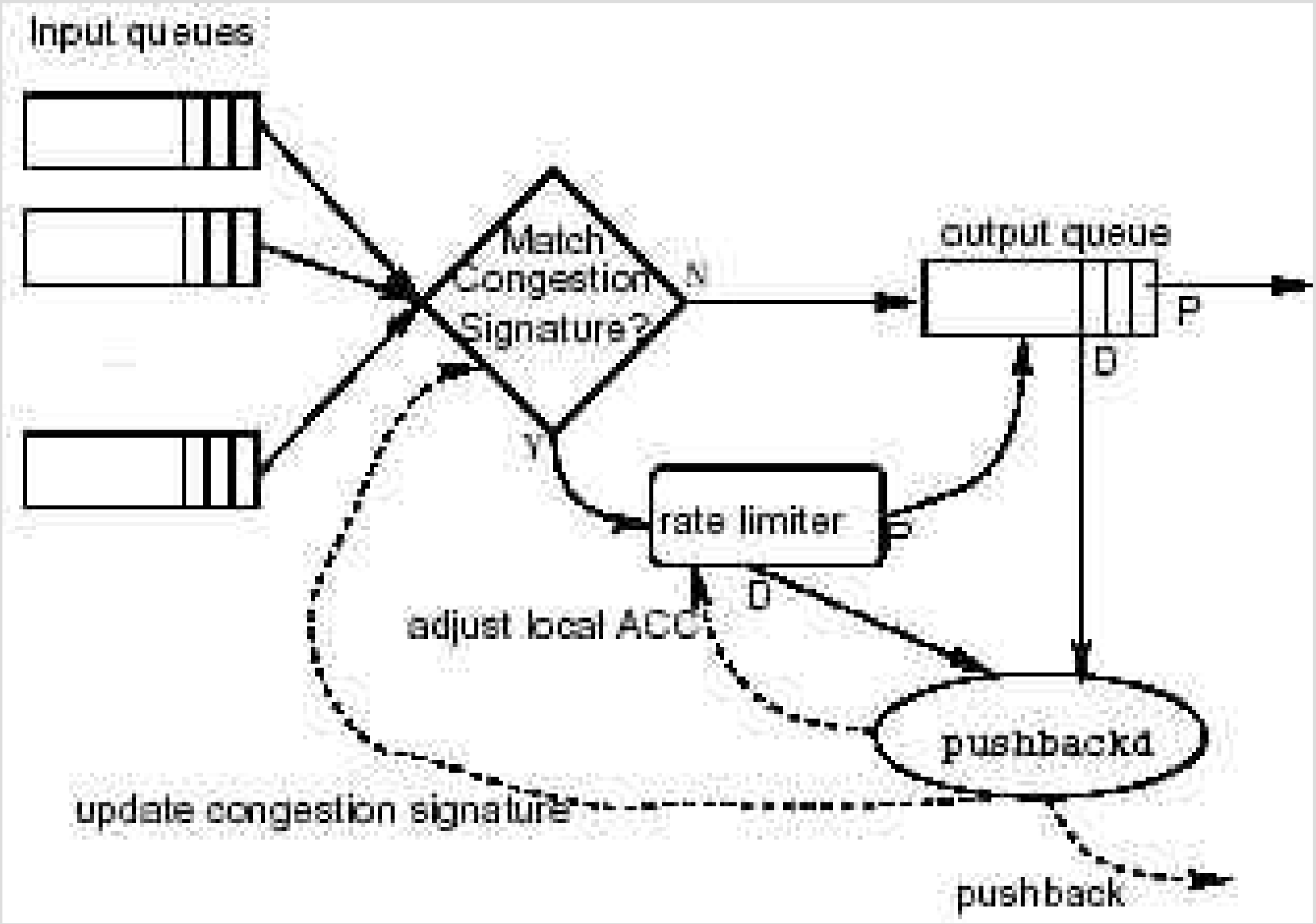
Worm Vaccine Architecture



Pushback Notes

- Rudimentary auth is a TTL of 255
- Can separate rate-limit logic from packet-dropping functionality
- Packets that make it to the output queue are **not** treated preferentially
- Pushback messages have a “depth” that acts as an initiator-set TTL for the pushback request

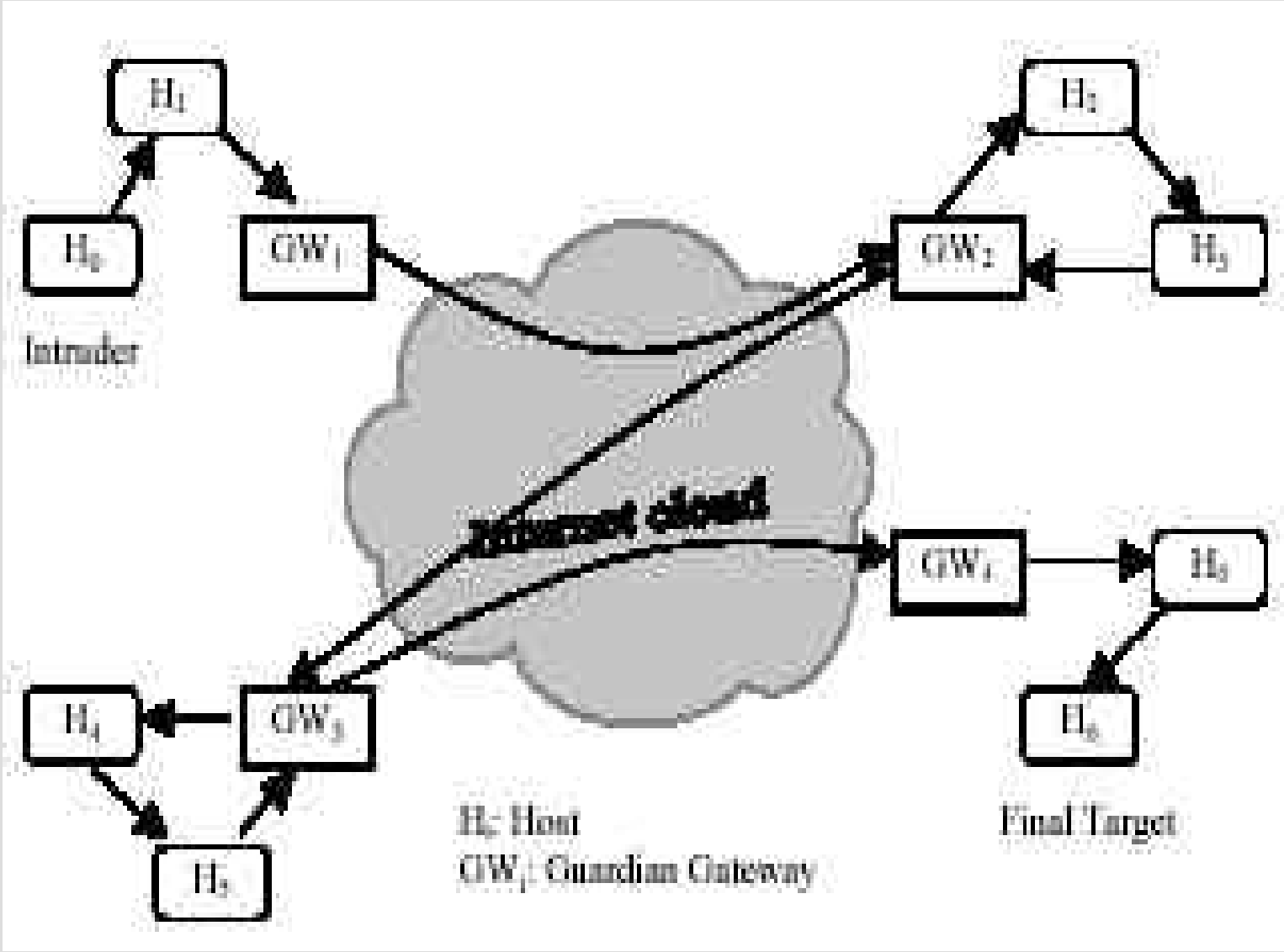
Pushbackd


Input queues
Match Congestion Signature?
N
Y
output queue
P
D
rate limiter
D
adjust local ACC
pushbackd
update congestion signature
pushback

TBAIR (cont.)

- Assumes no link-link encryption
- Suggested responses suffer from the collaboration problem
- One interesting response is to move a sandpot/honeybox back along the collaborating routers
- Not clear what traceback gets you
  - Innocent bystander
  - Cross national borders

TBAIR


H_1
H_2
H_0
GW_1
GW_2
H_3
Intruder
Internet cloud
GW_4
H_5
H_4
GW_3
H_6
H_7
H_0: Host
GW_i: Guardian Gateway
Final Target

SABER [Keromytis2003]

- A fusion of different research efforts
- Good detection mechanisms
  - Antura/SD
  - RAD, FWRAP
- Protection & Communication
  - SOS
  - MEET
- Response mechanisms
  - Auto-patch (based on Worm Vaccine)
  - Auto-move (based on ZAP)