Real Time Threat Mitigation Techniques

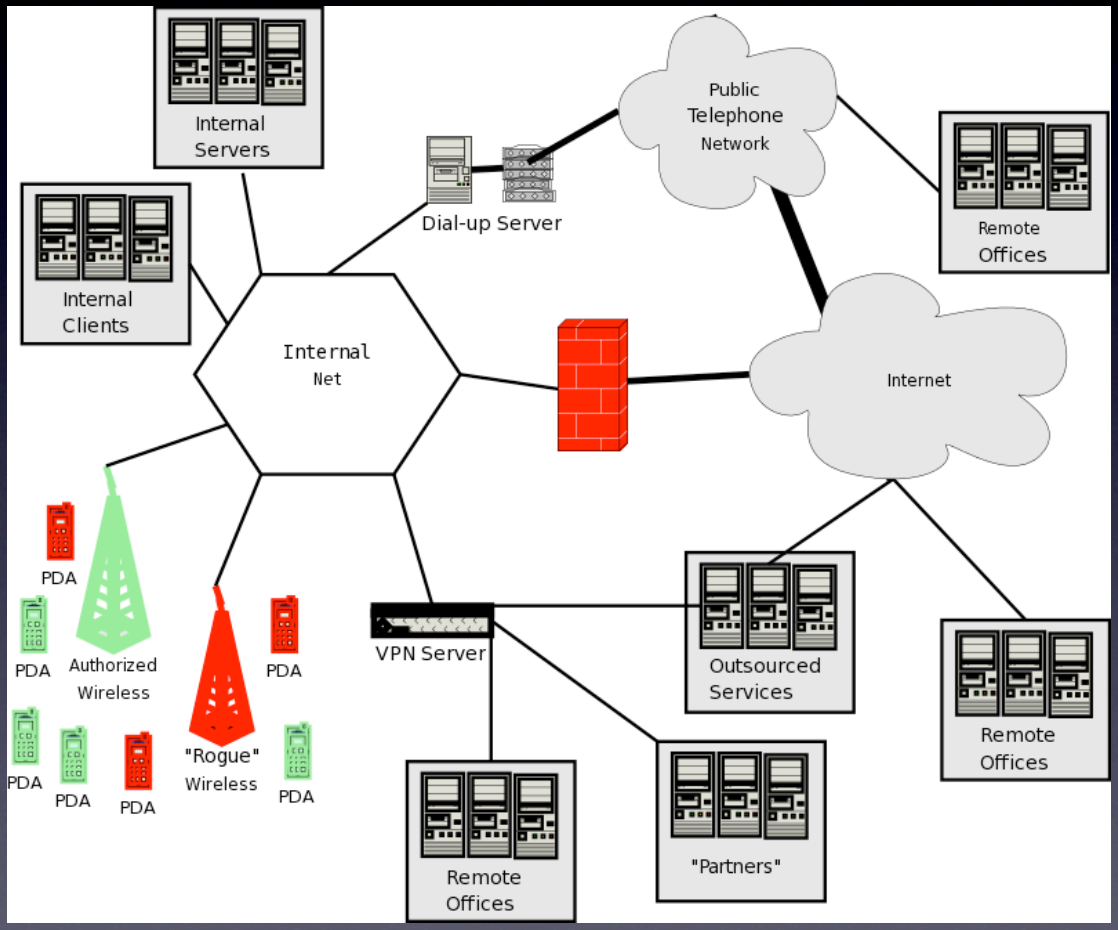
Non-signature based worm detection and isolation

Josh Ryder
Computer Security Administrator
University of Alberta

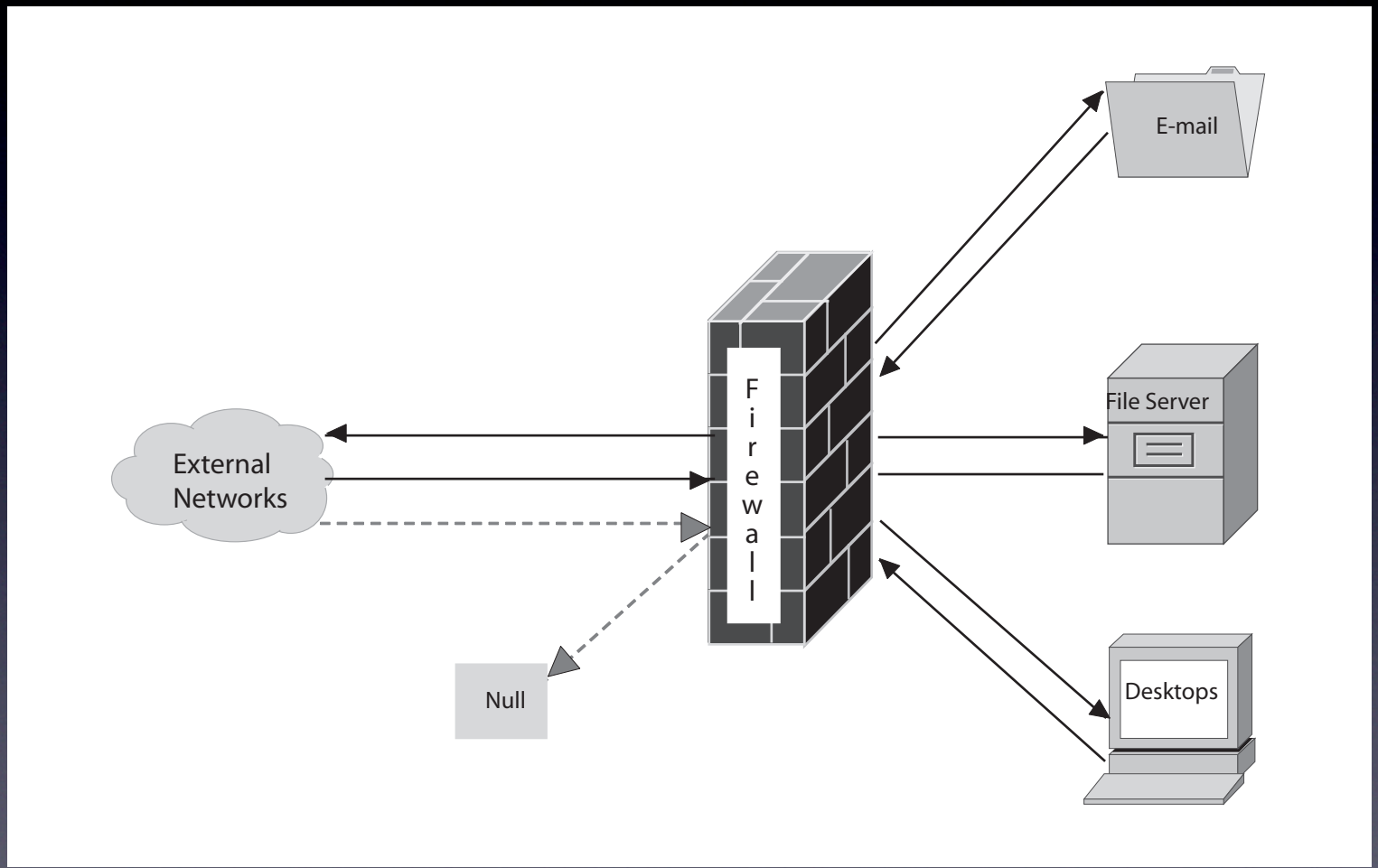
What we're covering today

- The corporate network
- Warhol worms
- Test environment
- Honeywall mechanism
- Detection and isolation results
- Conclusion and Future Research

The Corporate Network



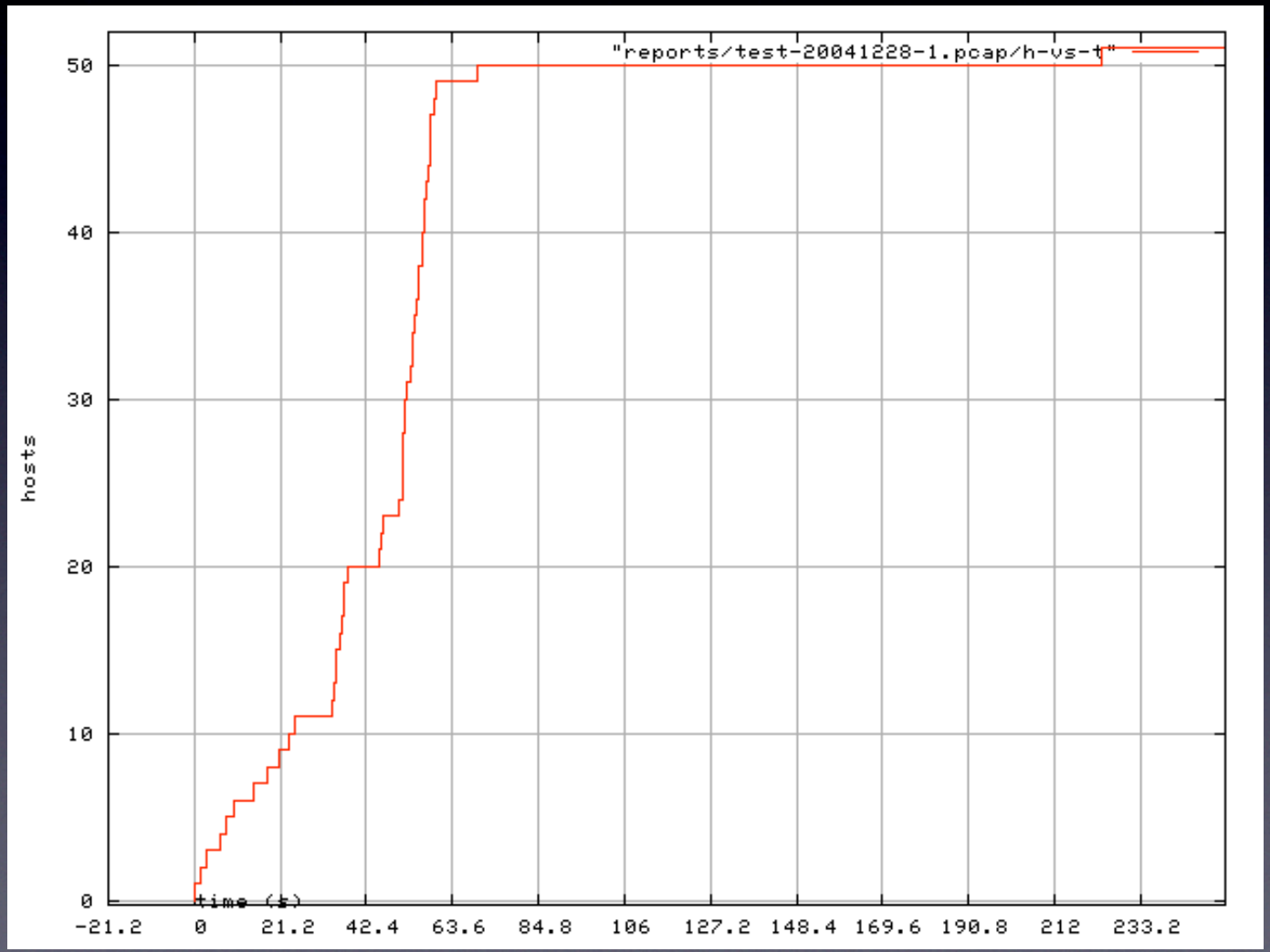
The Corporate Network



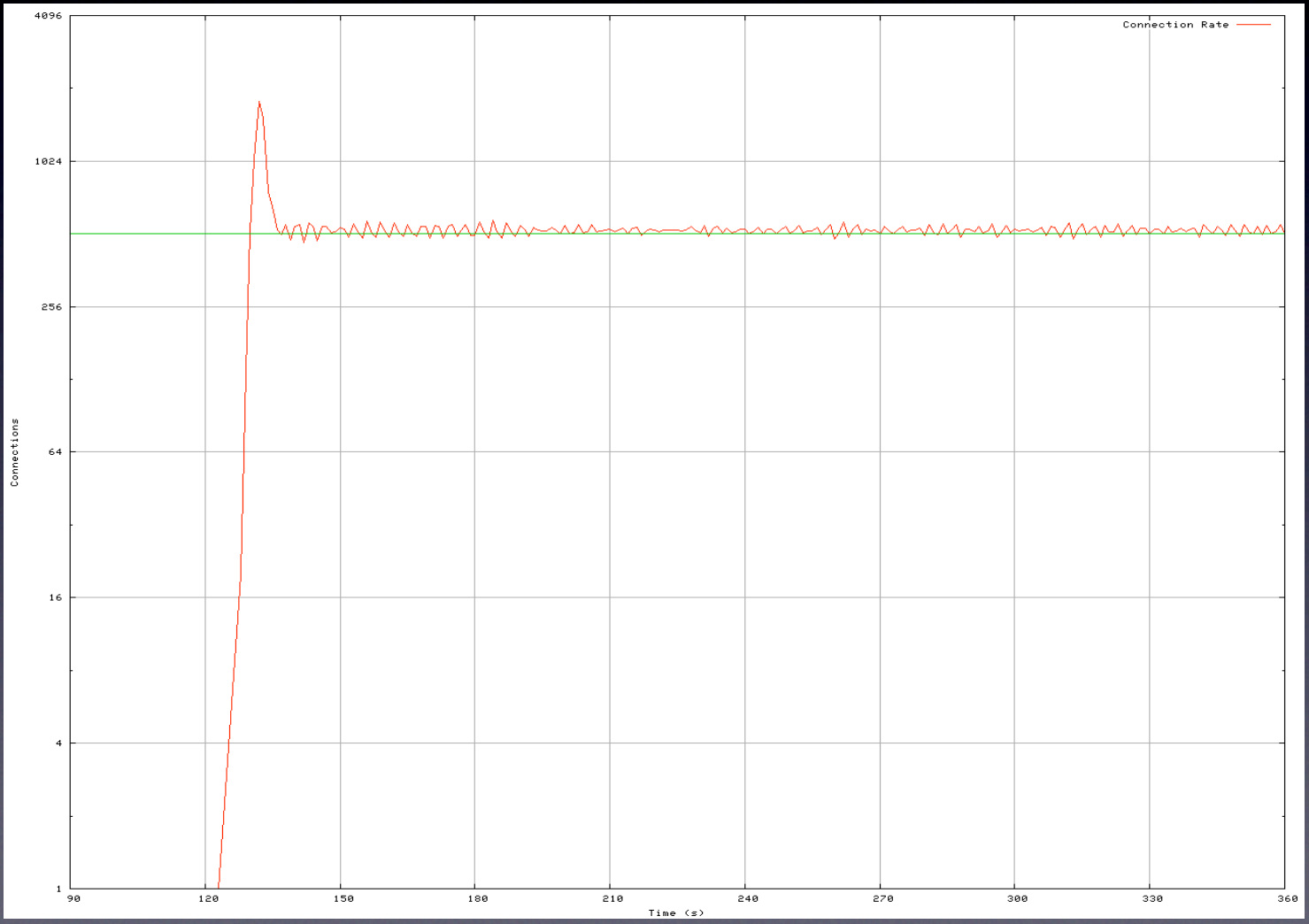
Worms

- Capable of spreading themselves without user intervention
- Multi-vector: Targets multiple vulnerabilities
- Spread rates can be very high, the fastest are known as Warhol worms

Warhol and your Network



Warhol and your Network



Our Warhol worm

- Models the spread of MS.Blast
- On a Class C network, the pseudorandomness of a worm does not adversely affect detection results so a linear scan was used.

Feature	Custom Worm	MS.Blast
Target Port	5678	135, Listens 4444, UDP 69
Targets of Worm	Vulnerable host process	DCOM RPC (vulnerable dll)
Probability of Infecting on LAN machine	40%	40% **
Probability of Infecting off LAN machine	60%	60% **
Scanning Threads	20	20
Payload (bytes)	6197	6176

** Note that MS.Blast will send a Windows XP exploit 80% of the time and Windows 2000 20%

Test Environment

- 50 identical machines
- Each system had the same vulnerable host process on it.
- Aggregated through VPN
- 100Mbit connections to aggregator

Worm Detection and Isolation

1. Worm enters network
2. Sensor reports worm traffic to collector
3. Collector analyzes reports
4. Collector signals Reactor
5. Reactor takes appropriate action

Baseline test cases
Conventional mechanisms

- Firewall
- Useful in protecting against known threats on specific ports
- Fails when worms uses permitted ports

Baseline Test Cases Conventional Mechanisms

- pf – connection rate limiting
- Threshold model used
- During normal usage a desktop computer uses 25-35 states
- We exploit the "known" behaviour of the average desktop to choose activity thresholds

Baseline Test Cases
Snort

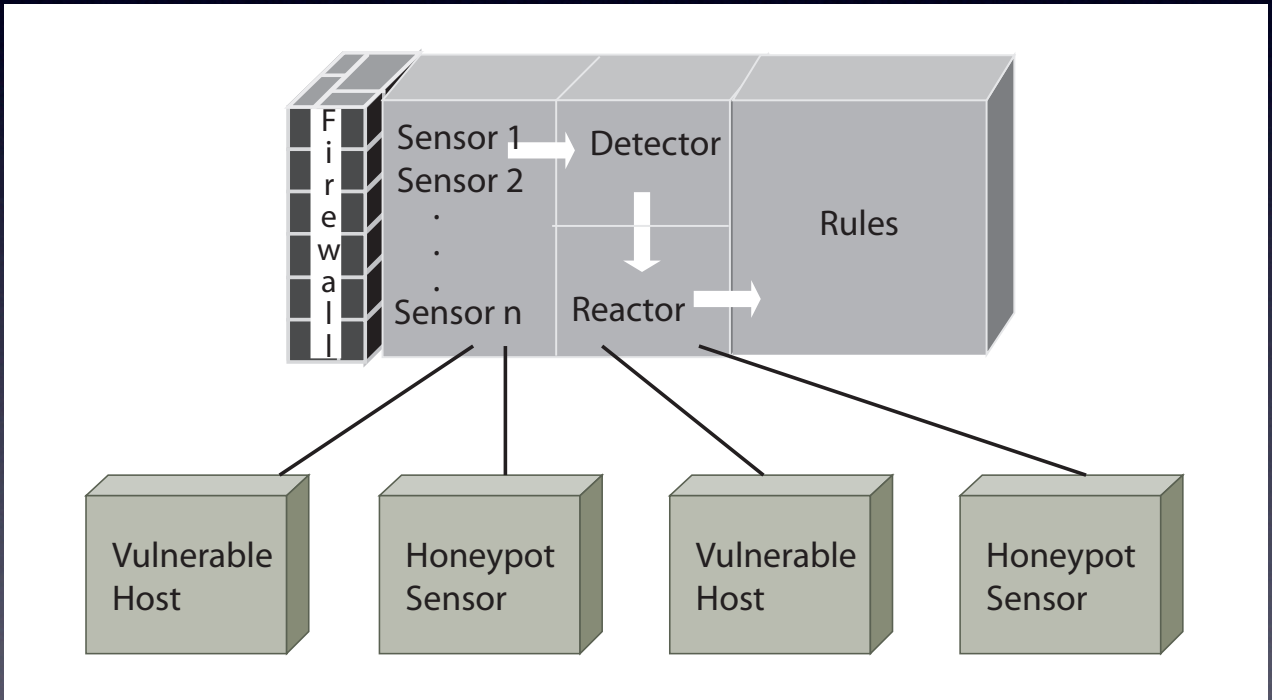
- Snort – signatures
- Snort without a signature doesn't detect the worm traffic
- Signature matching may provide lower detection latency

Honeypots

- Created as a research tool to investigate how systems are compromised
- Provides illusion of real hosts/services
- Exists so that its connection activity can be analyzed
- Any traffic to the honeypot is highly suspect

Honeywall

Honeypot + firewall == Honeywall



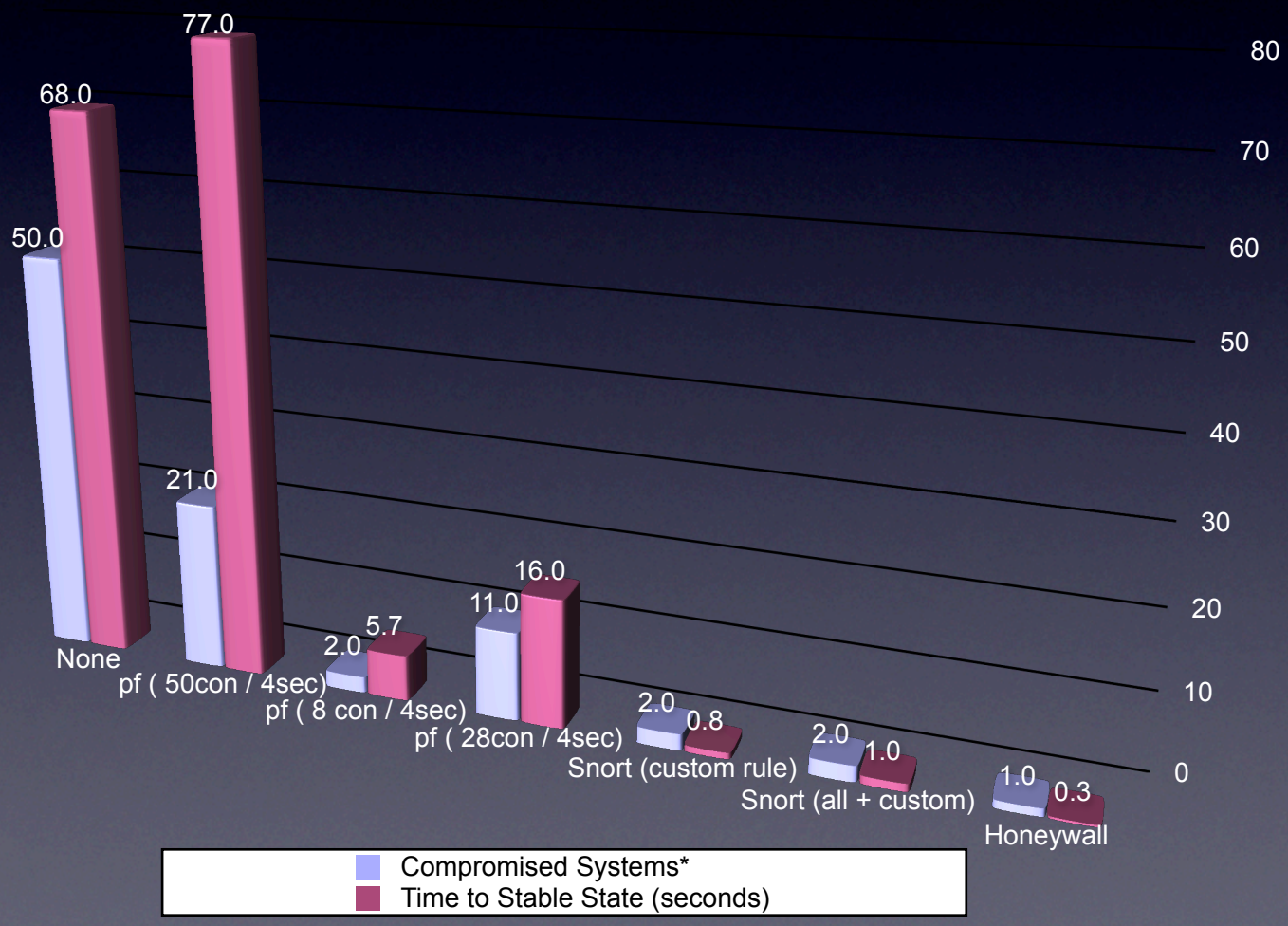
Honeywall

- Test environment has 50 real computers each running a vulnerable host process.
- Each of the 50 computers are sparsely distributed across the network
- The space between the real computers is populated with ultra-low interaction honeypot sensors.

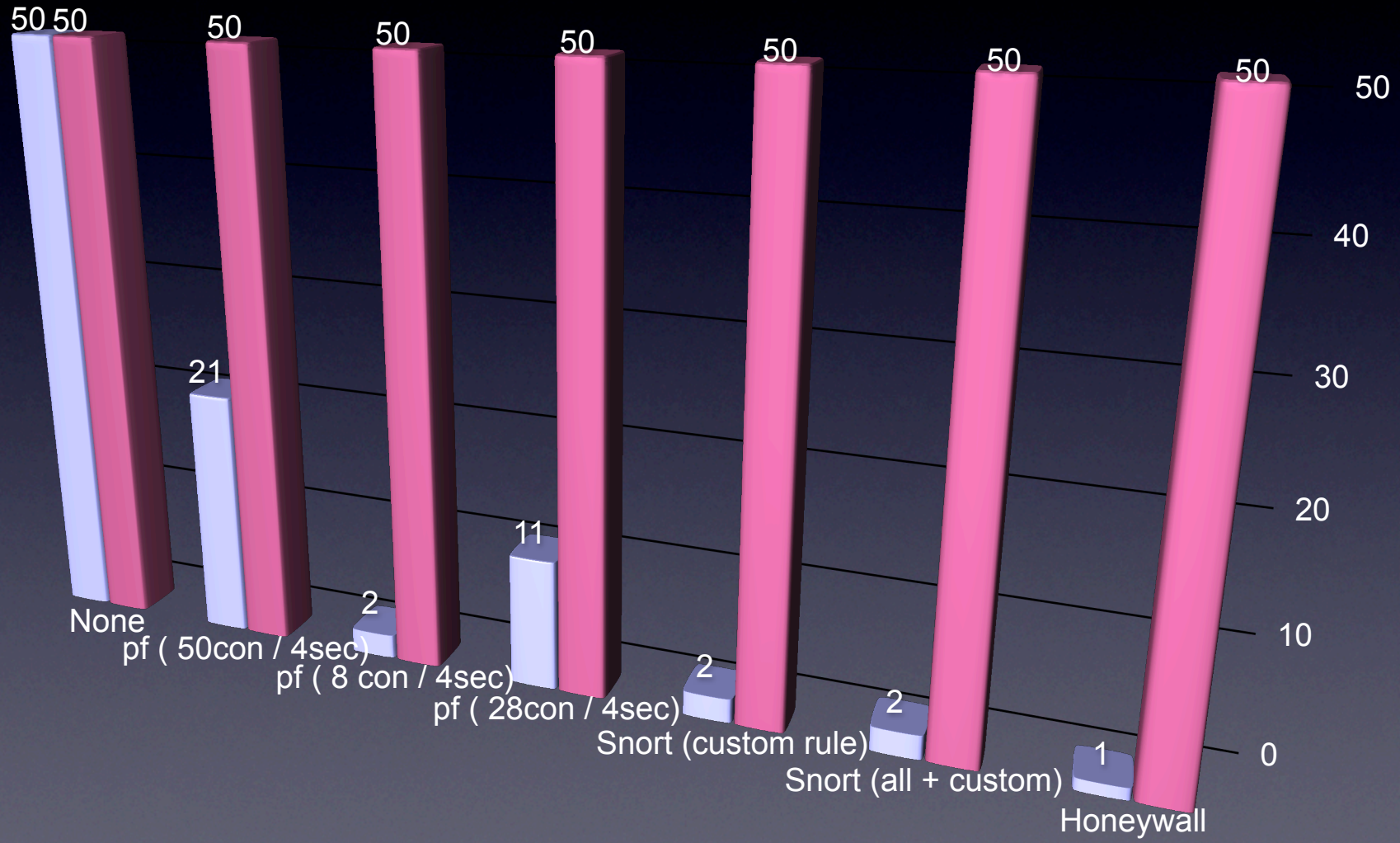
Results

Protection Type	Vulnerable Systems	Compromised Systems*	Time to Stable State (seconds)	Percentage Compromised
None	50	50	68	100%
Per subnet	27	27	27	100%
pf (50con / 4sec)	50	21	77	42%
pf (8 con / 4sec)	50	2	5.7	4%
pf (28con / 4sec)	50	11	16	22%
Snort (custom rule)	50	2	0.78	4%
Snort (all + custom)	50	2	0.99	4%
Honeywall	50	1	0.27	2%

Compromised Systems vs Time to Stable State



Infected vs Vulnerable Hosts



Future work

- Density and Distribution of Honeypot sensors
- Improvement of response times
- Hybrid approach

Where can this technology go?

- The honeywall technology is well suited to small LANs
- Ideally it is deployed on your network switch
- Could be deployed across multiple remote sites at aggregation points to prevent widespread infections within a distributed corporate LAN

Conclusion and Questions

- We have demonstrated that it is possible to use an ultra-low interaction honeywall to detect and isolate fast spreading worms
- Questions?