

On the Design and Analysis of Characteristics-Based Worm Detection Heuristics



Shad Stafford
staffors@cs.uoregon.edu

Zhen Wu
zwu@cs.uoregon.edu

Paul Knickerbocker
pknicker@cs.uoregon.edu

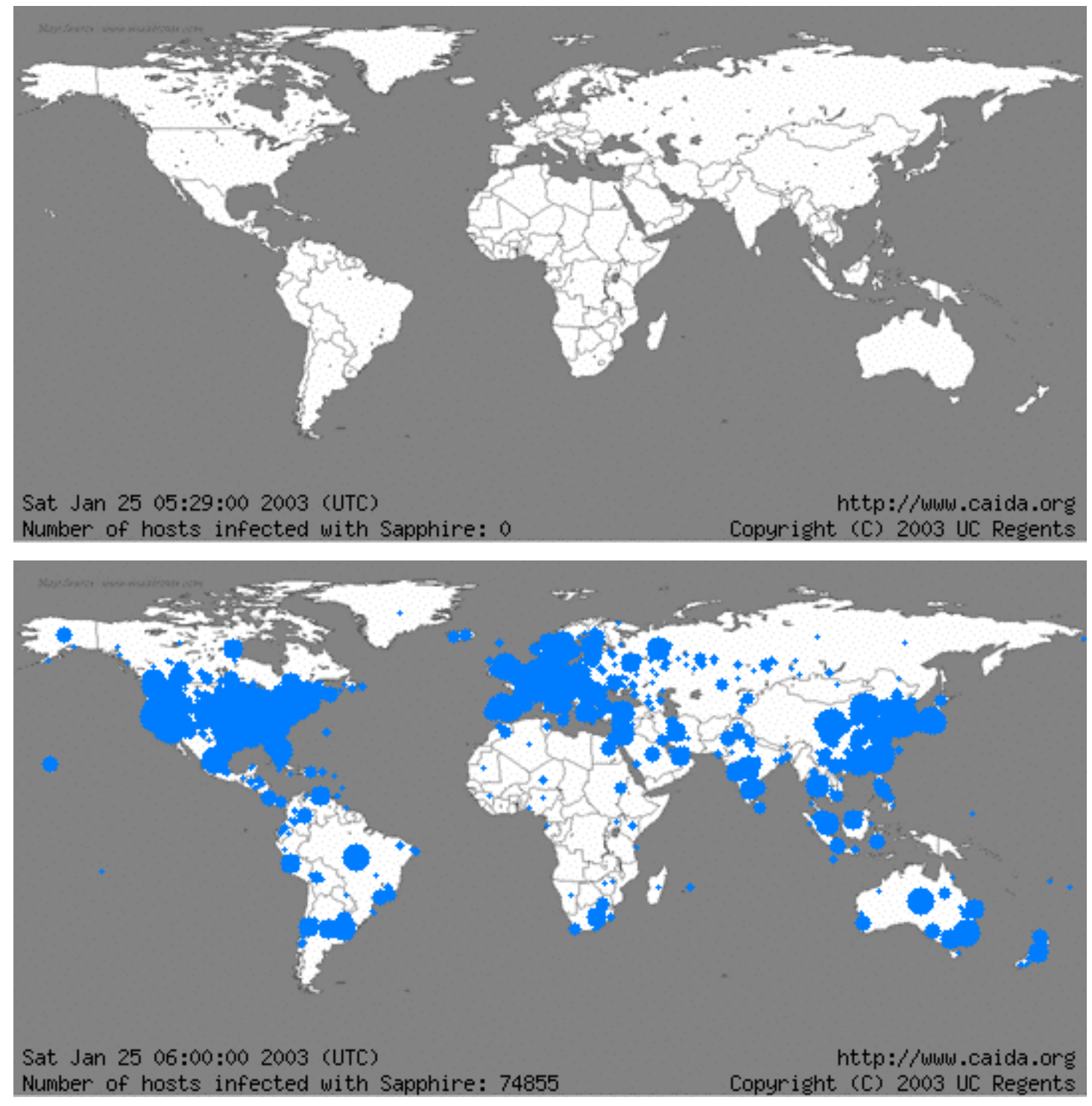
Toby Ehrenkranz
tehrenkr@cs.uoregon.edu

NetSec Lab - University of Oregon

Principle Investigator: Jun Li (lijun@cs.uoregon.edu)

The Worm Menace

- The Internet is now a critical infrastructure and is at risk of shutdown due to worm activity. The CodeRed and Sapphire/Slammer worms are estimated to have cost \$3 billion dollars in damages and lost productivity.
- Sapphire/Slammer achieved significant penetration in less than 30 minutes (see figures at right), but current worm countermeasures require the manual creation of byte-stream signatures, a process that can take hours or days.
- To counter this threat, a fast, automated worm defense system is required.
- We present our worm detection heuristics which automatically detect current and future worms. These heuristics will form the basis for an automated response system.



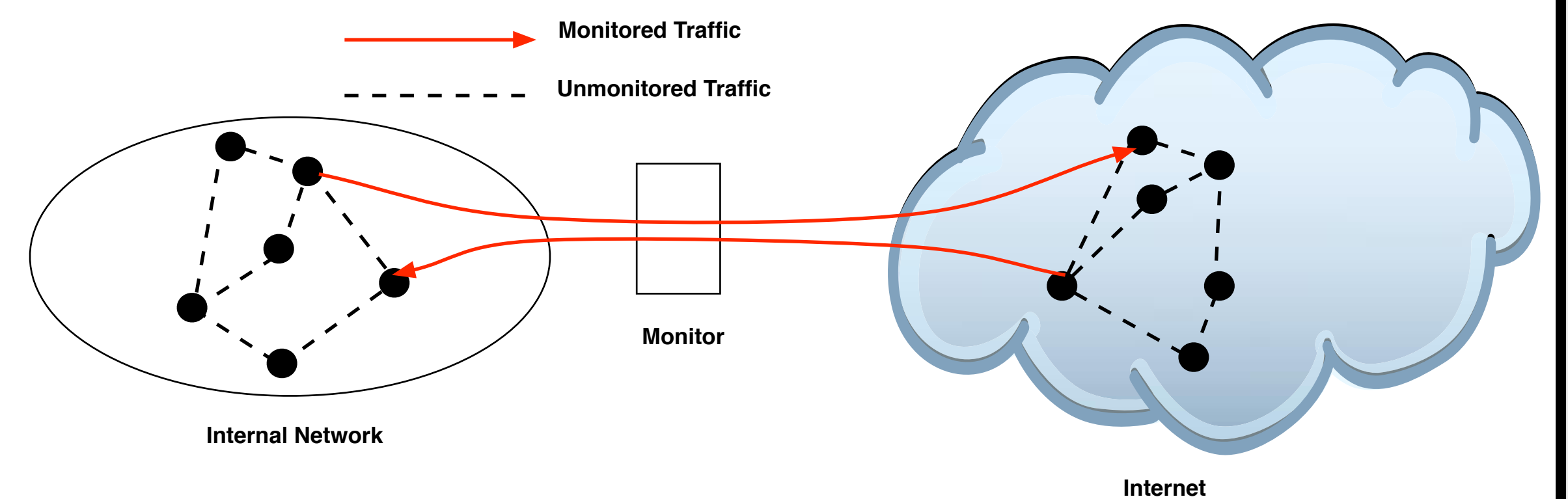
How to Catch a Worm

Rather than identifying worms based on byte-stream signatures, we focus on some intrinsic behaviors of worms:

- Self-replication/Self-similarity:** Worm traffic can be tracked as chains of infections as the worm propagates from host to host. This traffic tends to be self-similar, while normal traffic does not.
- Target Selection:** In order for a worm to spread it must select target addresses to infect, it often chooses addresses with no host, or hosts that don't provide the targeted service.
- Connection patterns:** Worms attempt to connect to a large number of different hosts in a pattern that is quite different than what is seen in normal traffic.

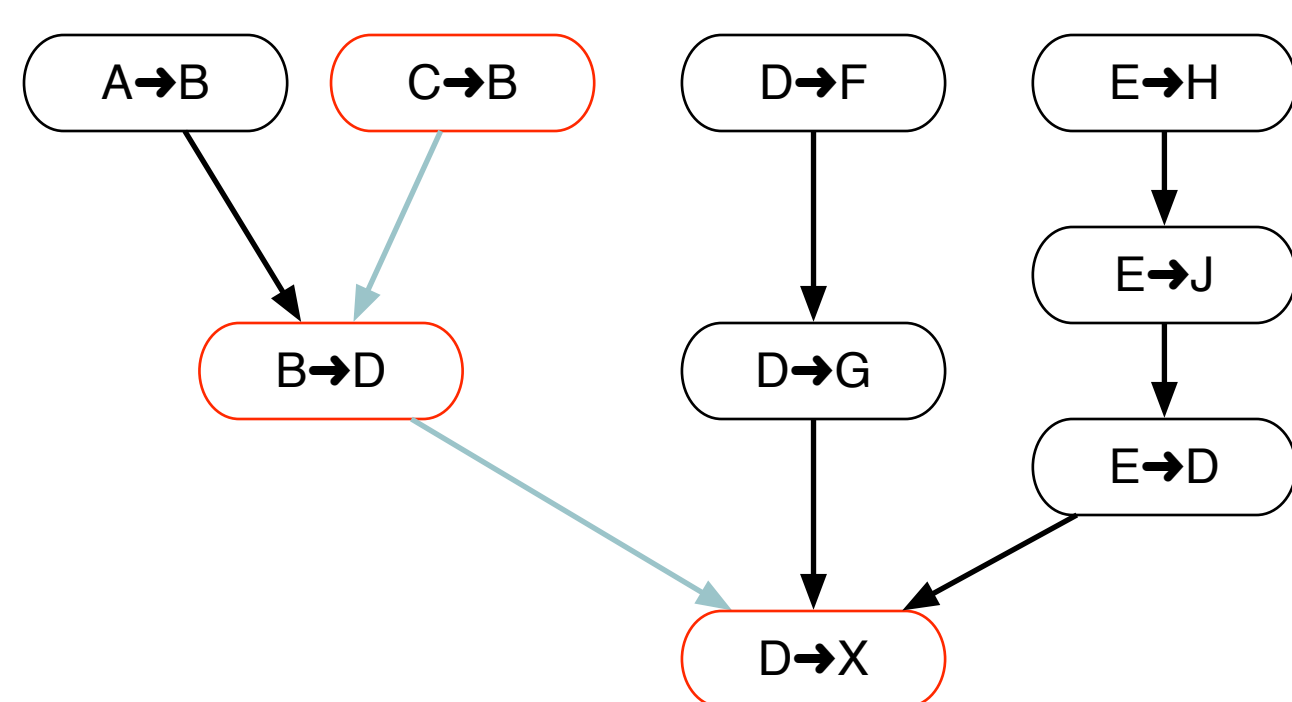
Our Worm Catcher

- It is prohibitively expensive to monitor all connections on a network. Instead we place a monitor at the gateway to the network and watch traffic there.
- To further simplify the data, we coalesce packets into ConnectDescriptors and do our analysis at the connection level. This avoids pitfalls with polymorphic or encrypted payloads.
- Our monitor applies three heuristics to each connection and uses the majority opinion to determine if connection is from an infected host (see below for heuristic descriptions).



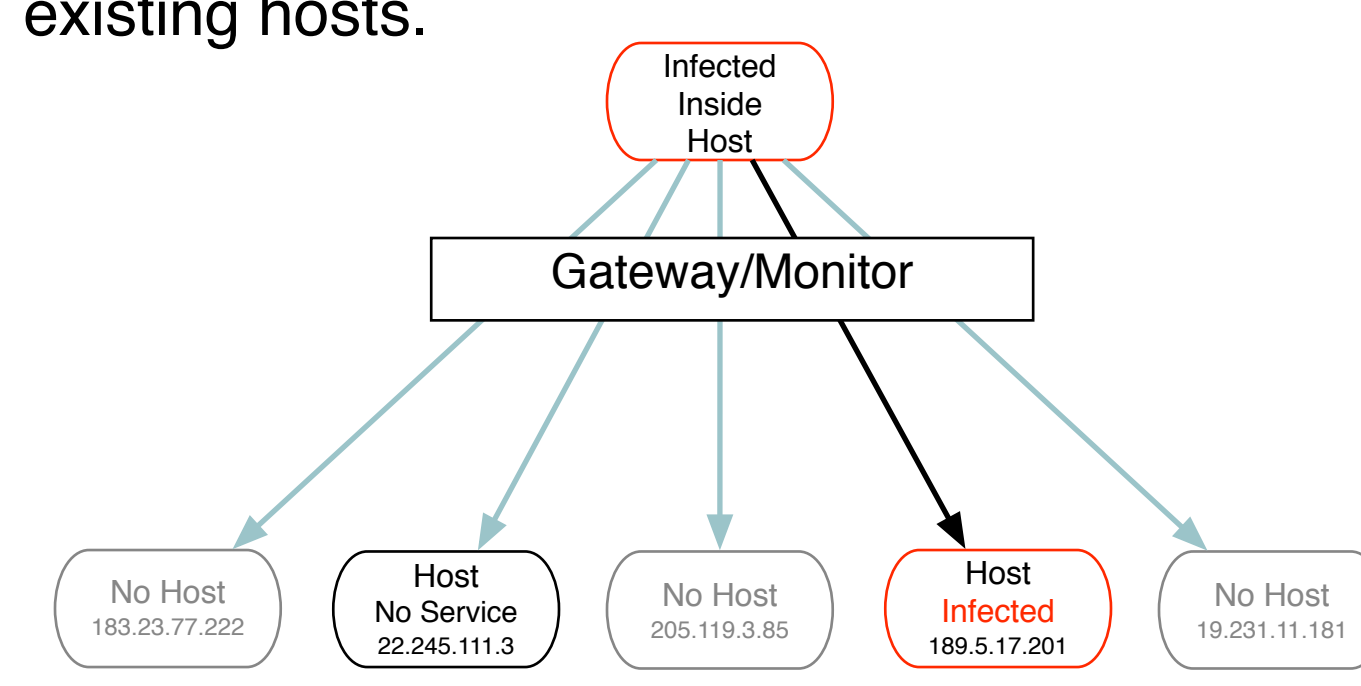
Causal Similarity Heuristic

- Adds each connection to a *causal connection graph* representing all the possible causes of a connection using Lamport's *happened-before* relationship.
- For each new connection, make comparisons to its ancestors in the causal connection graph. If enough of these are similar, this connection may be a worm infection.
- Below is an example connection graph used to analyze the D→X connection. It shows that host C infected host B, who then infected host D.



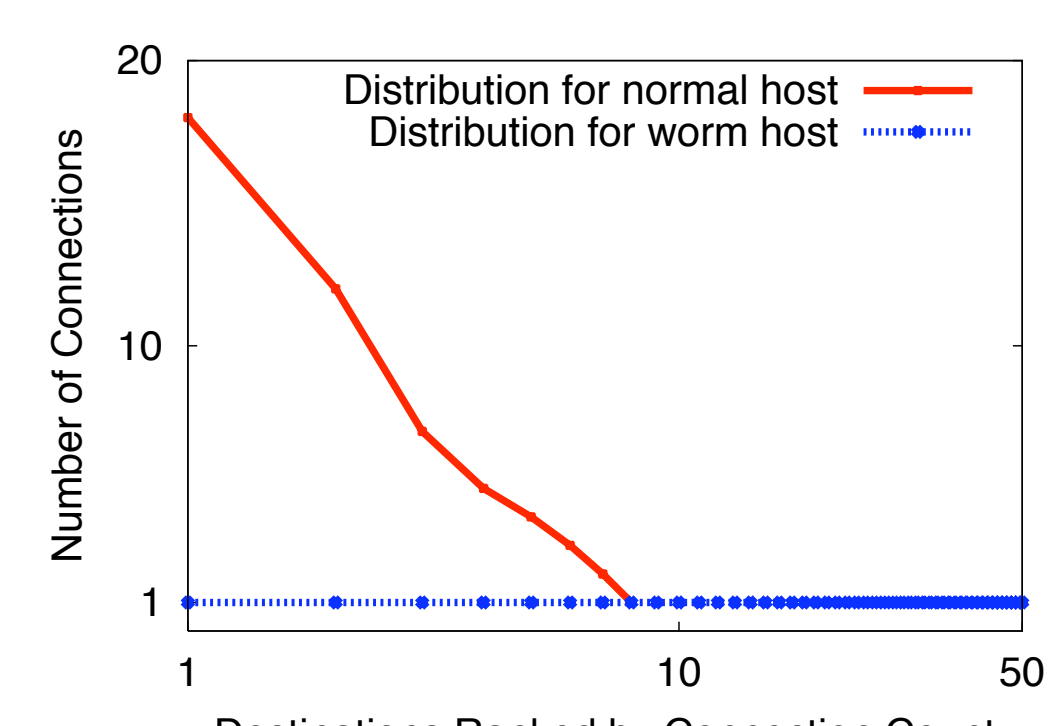
Non-Existent Service Heuristic

- Infected hosts make an abnormally large number of connections to addresses not associated with a host and to ports where no service is available.
- This heuristic tracks the number of connections to non-existent hosts and services, and when that number exceeds a given threshold, the host is considered to be infected.
- The diagram below illustrates the connection pattern of an infected host. The blue arrows represent connections to non-existent hosts or services and far outnumber the connections to existing hosts.



Address Distribution Heuristic

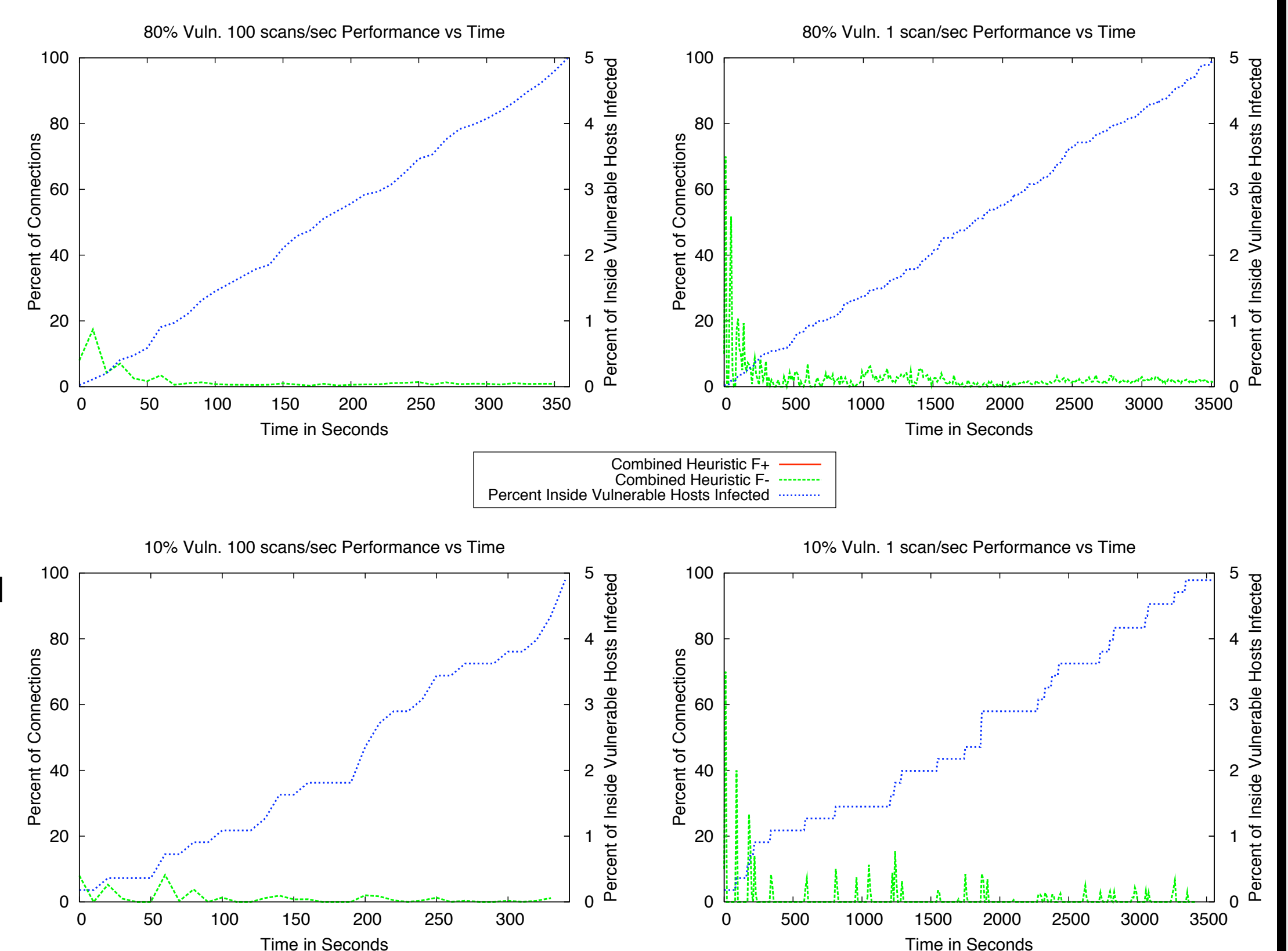
- The normal connection pattern for a host is to repeatedly connect to a limited set of addresses. In fact, the connection history maps to a power-law distribution. Worms on the other hand, typically make single connections to a large set of target addresses.
- This heuristic examines the connection history of each host every time it makes a new connection. When the connection history no longer resembles a power-law distribution, the host is considered very likely to be infected.



Experimental Results

- To evaluate our heuristics, we ran them against a trace consisting of real traffic recorded at a gateway router at Auckland University combined with worm traffic simulated on the Auckland network topology.
- The worm in our simulation was modeled as a *random scanning* worm. We ran multiple simulations with varying vulnerability levels and worm scan rates to give broad coverage of the threat spectrum.
- Connections scored as "worm" originating from the legitimate traffic of the Auckland recordings are considered false positives (F+). Connections scored as "non-worm" originating from the simulated worm traffic are failures to detect and are reported as false negatives (F-).
- Once a host is infected, some of its normal connections may be scored as "worm" because they are intermingled with outgoing worm connections. False positives of this nature (those taking place after we detect "worm" connections from a given host) are considered acceptable so a corrected measure of false positives is presented as the Adjusted F+, which disregards connections of this sort. This can be considered a truer measure of the accuracy of the heuristics.

Per Connection Accuracy until 5% of inside hosts are infected				
Vulnerability	Scan Rate	F-	F+	Adj F+
80% Vulnerable	100 scans/sec	0.851%	0.001%	0%
80% Vulnerable	1 scan/sec	1.718%	<0.001%	<0.001%
10% Vulnerable	100 scans/sec	0.508%	0%	0%
10% Vulnerable	1 scan/sec	1.712%	4.419%	0%



Note: the false positives are plotted on the graphs but are so low as to not be visible.