Worm Detection

Ankur Agiwal

Worm Detection

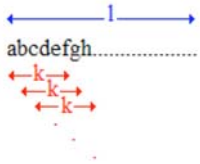
- Packet Content Matching
- Port Number Matching
- ICMP Packet Analysis

Packet Content Matching

- Which characteristic of worm is exploited?
  - Highly repetitive packet content
  - Increasing population of destinations being targeted
  - Increasing population of sources generating infections

Packet Content Matching

- Should whole packet content be a signature?
  - Check all possible substrings of a certain length
- How to make this substring-check fast?



O(l.k)

- Solution: Rabin fingerprints

Rabin Fingerprints

- Definition: Rabin fingerprint F_1 for a sequence of bytes $t_1,t_2,\ldots,t_k$ is:

  $(t_1.p^{k-1} + t_2.p^{k-2} + \ldots + t_k) \bmod M$

  k: length of substring $[t_1t_2\ldots t_k]$

  p, M: constants
- Property: Two equal substrings generate same Rabin fingerprint
- Not a perfect signature!

Rabin Fingerprints

- Compute Rabin fingerprints for all possible substrings
- Still O(l.k) ?
- No, computation can be done incrementally.
- Rabin fingerprint F_2 for a sequence of bytes $t_2,t_3,\ldots,t_{k+1}$ can be computed as:

  $(F_1.p + t_{k+1} - t_1.p^k) \bmod M$
- For efficient computation, pre-compute a table of all values of $t_i.p^k$

Signature Generation

- Compute a set of signatures for every packet payload
- Count number of distinct sources, distinct destinations, and distinct source-destination pairs
  - Counters are instantiated only for fingerprints with frequency greater than a threshold, *occuranceRate*.

Alerts

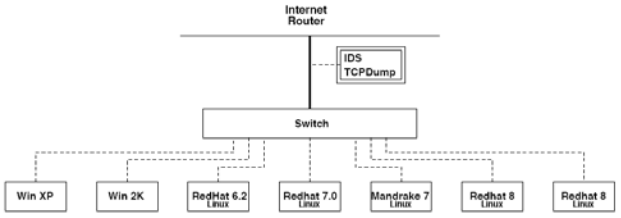
- As each packet generates multiple signatures, calculate *matchPct* (percentage of matching signatures)
- When *matchPct* and counters for number of hosts are above some threshold, generates an alert

Alerts (contd)

- As a general rule, the system alerts when:
  - Packets with similar contents are being sent to a number of hosts
  - Packets with similar contents are being sent from a large number of hosts
  - Packets with similar content are being sent from a number of hosts to a large number of hosts

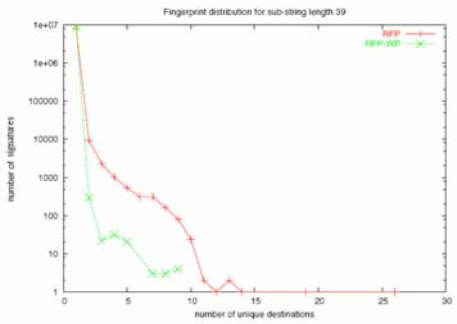
Evaluation

- A LAN of 7 hosts
- tcpdump trace of 9 days
- 4 million packets



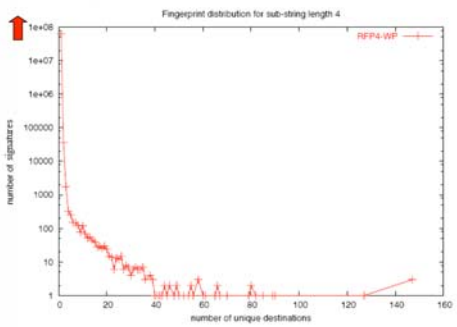
Fingerprint Distribution for k=39

- Each point represents total number of signatures destined for a number of destinations



Fingerprint Distribution for k=4

- Order of magnitude increase in number of signatures (more resources needed)



Results

- Packets marked as containing worm traffic

Source	Dest	SD Pairs	Prot/Port	Exploit (truncated)	Name/Incident
45	5	51	TCP/80	GET /default.ida?XXXXXXX	CodeRed variant
4	3	4	TCP/80	GET / HTTP /1.1	Slapper
1	4	4	TCP/80	GET /scripts/.%252e/.%252e/	Unicode exploit
1	4	4	TCP/80	GET /scripts/..%c0%af../	Unicode exploit
1	4	4	TCP/80	GET /scripts/..%255c%255c../	Unicode exploit
1	9	9	TCP/443	-	Slapper
1	3	3	TCP/80	GET http://www.s3.com HTTP/1.1	Not a Worm
498	4	742	TCP/139	-	Out of band attack
17	3	23	TCP/445	-	Out of band attack

False Positives

- Same piece of content is sent from one host to many different hosts (mailing list, http server)
- Same request is sent from many different clients to one server
- Solution: At least k distinct sources and at least k distinct destinations should be involved
- Not eliminated:
  - Request for objects like “robot.txt”
  - Single packet application identifier strings, eg. “SSH-1.99-3.11 SSH Secure Shell for Windows”

Worm Detection

- Packet Content Matching
- **Port Number Matching**
- ICMP Packet Analysis

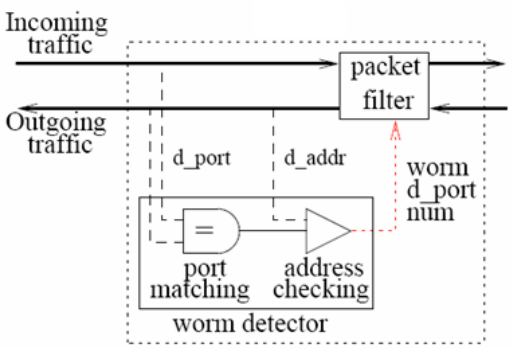
Motivation

- A worm exploits a security vulnerability corresponding to a specific network port number

Monitoring

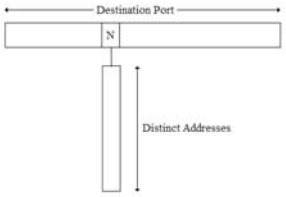
- Why not monitor source and destination addresses?
- How to count packets with same destination port number?

Worm Detection



Worm Detection (contd)

- How to find prominent ports?
  - Maintain a list of the number of connections to different destination ports
- Timer for each list entry



Worm Detection (contd)

- When to alarm?
  - For every T second interval, check the number of connections. (Detection Interval)
- What to compare?

  if $N > N_{avg}.(1+\delta)$ **worm traffic !**

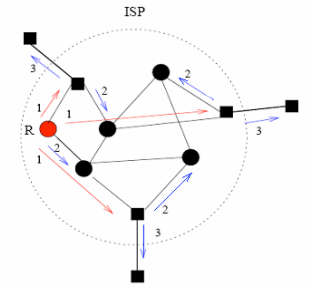
  N: number of unique addresses

  N_{avg}: long term average

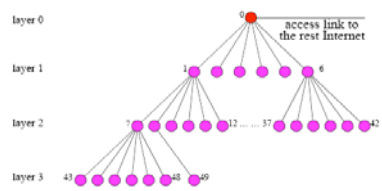
  $$N_{avg} = \alpha.N_{avg} + (1-\alpha).N$$

Packet Filtering

- Routers drop packets with automatically discovered suspicious destination ports



Simulation



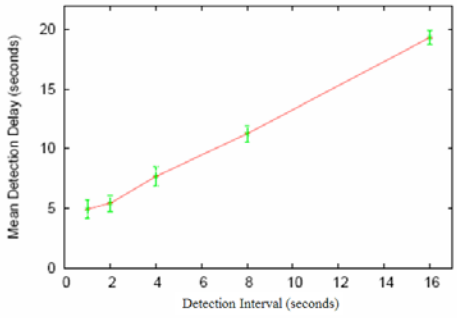
- ns (network simulator)
- A topology of 6-nary tree with total 50 routers
- All connections have 100 Mbps bandwidth and 50ms propagation delay
- Each router connects 50 hosts with 100 Mbps links and 25ms propagation delay

Simulation (contd)

- Worm traffic was generated using an epidemic worm propagation model
- Randomly 30% of hosts were made vulnerable
- With full deployment and 1 second detection interval, worm traffic detected in just **3.87** seconds

Simulation (contd)

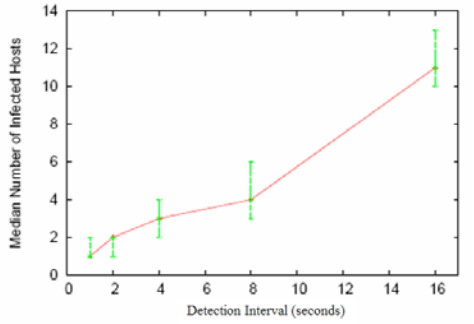
- Effect of detection interval on detection delay



increasing

Simulation (contd)

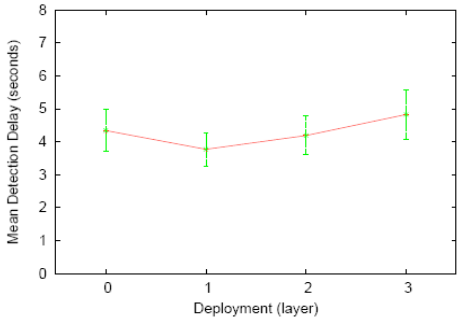
- Effect of detection interval on #infected hosts



increasing

Simulation (contd)

- Effect of deployment on detection delay



insensitive

Simulation (contd)

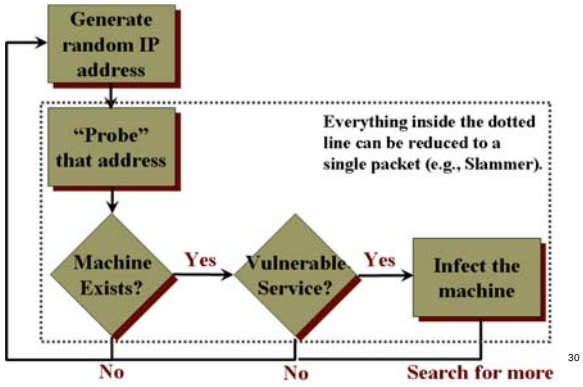
- Effect of sensitivity, δ (tradeoff between detection delay and false alarms)
  - δ=1, no false alarm
  - δ=0.5, false alarm for Web and DNS
  - δ=0.25, false alarm for FTP

- Summary
- Detects at early stage to suppress worm before it gets out of control
  - Signature-based IDS (time-consuming)
  - Anomaly-based IDS (high false alarm rate)
- Low speed worms?

Worm Detection

- Packet Content Matching
- Port Number Matching
- **ICMP Packet Analysis**

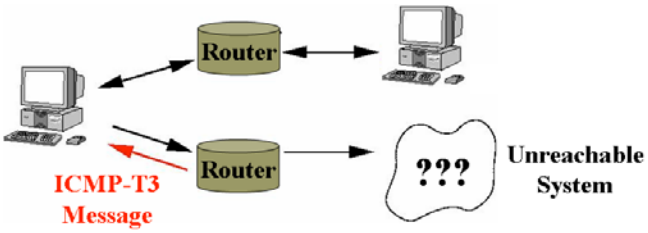
How do most worms work?



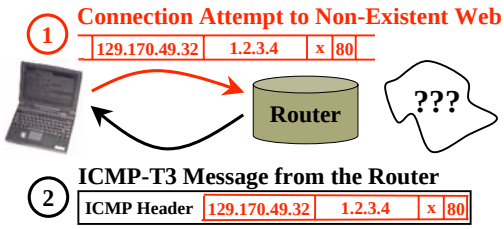
Motivation

- Due to random scanning behavior of worms, many vacant IP addresses are probed
- What happens if a vacant IP address is probed?
  - ICMP unreachable message

ICMP Destination Unreachable



Embedded Content



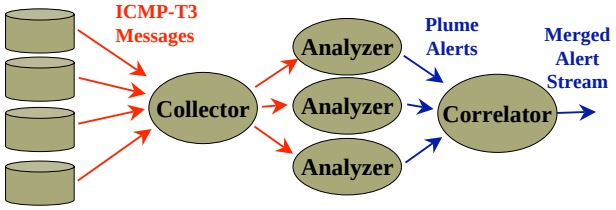
So we know…

- The machine which made the attempt (129.170.249.32)
- What it was trying to contact (Port 80)

Worm Detection

- How to make use of these ICMP packets?
  - Routers generate duplicate ICMP destination unreachable messages and forward them to a central collector

Scalability



- Collector divides entire IP address space among a number of analyzers
- Collector sends two copies of ICMP packets

Analyzers

- Look for the case when
  - one IP address has contacted at least N different other IP addresses on the same port p using the same protocol P in the last δt seconds

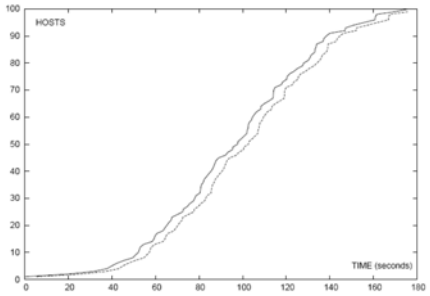
  OR
  - one IP address was contacted by at least N different other IP addresses on the same port p using the same protocol P in the last δt seconds

Correlator

- Compare all alerts received in previous δt time and identify similarities
- Report sent to the user
  - List of IP addresses
  - Scanning behavior
  - Protocol
  - Port number
  - Timestamps

Simulation

- Assumed epidemic worm propagation model
  - Solid line: Total instances of worm
  - Dotted line: Total worms detected



- Summary
- Router updates

Thank You!

References

- The EarlyBird System for Real-time Detection of Unknown Worms, Sumeet Singh, Cristian Estan, George Varghese and Stefan Savage, University of California San Deigo, Department of Computer Science, Technical Report CS2003-0761, August 2003.
- Detecting Early Worm Propagation through Packet Matching, Xuan Chen and John Heidemann, USC Information Sciences Institute Technical Report, ISI-TR-2004-585, February 2004.
- Designing a Framework for Active Worm Detection on Global Networks, Vincent Berk, George Bakos, Robert Morris, First IEEE International Workshop on Information Assurance (IWIA'03), Darmstadt, Germany, March 2003, pages 13-24.