Autograph

Toward Automated, Distributed Worm Signature Detection

Hyang-Ah Kim

Carnegie Mellon University

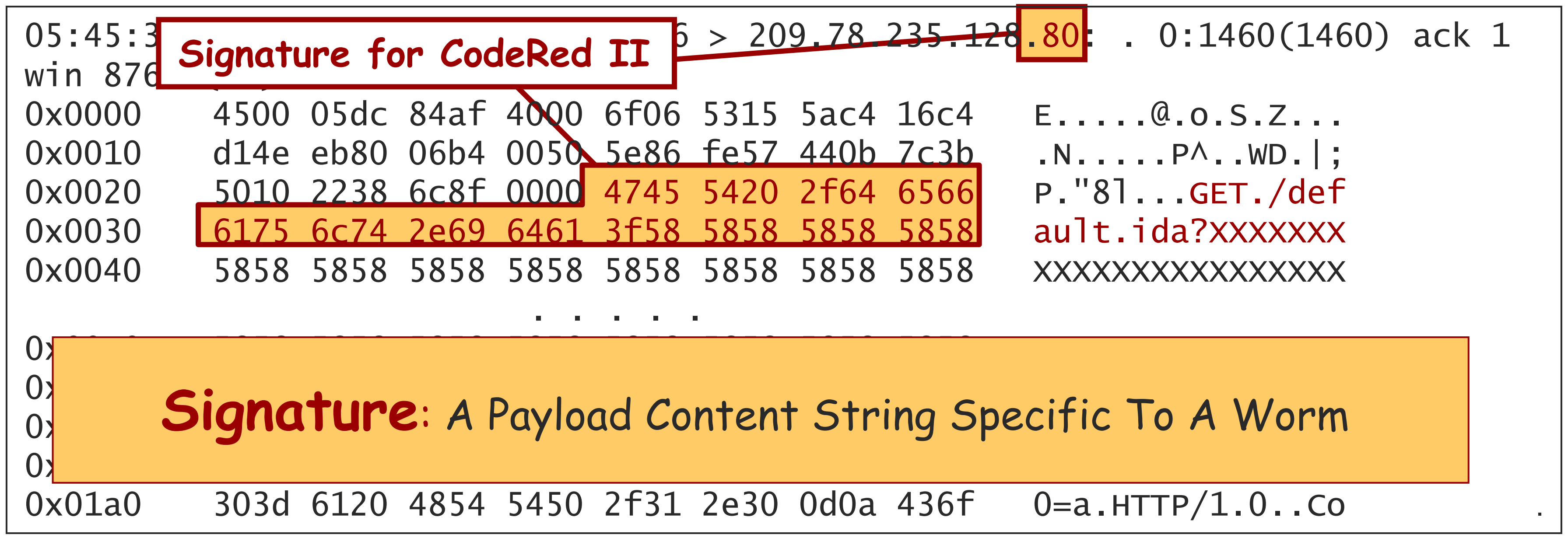
Brad Karp

Intel Research &
Carnegie Mellon University

Usenix Security 2004

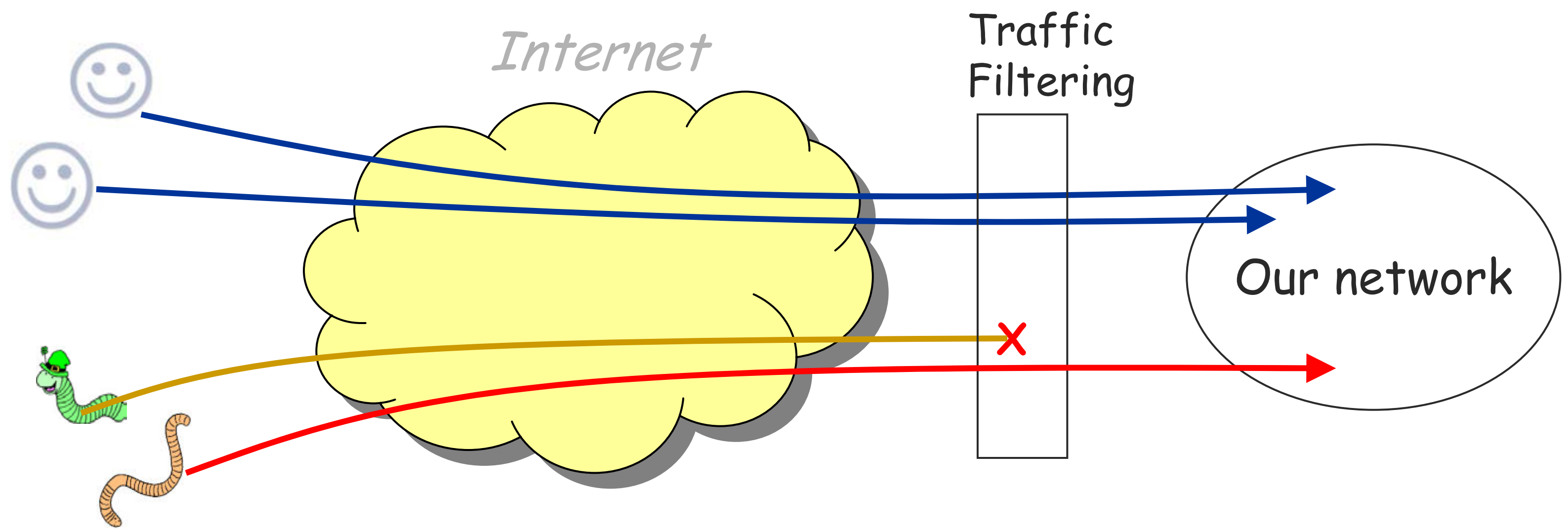
Internet Worm Quarantine

- Internet Worm Quarantine Techniques
  - Destination port blocking
  - Infected source host IP blocking
  - Content-based blocking [Moore et al., 2003]
- Worm Signature



Content-based Blocking

Signature for CodeRed II



- Can be used by Bro, Snort, Cisco's NBAR, ...

Signature derivation is too slow

- Current Signature Derivation Process
  - New worm outbreak
  - Report of anomalies from people via phone/email/newsgroup
  - Worm trace is captured
  - Manual analysis by security experts
  - Signature generation

⇒ Labor-intensive, Human-mediated

Goal

Automatically generate signatures of previously unknown Internet worms

- as accurately as possible
  - ⇒ Content-Based Analysis
- as quickly as possible
  - ⇒ Automation, Distributed Monitoring

Assumptions

- We focus on **TCP worms that propagate via scanning**

  Actually, any transport
  - in which spoofed sources cannot communicate successfully
  - in which transport framing is known to monitor

- Worm's payloads share a common substring
  - Vulnerability exploit part is not easily mutable
    - Not polymorphic

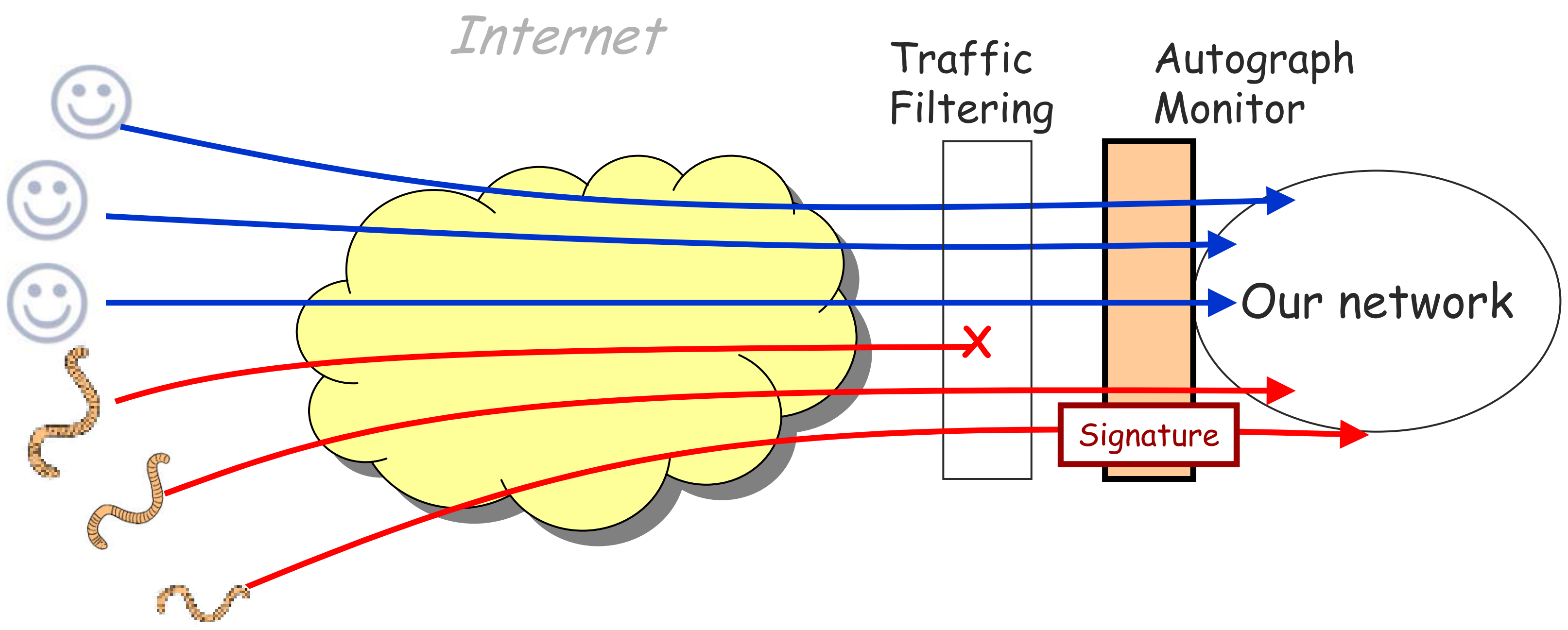
Outline

- Problem and Motivation
- Automated Signature Detection
  - Desiderata
  - Technique
  - Evaluation
- Distributed Signature Detection
  - Tattler
  - Evaluation
- Related Work
- Conclusion

Desiderata

- Automation: Minimal manual intervention
- Signature quality: Sensitive & specific
  - Sensitive: match all worms ⇒ low false negative rate
  - Specific: match only worms ⇒ low false positive rate
- Timeliness: Early detection
- Application neutrality
  - Broad applicability

Automated Signature Generation



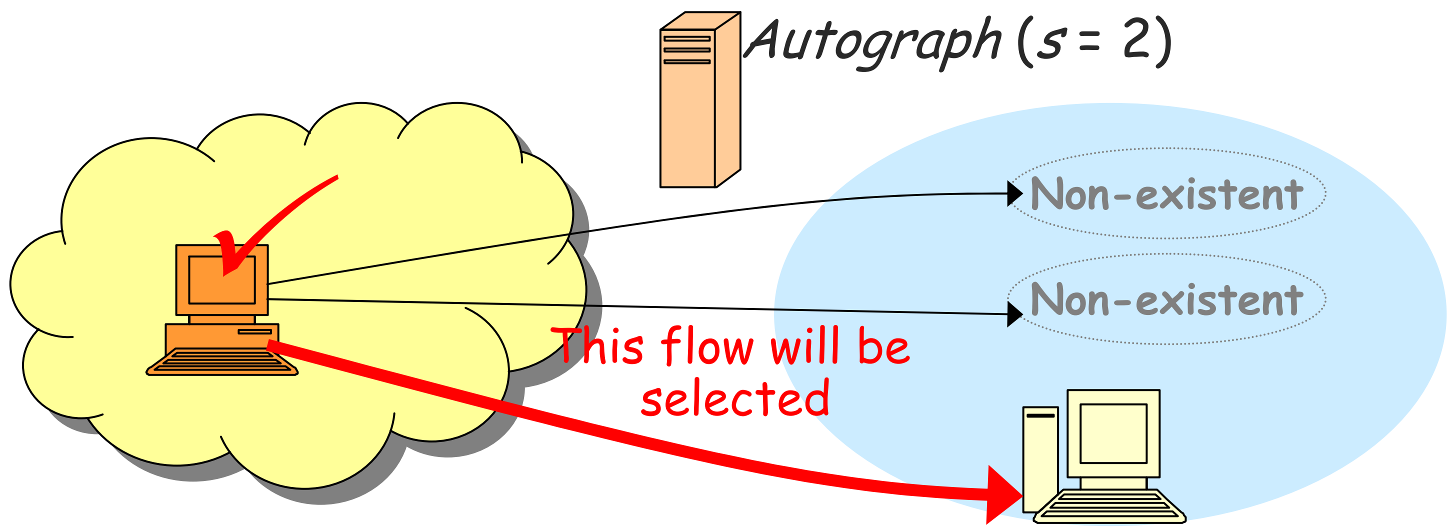
- Step 1: Select suspicious flows using heuristics
- Step 2: Generate signature using content-prevalence analysis

S1: Suspicious Flow Selection

Reduce the work by filtering out
vast amount of innocuous flows

- Heuristic: Flows from scanners are suspicious
  - Focus on the successful flows from IPs who made unsuccessful connections to more than *s* destinations for last 24hours

⇒ Suitable heuristic for TCP worm that scans network



S1: Suspicious Flow Selection

Reduce the work by filtering out
vast amount of innocuous flows

- Heuristic: Flows from scanners are suspicious
  - Focus on the successful flows from IPs who made unsuccessful connections to more than *s* destinations for last 24hours
  - ⇒ Suitable heuristic for TCP worm that scans network

- Suspicious Flow Pool
  - Holds reassembled, suspicious flows captured during the last time period *t*
  - Triggers signature generation if there are more than θ flows

S2: Signature Generation

Use the most frequent byte sequences across suspicious flows as signatures

All instances of a worm have a common byte pattern specific to the worm

Rationale

- Worms propagate by duplicating themselves
- Worms propagate using vulnerability of a service

How to find the most frequent byte sequences?

Worm-specific Pattern Detection

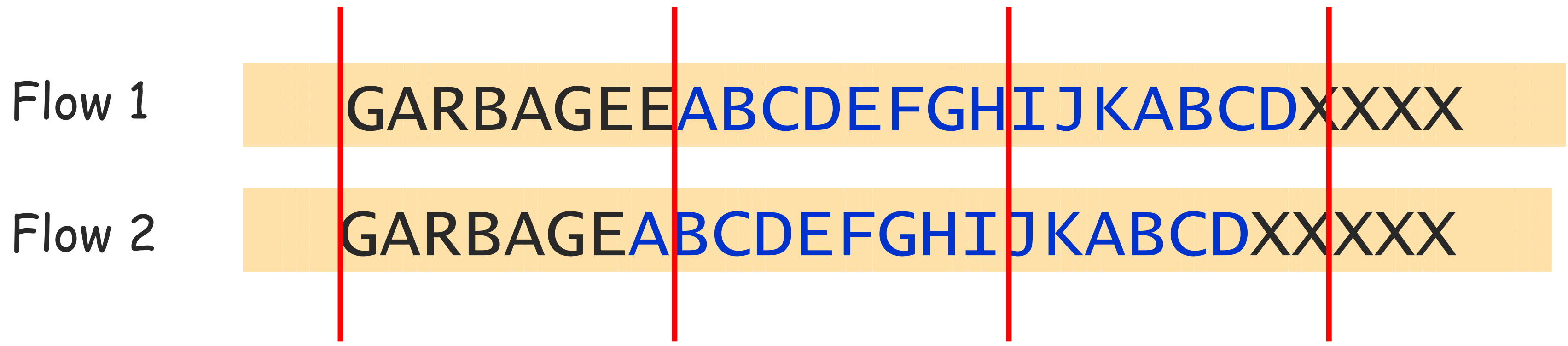
- Use the entire payload
  - Brittle to byte insertion, deletion, reordering

Flow 1	GARBAGEEABCDEFGHIJKABCDXXXX
Flow 2	GARBAGEABCDEFGHIJKABCDXXXXX

Worm-specific Pattern Detection

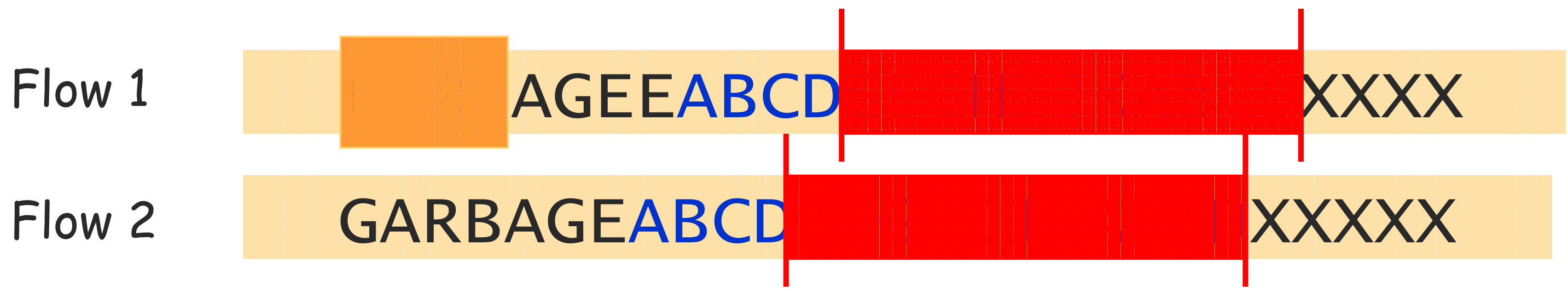
Partition flows into non-overlapping small blocks and count the number of occurrences

- Fixed-length Partition
  - Still brittle to byte insertion, deletion, reordering



Worm-specific Pattern Detection

- Content-based Payload Partitioning (COPP)
  - Partition if Rabin fingerprint of a sliding window matches Breakmark ⇒ Content Blocks
  - Configurable parameters: content block size (minimum, average, maximum), breakmark, sliding window

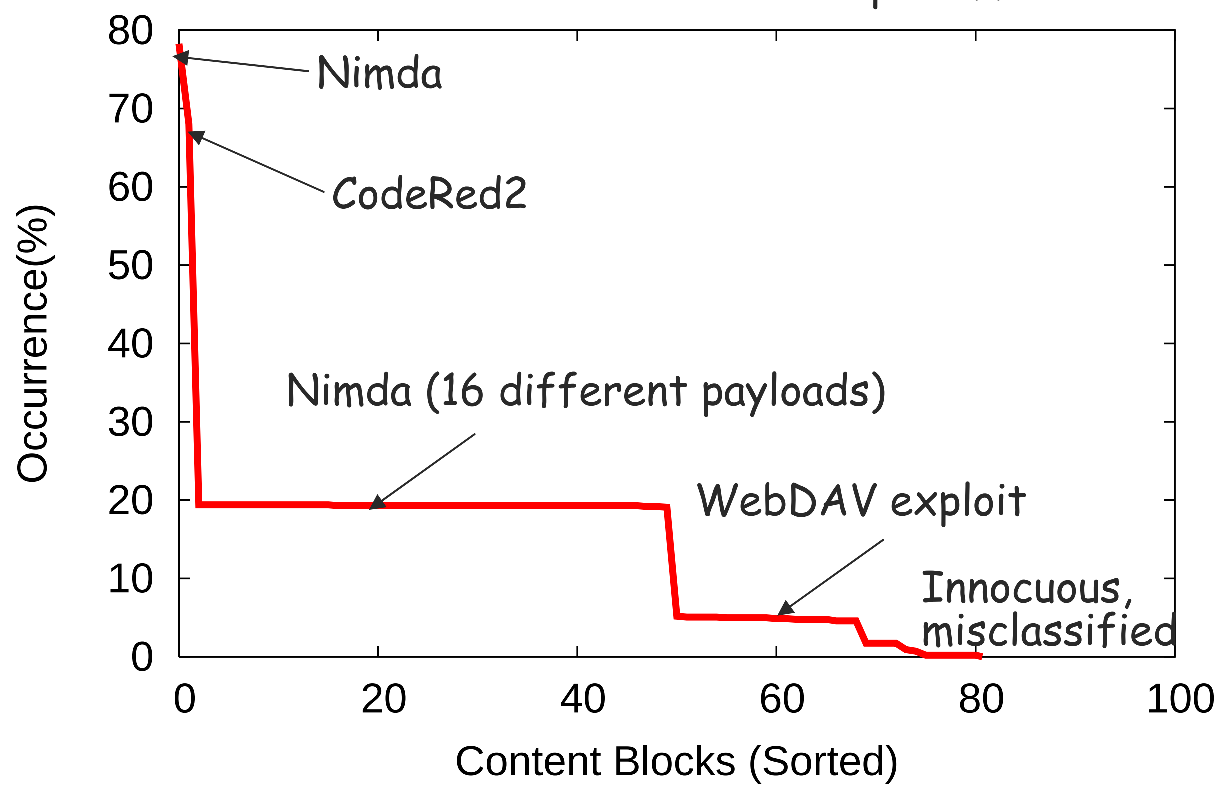


Breakmark = last 8 bits of fingerprint (ABCD)

Why Prevalence?

Prevalence Distribution in Suspicious Flow Pool

- From 24-hr http traffic trace

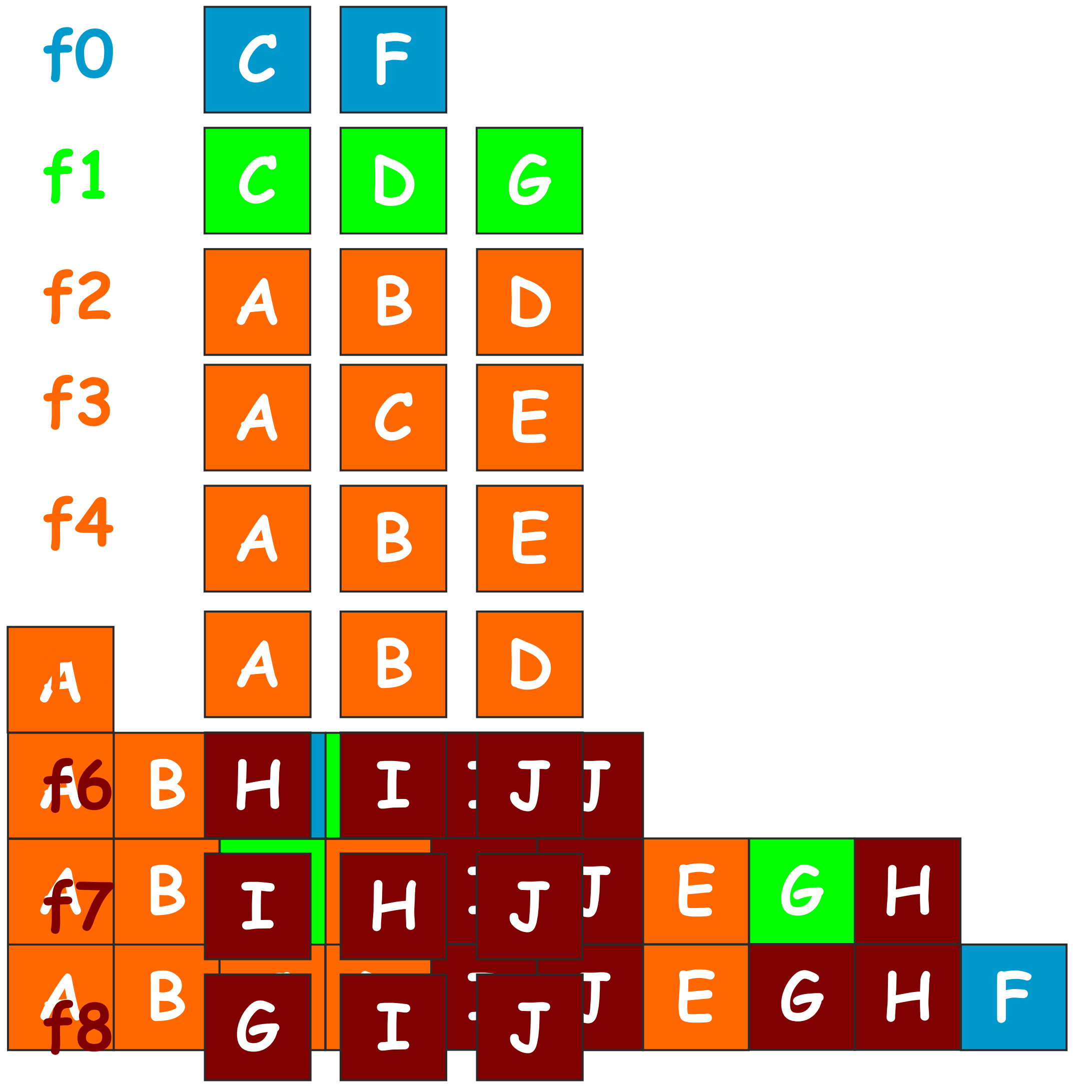


- Worm flows dominate in the suspicious flow pool
- Content-blocks from worms are highly ranked

Select Most Frequent Content Block

f0	C	F	
f1	C	D	G
f2	A	B	D
f3	A	C	E
f4	A	B	E
f5	A	B	D
f6	H	I	J
f7	I	H	J
f8	G	I	J

Select Most Frequent Content Block



f0	C F
f1	C D G
f2	A B D
f3	A C E
f4	A B E
f5	A B D
f6	H I J
f7	I H J
f8	G I J

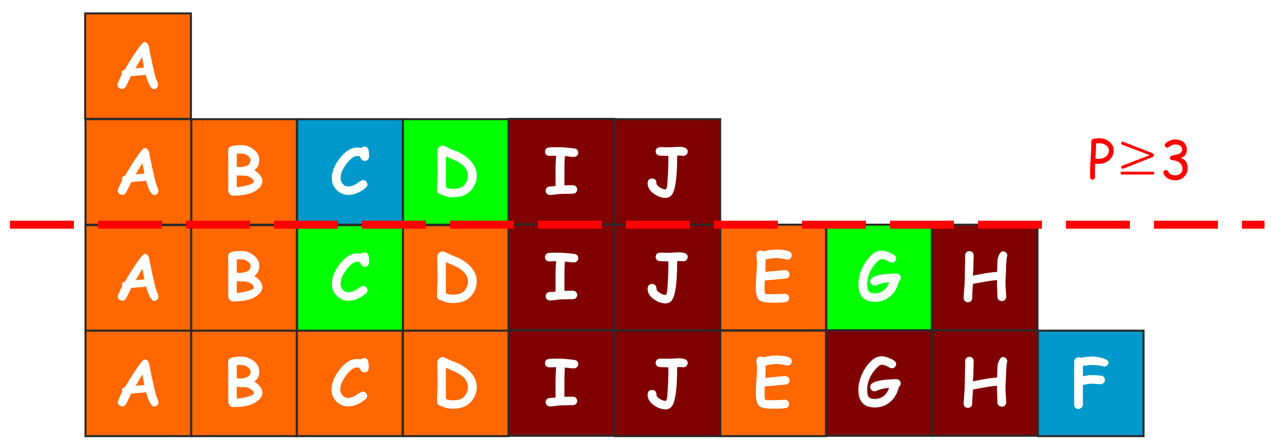
Select Most Frequent Content Block

Signature:

W≥90%

f0	C F
f1	C D G
f2	A B D
f3	A C E
f4	A B E
f5	A B D
f6	H I J
f7	I H J
f8	G I J

W: target coverage in suspicious flow pool
P: minimum occurrence to be selected



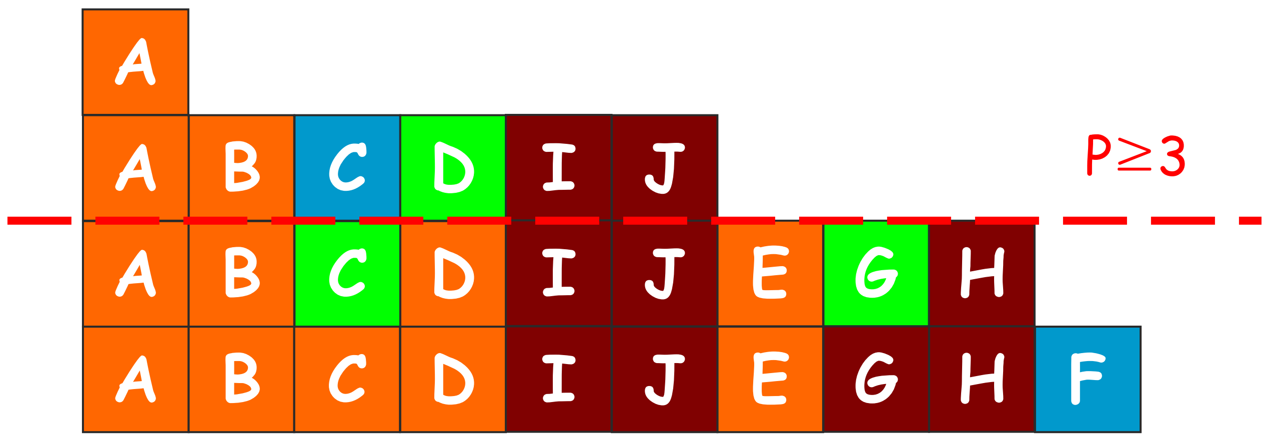
Select Most Frequent Content Block

Signature: *A*

W≥90%

f0	C F
f1	C D G
f2	A B D
f3	A C E
f4	A B E
f5	A B D
f6	H I J
f7	I H J
f8	G I J

W: target coverage in suspicious flow pool
P: minimum occurrence to be selected



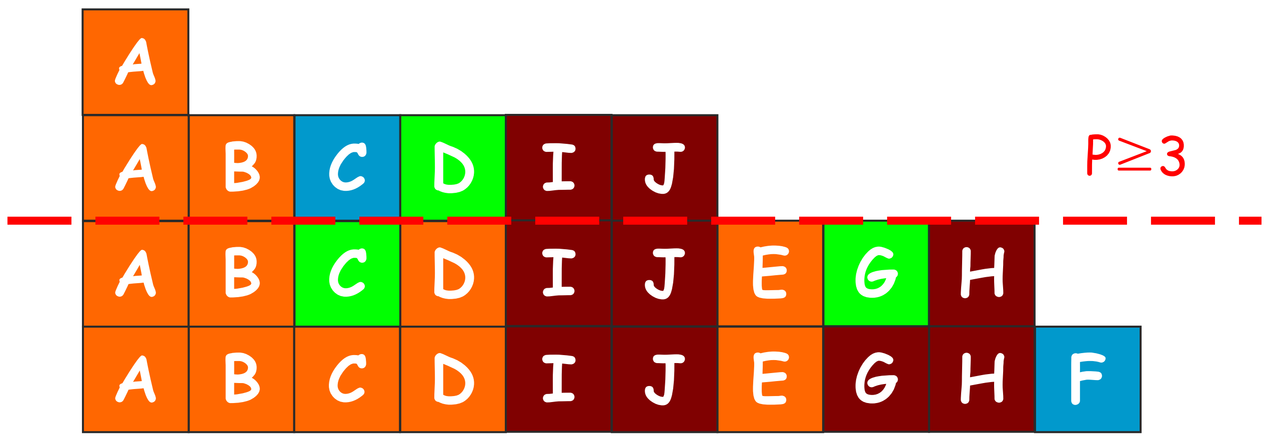
Select Most Frequent Content Block

Signature: *A*

W≥90%

f0	C F
f1	C D G
~~f2~~	~~A B D~~
~~f3~~	~~A C E~~
~~f4~~	~~A B E~~
~~f5~~	~~A B D~~
f6	H I J
f7	I H J
f8	G I J

W: target coverage in suspicious flow pool
P: minimum occurrence to be selected



Select Most Frequent Content Block

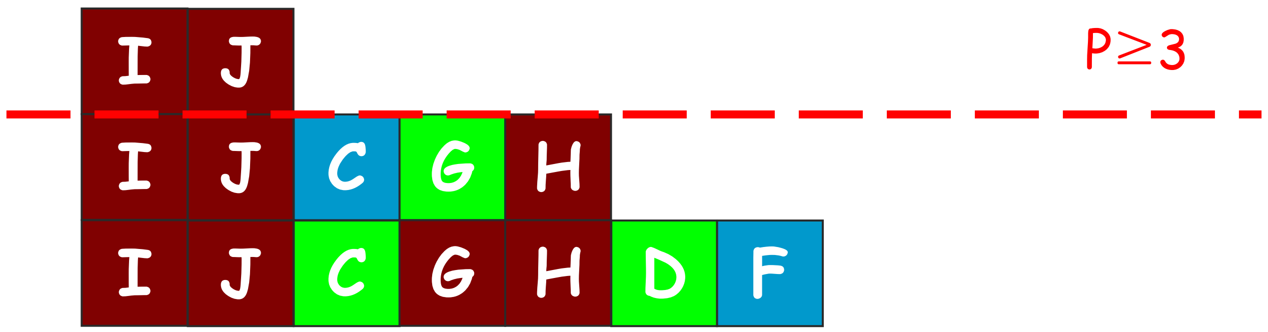
Signature: A I

W≥90%

f0	C F
f1	C D G
~~f2~~	~~A B D~~
~~f3~~	~~A C E~~
~~f4~~	~~A B E~~
~~f5~~	~~A B D~~
~~f6~~	~~H I J~~
~~f7~~	~~I H J~~
~~f8~~	~~G I J~~

W: target coverage in suspicious flow pool
P: minimum occurrence to be selected

P≥3



Select Most Frequent Content Block

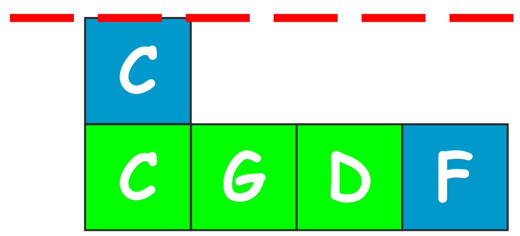
Signature: A I

W≥90%

W: target coverage in suspicious flow pool
P: minimum occurrence to be selected

f0	C F
f1	C D G
~~f2~~	~~A B D~~
~~f3~~	~~A C E~~
~~f4~~	~~A B E~~
~~f5~~	~~A B D~~
~~f6~~	~~H I J~~
~~f7~~	~~I H J~~
~~f8~~	~~G I J~~

P≥3



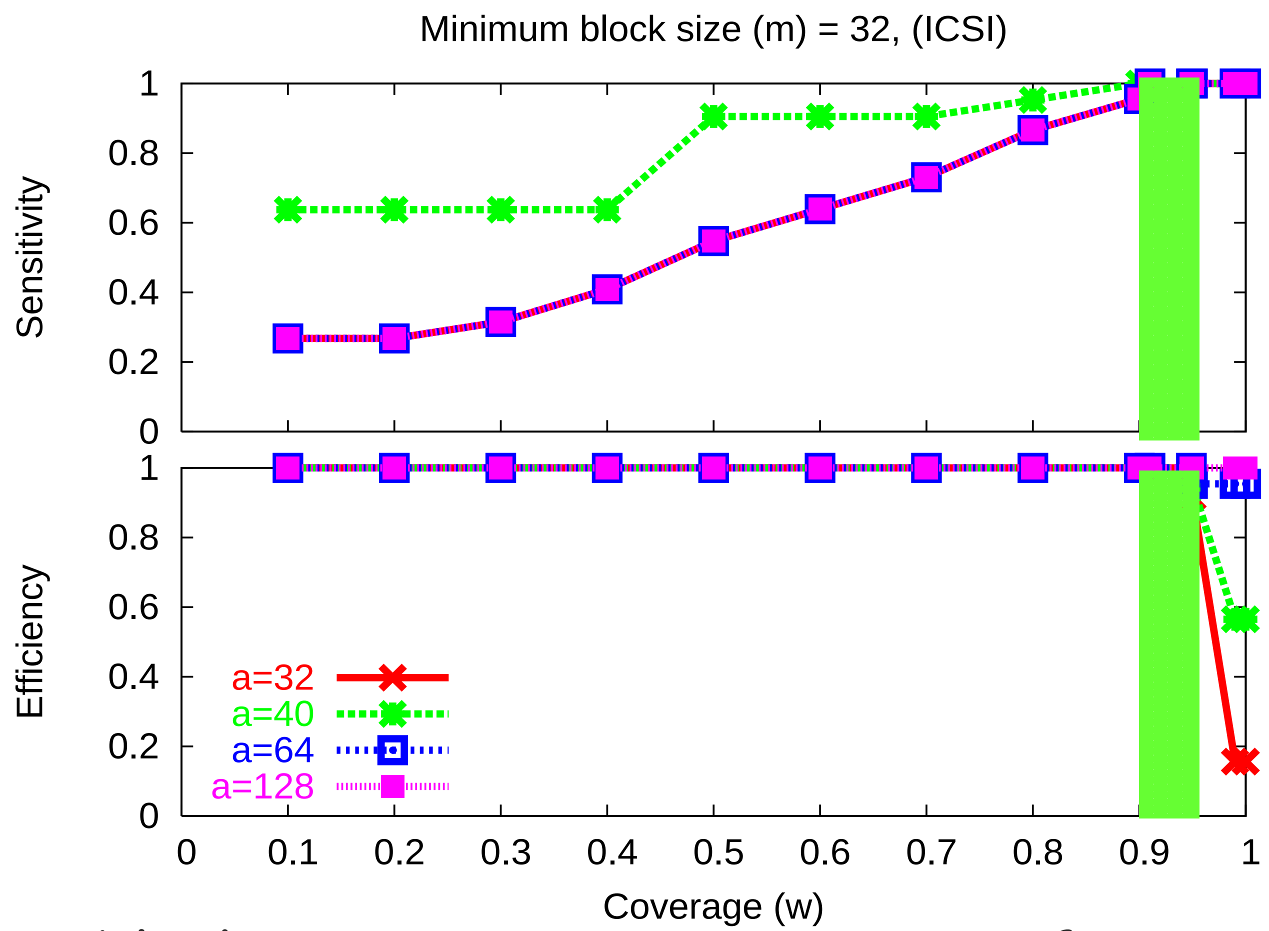
Outline

- Problem and Motivation
- Automated Signature Detection
  - Desiderata
  - Technique
  - Evaluation
- Distributed Signature Detection
  - Tattler
  - Evaluation
- Related Work
- Conclusion

Behavior of Signature Generation

- Objectives
  - Effect of COPP parameters on signature quality
- Metrics
  - Sensitivity = # of true alarms / total # of worm flows ⇒ false negatives
  - Efficiency = # of true alarms / # of alarms ⇒ false positives
- Trace
  - Contains 24-hour http traffic
  - Includes 17 different types of worm payloads

Signature Quality



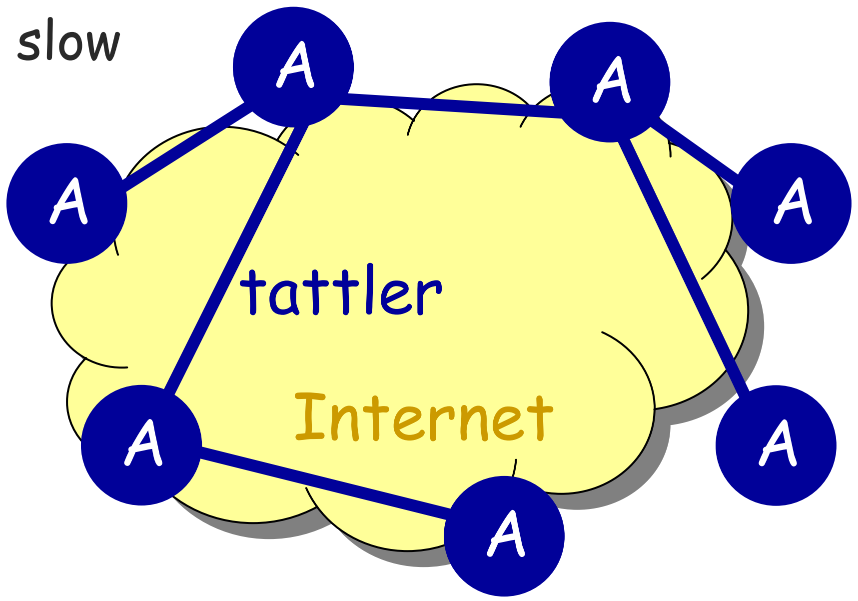
- Larger block sizes generate more specific signatures
- A range of w (90-95%, workload dependent) produces a good signature

Outline

- Problem and Motivation
- Automated Signature Detection
  - Desiderata
  - Technique
  - Evaluation
- Distributed Signature Detection
  - Tattler
  - Evaluation
- Related Work
- Conclusion

Signature Generation Speed

- Bounded by worm payload accumulation speed
  - Aggressiveness of scanner detection heuristic
    - s: # of failed connection peers to detect a scanner
  - # of payloads enough for content analysis
    - θ: suspicious flow pool size to trigger signature generation
- Single Autograph
  - Worm payload accumulation is slow
- Distributed Autograph
  - Share scanner IP list
  - Tattler: limit bandwidth consumption within a predefined cap



Benefit from tattler

- Worm payload accumulation (time to

Many innocuous misclassified flows

Info Sharing	Autograph Monitor	Fraction	
		Aggressiv (s = 1)	ve (s = 4)
None	Luckiest	2%	60%
	Median	25%	--
Tattler	All	<1%	15%

- Signature generation
  - More aggressive scanner detection (s) and signature generation trigger (θ) ⇒ faster signature generation, more false positives
  - With s=2 and θ=15, Autograph generates the good worm signature before < 2% hosts get infected

Related Work

- Automated Worm Signature Detection

	EarlyBird [Singh et al. 2003]	HoneyComb [Kreibich et al. 2003]	Autograph
Signature Generation	Content prevalence → Address Dispersion	Honeypot + Pairwise LCS	Suspicious flow selection → Content prevalence
Deployment	Network	Host	Network
Flow Reassembly	No	Yes	Yes
Distributed Monitoring	No	No	Yes

- Distributed Monitoring
  - Honeyd[Provos2003], DOMINO[Yegneswaran et al. 2004]
  - Corroborate faster accumulation of worm payloads/scanner IPs

Future Work

- Attacks
  - Overload Autograph
  - Abuse Autograph for DoS attacks
- Online evaluation with diverse traces & deployment on distributed sites
- Broader set of suspicious flow selection heuristics
  - Non-scanning worms (ex. hit-list worms, topological worms, email worms)
  - UDP worms
- Egress detection
- Distributed agreement for signature quality testing
  - Trusted aggregation

Conclusion

- Stopping spread of novel worms requires early generation of signatures
- Autograph: automated signature detection system
  - Automated suspicious flow selection→ Automated content prevalence analysis
  - COPP: robustness against payload variability
  - Distributed monitoring: faster signature generation
- Autograph finds sensitive & specific signatures early in real network traces

For more information, visit

http://www.cs.cmu.edu/~hakim/autograph

Attacks

- Overload due to flow reassembly

  Solutions

  ⇒ Multiple instances of Autograph on separate HW (port-disjoint)

  ⇒ Suspicious flow sampling under heavy load

- Abuse Autograph for DoS: pollute suspicious flow pool
  - Port scan and then send innocuous traffic

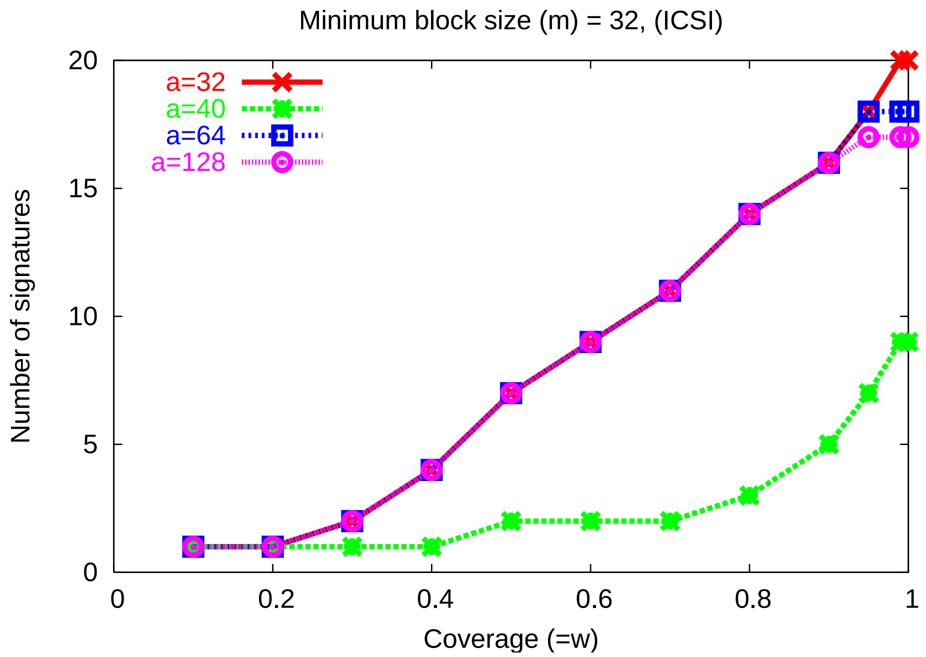
    Solution

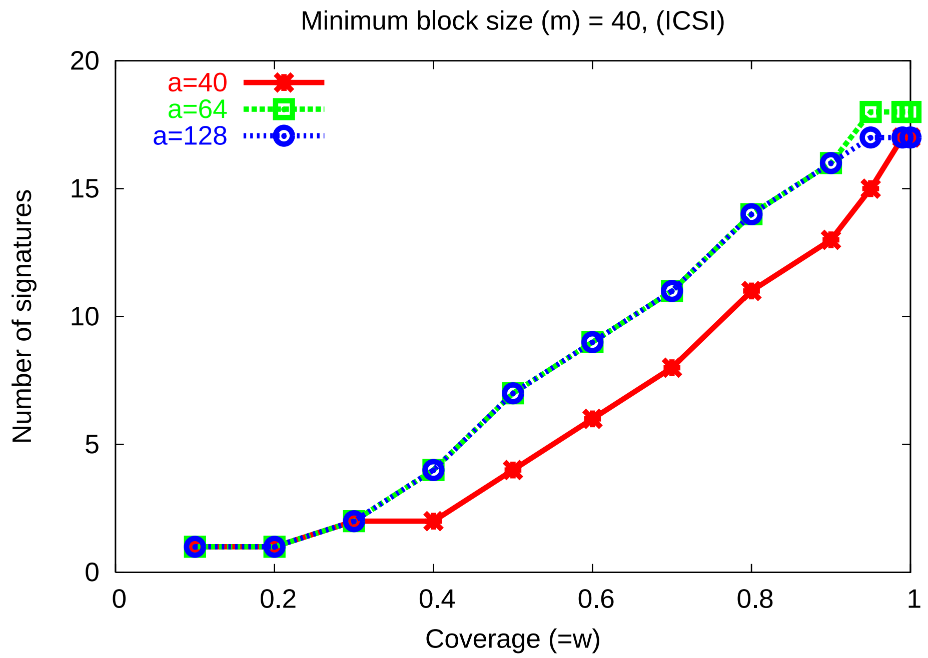
    ⇒ Distributed verification of signatures at many monitors
  - Source-address-spoofed port scan

    Solution

    ⇒ Reply with SYN/ACK on behalf of non-existent hosts/services

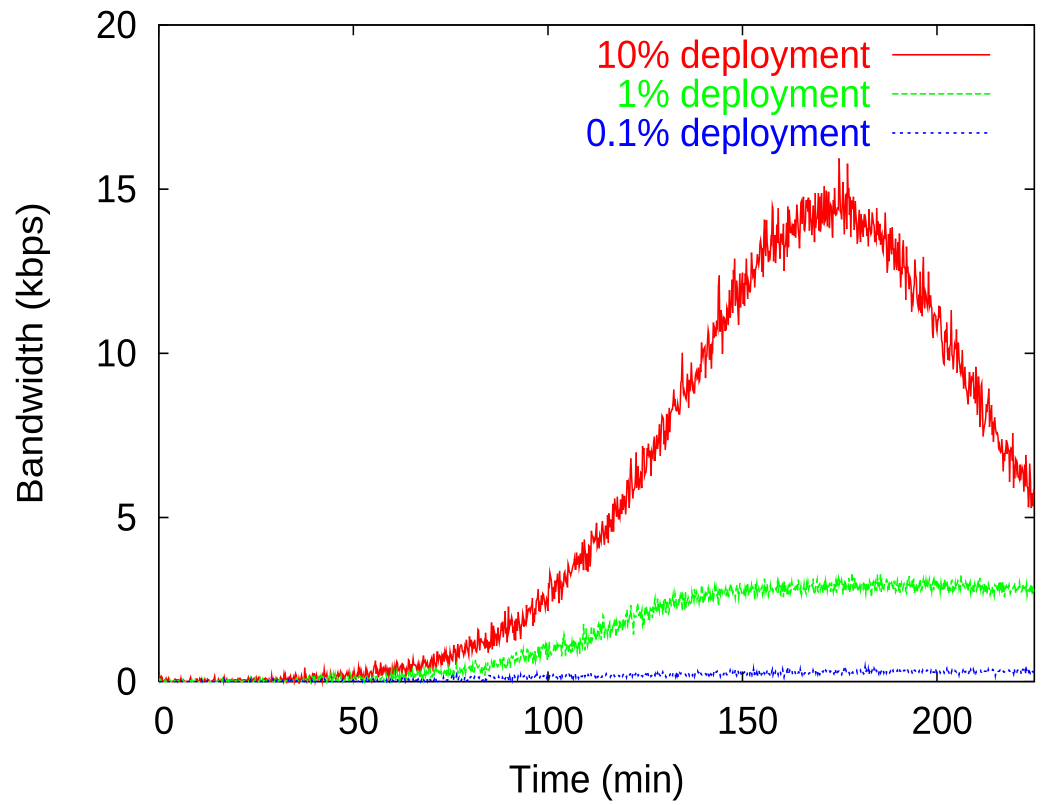
Number of Signatures





- Smaller block sizes generate small # of signatures

tattler



- A modified RTCP (RTP Control Protocol)
- Limit the total bandwidth of announcements sent to the group within a predetermined cap

Simulation Setup

- About 340,000 vulnerable hosts from about 6400 ASes
- Took small size edge networks (/16s) based on BGP table of 19th of July, 2001.
- Service deployment
  - 50% of address space within the vulnerable ASes is reachable
  - 25% of reachable hosts run web server
  - 340,000 vulnerable hosts are randomly placed.
- Scanning
  - 10probes per second
  - Scanning the entire non-class-D IP address space
- Network/processing delays
  - Randomly chosen in [0.5, 1.5] seconds