Notes and Observations from RAID 2005

NEbraskaCERT CSF

September 21, 2005

Stephen M. Nugen, CISSP
Senior Research Fellow
Nebraska University Consortium for Information Assurance
College of IS&T, Peter Kiewit Institute
University of Nebraska, Omaha

Overview

- Selected observations from RAID 2005 Conference
  - No claims of completeness, accuracy, or fairness
  - Hope is that some of these topics will provoke your own ideas and contributions
- Presentation relies on preliminary proceedings, subjective memories, and handwritten notes
  - All the good stuff is credited to the Conference authors and presenters
  - All errors are Nugen's fault

RAID 2005 Conference

- RAID 2005: Eighth International Symposium on Recent Advances in Intrusion Detection
  - Sept 6 - 9, 2005; Seattle Washington
  - http:www.conjungi.com/RAID
  - Official Conference Proceedings will be published by Springer
- Context
  - Yearly since 1998
    - Location alternates between North America and Europe

RAID 2005 Conference cont'd

- Context cont'd
  - ~130 attendees in 2005
    - Academia, National Labs, Industry (providers and enterprises)
  - Program Selections
    - 25-member program selection committee
    - Single session for all papers
      - Accepted 17 papers from 83 submissions
      - 30-min presentations
    - Posters (first time)
      - Accepted 13 posters from 15 submissions
      - 5-min presentation to introduce topic

Keynote

- Speaker: Phil Attfield
  - Formerly Boeing; now Northwest Security Institute
  - Key witness for government prosecution of "Flyhook" case, US vs. Gorshkov (and others) in 1999
- Multiple ISPs and organizations compromised, then contacted by consultants offering information security services
  - Declining the service led to more intrusions
  - At least one ISP agreed to consulting terms

Keynote cont'd

- Two attackers persuaded to visit fictitious US company for job interview
  - Job interview included demonstrating hacking skills using FBI-supplied computers with keyloggers
- Information from keyloggers used to interrogate attacker's servers in Russia, finding:
  - 56,000 credit card numbers
  - Hacked passwords
  - Victim network topologies
  - Previously unknown victims, including PayPal, eBay, and Verio

Keynote cont'd

- Discovered tools included
  - Virtual browser with free email (using other services like hotmail and yahoo for backend... through a TCP relay)
  - Special "front end" to eBay and PayPal services
    - Spoofed sellers, bidders, and raters (all of them satisfied customers)... attracting real bidders with real funds
  - Developing new tools to exploit race conditions
- Outcomes
  - US Court convicted two Russian hackers
  - Russian court convicted FBI agent, in abstentia

Keynote cont'd

- Key point, repeatedly reiterated, several times...
  - *Information Security $\neq$ Business Security*
  - The most damaging compromises were not computer intrusions that would be detected by IDS
    - Attacks not aimed at the operating system or software, but rather at information processes and data exchanges
    - Initial processes and data exchanges for online business like eBay and PayPal developed very rapidly, with less rigor and maturity than normal for financial institutions

Keynote cont'd

- Speaker's challenge
  - – Incorporate more business logic into intrusion systems and processes
    - Take lessons from fraud detection
    - Define normal behavior and identify deviations from normal
  - – Distinguish between
    - Actual (human) users
    - Synthetic users (agents)
      - – Speaker: All, or nearly all, synthetic users are hostile
      - – Timing patterns insufficient discriminator in some cases (e.g., web browser with autocomplete)

Intrusion Detection Context

- Acronyms
  - ID = Intrusion Detection
  - IDS = Intrusion Detection System
  - NIDS = Network-based IDS (passive)
  - HIDS = Host-based IDS
  - IPS = Intrusion Prevention System (inline)
- Four states
  - True negative: No intrusion, no detection
  - True positive: Actual intrusion, detected
  - False negative: Actual intrusion, not detected
  - False positive: No intrusion, but detected as one

Intrusion Detection Context cont'd

- Signature-based IDS
  - Detect intrusion by comparing observed behavior to patterns of known misuse (signatures)
    - No match => no intrusion
    - Match => intrusion
  - Common use
  - Few false positives
  - More false negatives
  - Corresponds to security policy: Permit everything not expressly prohibited

Intrusion Detection Context cont'd

- Anomaly-based IDS
  - Detect intrusion by comparing observed behavior to patterns of known normal use
    - No match => Intrusion
    - Match => No intrusion
  - Uncommon in use; common in research
  - More false positives
  - Fewer false negatives
  - Corresponds to security policy: Deny everything not expressly permitted

Rate Limiting Mechanisms

- Reference
  - Cynthia Wong, Stan Bielski, Ahren Studer, and Chenxi Wang
  - Empirical Analysis or Rate Limiting Mechanisms
  - In *Recent Advances In Intrusion Detection (RAID) 2005*, September 2005
  - All authors from Carnegie Mellon University; supported by National Science Foundation

Rate Limiting Mechanisms cont'd

- Motivation: Constrain the harmful effects of worm propagation
  - Spreading the infection to new hosts
  - Collateral impact of worm-generated network traffic impacting non-infected host communications
- Goal: Interfere with worms' network traffic without preventing legitimate traffic
  - Requires detecting worms based on their behavior
  - False positives limit or prevent legitimate traffic
  - False negatives permit worm propagation

Rate Limiting Mechanisms cont'd

- Evaluation context
  - Authors evaluated methods developed by others and themselves
  - Used actual traffic traces collected from academic network border
    - 1200 externally routed hosts with multiple operating systems
    - Traffic traces included Blaster and Welchi worms
      - Infected 100 hosts
      - Increased outbound traffic volume from 500K to 11,000K flows/day

Rate Limiting Mechanisms cont'd

- Williamson's IP Throttling Scheme
  - Method
    - Normal applications typically exhibit a stable contact rate to limited number of external hosts
    - Create a "working set" of IP addresses
    - Connection requests to destinations in working set permitted without delay
    - Connection requests to destinations not in working set are routed to FIFO delay queue
    - When queue is full, new requests are dropped
    - Requests in delay queue satisfied at preset rate

Rate Limiting Mechanisms cont'd

- Williamson's IP Throttling Scheme cont'd
  - Performance parameters
    - Size of working set (e.g., 5 entries)
    - Size of delay queue (e.g., 100 entries)
    - Rate at which requests in delay queue are satisfied (granted and removed from queue) (e.g., 1/sec)
  - Findings
    - Varying the working set size from 4..10 entries had measurable, but not significant impact
    - Port scans cause delay queue to overflow
    - Works better when implemented at the host, rather than at edge router

Rate Limiting Mechanisms cont'd

- Williamson's IP Throttling Scheme cont'd
  - Findings cont'd
    - Normal host, 3-hour period
      - Most packets not delayed
      - False positive rate: 18%
      - Packets sent to delay queue delayed ~3 seconds

Delay Amt (seconds)	Benign Flows #	Benign Flows %
No delay	1,759	82%
1 - 10	385	18%
11 - 20	0	0%
Totals	2,144	100%

    - Infected host, 3-hour period
      - 91% of worm traffic dropped 97% of legit traffic dropped
      - Net benefit...

Delay Amt (seconds)	Benign Flows #	Benign Flows %	Malicious Flows #	Malicious Flows %
No delay	1	1%	12	0%
1 - 30	1	1%	36	0%
31 - 60	1	1%	36	0%
61 - 90	1	1%	50	0%
91 - 100	0	0%	10,115	9%
Dropped	141	97%	107,080	91%
Totals	145	100%	117,329	100%

Rate Limiting Mechanisms cont'd

- Chen's Failed Connection Rate Limiting (FC)
  - Method
    - Infected hosts generate large number of failed TCP requests
    - Edge router stores failure statistics for each host
      - Failed request iff destination responds with TCP RST
        - » *Flawed assumption that understates # of failed connections*

Rate Limiting Mechanisms cont'd

- Chen's FC Rate Limiting
  - Method cont'd
    - Basic rate limiting
      - Define short-term failure rate (SFR)
      - When # of failed connections exceeds SFR, further connections are limited using leaky bucket token algorithm
        - » Every failed connection removes token from bucket
        - » When the bucket is empty, connection requests are dropped
        - » Every SFR seconds, one token is added to bucket

Rate Limiting Mechanisms cont'd

- Chen's FC Rate Limiting cont'd
  - Method cont'd
    - Temporal rate limiting
      - Like basic, but adds a daily failure rate (DFR, e.g. 300/day) to contain less aggressive worms
  - Findings
    - Worms quickly deplete available tokens with the result that worm propagation is severely constrained
    - Some bursty applications falsely detected as worms, constraining their legitimate traffic
      - Ex: Peer-to-peer sharing, HTTP clients
    - Expanding definition of failed request to include TCP Timeouts reduced number of false negatives

Rate Limiting Mechanisms cont'd

- Schechter's Credit-based Rate Limiting (CB)
  - Method
    - Rate limits just first-contact connections (Outgoing requests to destinations not previously contacted)
    - Stores statistics about failures and successes
    - CB limiting
      - Preset # of credits allocated to each host
      - Every first-contact failed connection removes one credit
      - Every first-contact successful connection adds one credit

Rate Limiting Mechanisms cont'd

- Schechter's Credit-based Rate Limiting (CB)
  - Method cont'd
    - CB limiting cont'd
      - When there are no credits, first-contact requests are dropped
        - » No effect on traffic to destinations previously visited
        - » *Presented paper did not address how a credit balance of zero is ever changed to non-zero*

Rate Limiting Mechanisms cont'd

- Schechter's Credit-based Rate Limiting (CB)
  - Findings
    - Fewer false positives than Chen's FC Rate Limiting for bursty applications
    - Host implementation
      - Average false positives: 8%
      - Average false negatives: 5%

Rate Limiting Mechanisms cont'd

- DNS-Based Rate Limiting
  - Developed by authors using Ganger's observation
    - Worms induce different DNS statistics than legitimate applications... they make connect requests without first making a DNS request
  - Method
    - Connection requests to destinations with previous DNS translation is permitted without delay
    - Connection requests to untranslated destinations are delayed by Cascading Bucket Scheme

Rate Limiting Mechanisms cont'd

- DNS-Based Rate Limiting cont'd
  - Method cont'd
    - Cascading Bucket Scheme (vastly oversimplified)
      - Non-translated requests are inserted into logical buckets
      - Each bucket contains a fixed number of entries
      - Buckets cascade into each other
      - Buckets are emptied at preset rate
      - When buckets are full, new requests are dropped

Rate Limiting Mechanisms cont'd

- DNS-Based Rate Limiting cont'd
  - Findings (compare to Williamson's IP Throttle)
    - Normal host, 3-hour period

IP Throttled

Delay Amt (seconds)	Benign Flows	
	#	%
No delay	1,759	82%
1 - 10	385	18%
11 - 20	0	0%
Totals	2,144	100%

DNS Rate Limited

Delay Amt (seconds)	Benign Flows	
	#	%
No delay	2,136	100%
1 - 10	8	0%
>10	0	0%
Totals	2,144	100%

Rate Limiting Mechanisms cont'd

- DNS-Based Rate Limiting cont'd
  - – Findings cont'd
    - Infected host, 3-hour period

IP Throttled

Delay Amt (seconds)	Benign Flows #	Benign Flows %	Malicious Flows #	Malicious Flows %
No delay	1	1%	12	0%
1 - 30	1	1%	36	0%
31 - 60	1	1%	36	0%
61 - 90	1	1%	50	0%
91 - 100	0	0%	10,115	9%
Dropped	141	97%	107,080	91%
Totals	145	100%	117,329	100%

DNS Rate Limited

Delay Amt (seconds)	Benign Flows #	Benign Flows %	Malicious Flows #	Malicious Flows %
No delay	806	80%	1	0%
1 - 30	4	0%	34	0%
31 - 60	2	0%	35	0%
61 - 100	12	1%	40	0%
> 100	11	1%	4,903	4%
Dropped	172	17%	112,862	96%
Totals	1,007	100%	117,875	100%

COTS Diversity Intrusion Detection

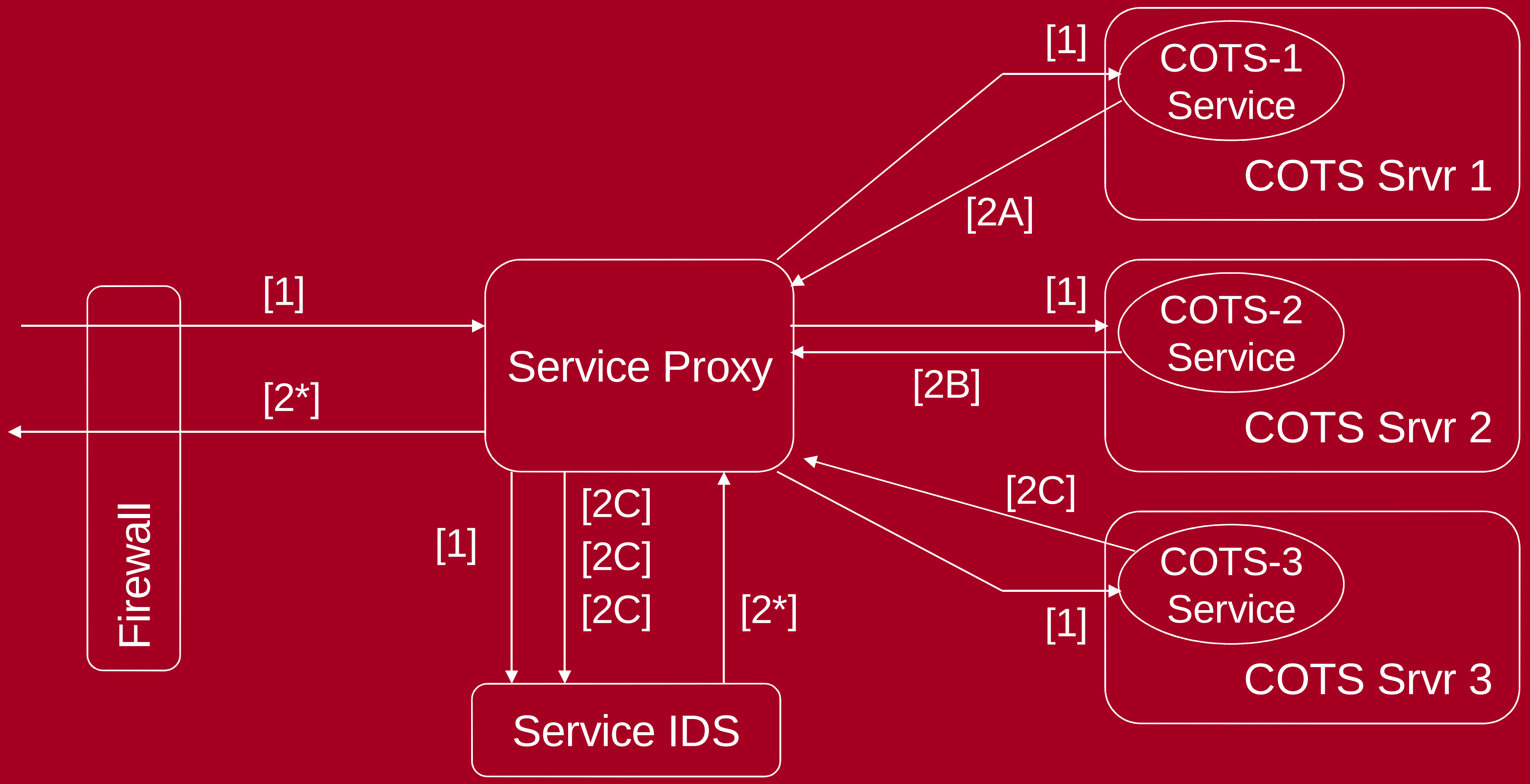
- Reference
  - Eric Totel, Frédéric Majorczyk, Ludovic Mé
  - COTS Diversity Based Intrusion Detection and Application to Web Servers
  - In *Recent Advances In Intrusion Detection (RAID) 2005*, September 2005
  - All authors from Supélec, Cesson-Sévigné Cedex, France, partly supported by Conseil Régional de Bretagne, part of French Ministry of Research DADDi project

COTS Diversity Intrusion Detect cont'd

- Approach
  - Detect anomalous behavior based on design diversity
  - Consider different COTS programs
    - Implemented to a common spec (e.g. Web Server)
    - Provided with common inputs
  - Any differences in resulting output must be due to
    - (1) Design and implementation differences; or
    - (2) One of the programs being compromised
  - If we can distinguish between these two causes, we can detect actual intrusions
    (most IDS detect potential intrusions)

COTS Diversity Intrusion Detect cont'd

- Simplified Architecture



COTS Diversity Intrusion Detect cont'd

- Service IDS compares outputs of all servers
  - Differences classified
    - Due to design/implementation differences not associated with vulnerability: False Positive, so no alert
    - Due to compromise: True Positive, Alert
  - Voting used to select highest-confidence value to return to client
- Findings
  - 36 rules sufficient to mask legitimate design differences... difficult to create and maintain
  - Low overhead, few false positives

Behavioral Distance for Intrusion Detect

- Reference
  - Debin Gao, Michael Reiter, and Dawn Song
  - Behavioral Distance for Intrusion Detection
  - In *Recent Advances In Intrusion Detection (RAID) 2005*, September 2005
  - All authors from Carnegie Mellon University

Behavioral Distance for ID cont'd

- Approach
  - Define Behavioral Distance as the extent to which different processes (potentially running different programs on different operating systems) behave similarly in response to common input
    - Inspired by Evolutionary Distance, a method used in biology to align different DNA sequences that arise from common ancestor, but have changed
  - The more similar the processes (replicas)
    - The easier it is to compare them
    - The less value there is in comparing them since they are more likely to be compromised at the same time

Behavioral Distance for ID cont'd

- Approach cont'd
  - Sensed behavior: How the process interacts with the operating system... sequences of system calls
    - Different than previous method which sensed external outputs
  - Some options illustrated
    - Compare Apache to IIS on Windows O/S
    - Compare Apache on Windows to Apache on Linux
      - This is the option chosen by the authors... comparing the Apache system call sequence on Linux to the Apache system call sequence on Windows... very different system calls

Behavioral Distance for ID cont'd

- Approach cont'd
  - Observed sequences of system calls organized into phrases (subsequences)
    - This reduces the need to consider the arguments to the system calls
  - Equivalence learned through clean training data...
    - Minimizing the behavioral distance between two functionally equivalent sequences
    - Learning stops when stored table of behavioral distances stable
  - In operation, if the behavioral distance between two replicas > threshold, detect as intrusion

Behavioral Distance for ID cont'd

- Findings
  - Evaluated six replicas
    - Three different web servers (Apache, Myserver, and Abyss)
    - Executing on two operating systems (Linux and Windows)
  - After training
    - Nominal requests and responses exhibit very small behavioral distances
    - Intrusions detected, even those that emulated mimicry attacks
    - Throughput overhead measured at ~7%
    - Latency overhead measured at ~6%

Software-Based IDS on NIC

- Reference
  - H. Bos and Kaiming Huang
  - Towards Software-Based Signature Detection for Intrusion Prevention on the Network Card
  - In *Recent Advances In Intrusion Detection (RAID) 2005*, September 2005
  - Bos is from Vrije Universiteit, Amsterdam, The Netherlands
  - Huang is from Xiamen University, Xiamen, China

Software-Based IDS on NIC cont'd

- Motivation
  - Move IDS closer to the host
  - NIC cards more capable than hosts for high data rates
    - 69Mbps overwhelms signature-based IDS on host with 1.8GHz P4
  - NIC card harder to subvert
- Approach: Signatures implemented in deterministic finite automatons
- Note: Purdue Grad Student porting Snort to NIC card, independent of this effort

Others

- Virtual playground to study worm propagation
  - Hosts, routers, switches
  - Demonstrated creating 41-host network in < 2 min implemented on single workstation
  - Reported ability to create 2000-3000 virtual nodes with 10 workstations
- Multiple schemes to detect executable code in network payloads...
- Visualizations to detect port scans

Others cont'd

- Detection scheme monitoring for dynamically created executable code in user space found two such cases
  - Windows... until it is activated
  - Java and JVM
- Special multi-agency focus on SCADA systems
- Higher-level reasoning systems

Others cont'd

- Detecting link-layer MAC spoofing in wireless networks
  - Some attacks work at link layer, not detected by IP-based IDS

- Handling very large volumes of sensor data (200M records/day)

Questions, Contributions?

- Presenter contact info
  - Steve Nugen
  - smnugen@nugensoft.com
  - smnugen@nucia.unomaha.edu
  - 402.554.3007