An Architecture for Generating Semantics-Aware Signatures

Vinod Yegneswaran, Jonathon T. Giffin, Paul Barford, Somesh Jha

University of Wisconsin
{vinod,giffin,pb,jha}@cs.wisc.edu

2005 USENIX Security Symposium
Worldview

Signature Repository

Not You

Packet Traces

IDS

IPS

Signature Database

You
Worldview

Signature Repository

Not You

Packet Traces

IDS

IPS

Signature Database

You
Worldview

Signature Generator

Signature Database

Packet Traces

IDS

IPS

4 August 2005
An Architecture for Generating Semantics-Aware Signatures
Automatic Signature Generation

Specific signatures
Identify only characteristics of attack profiles

General signatures
Match variants of known attack profiles

Balance specificity and generality
Related Work

Controlled virus infection [Kephart & Arnold 1994]
Honeycomb [Kreibich & Crowcroft 2003]
Autograph [Kim & Karp 2004]
Earlybird [Singh et al. 2004]
Polygraph [Newsome et al. 2005]

• Not aware of application-level protocol semantics
  – Distracted by irrelevant byte sequences
  \\r\\nConnection: Keep-Alive\\n\\r\\n• Worm-oriented
• Real-time use
Semantics-Aware Signatures

- Aggregate TCP flows
- Canonical encoding of HTTP URLs
- Field weights indicate significance of data

- Defragment IP packets
- Reassemble TCP flows
- Prevent insertion & evasion attacks
Semantics-Aware Signatures

- Generate signatures for attacks where the exploit is a small part of entire payload
- Generate contextual connection- and session-level signatures for multi-step attacks
- Produce generalized signatures from small number of training samples
- Produce signatures that are easy to understand & validate
Architecture

1. Data Collector
2. Flow Aggregator
3. Service Normalizer

Protocol Semantics

4. Connection Clustering
5. Session Clustering

Signature Generalizer

Standard IDS/IPS Signatures
Data Collection

• Problem: build signatures only for malicious traffic

• Solution: collect traffic sent to honeynet
  – Routed but unused IPs
  – Legitimate traffic never sent to honeynet
  – Actively respond to HTTP & NetBIOS traffic

[Yegneswaran et al. 2004]
Flow Aggregation

Session
Source IP
Destination IP

Connection
Source port
Destination port

Request
Response
Flow Aggregation & HTTP Semantics

Session
Source IP
Destination IP

Connection
Source port
Destination port

Request
Method
URL
Header

Response
Code #
Reason
Header

...
Flow Aggregation & HTTP Semantics

attacker:2492 → honeypot:80
   GET /scripts/root.exe?/c+dir
   Connection: Close

attacker:2492 ← honeypot:80
   404 Object Not Found

Nimda exploiting Code Red backdoor
### Flow Aggregation & HTTP Semantics

**Session**
- **Source IP**: "attacker"
- **Destination IP**: "honeynet"

**Connection**
- **Source port**: 2492
- **Destination port**: 80

**Request**
- **Method**: GET
- **URI**: 
- **Headers**:
  - weight 1000: 
  - weight 1: /scripts
  - weight 50: /threads?
  - weight 1: /c+dir
  - weight 0: Connection: close

**Response**
- **Code**: 404
- **Reason**:
  - weight 1: Code
  - weight 0: Reason: Object not found
Architecture

Packets → Data Collector → Flow Aggregator → Service Normalizer → Protocol Semantics

Connection Clustering → Session Clustering → Signature Generalizer

Standard IDS/IPS Signatures
Clustering

- Star clustering algorithm
  - Construct similarity graph
    - Connections become nodes
    - Edges between nodes weighted with connection similarity
  - Find a star cover comprised of star clusters
  - Robust to data ordering
  - Algorithm determines number of clusters
- Cosine similarity metric
Connection Clustering

attacker:2492 → honeypot:80
GET /scripts/root.exe?/c+dir
Connection: Close

attacker:2496 → honeypot:80
GET /MSADC/root.exe?/c+dir
Connection: close

attacker:2496 ← honeypot:80
403 Access Forbidden
Connection Clustering

attacker:2492 → honeypot:80
GET /scripts/root.exe?/c+dir
Connection: Close

attacker:2492 → honeypot:80
404 Object Not Found

attacker:2496 → honeypot:80
GET /MSADC/root.exe?/c+dir
Connection: close

attacker:2496 → honeypot:80
403 Access Forbidden
Connection Signature

- PFSA generalization
  - Compute probability that each edge is traversed
  - Merge states when probabilistically indistinguishable
  - Add transitions representing reordering & repetition
Connection Signature

- Subsequence creation
  - Accept any data at points of high variability

Let \( A, B \in \Sigma^* \)

Let \( w, x, y, z \in \Sigma \)

Convert signature accepting \( AwB, AxB, AyB, AzB \) to \( A[.*]B \)
Connection Signature

C1+C2:

GET

/scripts
/root.exe?
/c+dir

404
403

C1+C2:

GET

/scripts
/root.exe?
/c+dir

404
403

Nimda:

GET

.*
/root.exe?
/c+dir

.*
/c+dir
Experiments

• Trained on honeynet data (Two unused /19s)
  – HTTP: \hspace{1em} 2 days \hspace{1em} 25,587 connections
  – NetBIOS: \hspace{1em} 2 days \hspace{1em} 38,722 connections

• Detection effectiveness: 99.9%
  – Test period: \hspace{1em} 7 days \hspace{1em} 2,846,783 connections

• False alarms and misdiagnoses: 0
  – U.Wisc. CSL HTTP production data
    • 19,000 clients \hspace{1em} 4,400 servers
  – Test period: \hspace{1em} 8 hours \hspace{1em} 194,001 connections
## Effective Detection—HTTP

<table>
<thead>
<tr>
<th>Signature</th>
<th># Present</th>
<th>Nemean Detected</th>
<th>Snort (ver 2.1.0) Detected</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Connection</td>
<td>Session</td>
</tr>
<tr>
<td>Options</td>
<td>1172</td>
<td>1172</td>
<td>1160</td>
</tr>
<tr>
<td>Nimda</td>
<td>496</td>
<td>496</td>
<td>n/a</td>
</tr>
<tr>
<td>Propfind</td>
<td>229</td>
<td>229</td>
<td>205</td>
</tr>
<tr>
<td>Welchia</td>
<td>90</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>Win Media Player</td>
<td>89</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>Code Red Retina</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Kazaa</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
## Effective Detection—NetBIOS

<table>
<thead>
<tr>
<th>Signature</th>
<th># Present</th>
<th>Nemean Detected Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Srvsvc</td>
<td>19934</td>
<td>19930</td>
</tr>
<tr>
<td>Samr</td>
<td>8743</td>
<td>8741</td>
</tr>
<tr>
<td>Epmapper</td>
<td>1263</td>
<td>1258</td>
</tr>
<tr>
<td>NvcplDmn</td>
<td>62</td>
<td>61</td>
</tr>
<tr>
<td>Deloder</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>LovGate</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Balancing Specificity & Generality

**Specificity**
- Honeynet data collection
- Clustering
- Application-level protocol semantics-awareness

**Generality**
- Normalization
- PFSA generalization
- Subsequence creation
Questions?

... or send us email:

Vinod Yegneswaran  vinod@cs.wisc.edu
Jonathon Giffin     giffin@cs.wisc.edu
Paul Barford        pb@cs.wisc.edu
Somesh Jha          jha@cs.wisc.edu

Thanks to:

U.Wisc. CSL
Christian Kreibich
Fabian Monrose
Vern Paxson