Polygraph: Automatically Generating Signatures for Polymorphic Worms

Bil656 – 2013 Spring

Yasin KAVAK
N10164820

2005
citation: 489
Content

• Introduction
  • Polymorphic Worms
• Data Patterns & Signature Classes
• PolyGraph Architecture & Signature Generation
• Experiment & Results
• Strength & Weaknesses
• Conclusion
Introduction

• Intrusion Detection Systems (IDS) are vulnerable against polymorphic worms.

• **PolyGraph** – produces signatures for polymorphic worms

• **Automatic** Signature Generation?
Currently Used Techniques By IDS

• string matching at arbitrary payload offsets
• string matching at fixed payload offsets
• matching of regular expressions within a flow’s payload
Polymorphic Worm

• Changes while spreading
• Different with every instance
• Aim not changes

• The code is encrypted with a random key
• Generate a Decryptor for every instance

• main sources of invariant content
  • Exploit framing (reserved key words)
  • Exploit payload (alter control flow)
Polygraph features

• Samples of the same worm shares some invariant content due to the fact that they exploit the same vulnerability

• signatures consist of multiple disjoint content substring

• possible to generate signatures automatically that match the many variants of polymorphic worm

• offer low false positives and low false negatives
Exploits with invariant content

- Apache multiple-host-header vulnerability
- BIND TSIG vulnerability
- Slapper
- SQLSlammer
- CodeRed
- AdmWorm

- Data Patterns
  - **Invariant Bytes** – fixed value
  - **Wildcard Bytes** – completely random value
  - **Code Bytes** (encrypted) – decrypted by an obfuscated routine
**Sample**

<table>
<thead>
<tr>
<th>GET</th>
<th>URL</th>
<th>HTTP/1.1</th>
<th>Random Headers</th>
<th>Host: Payload Part 1</th>
<th>Random Headers</th>
<th>Host: Payload Part 2</th>
<th>Random Headers</th>
</tr>
</thead>
</table>

**Figure 1.** Polymorphed Apache-Knacker exploit. Unshaded content represents wildcard bytes; lightly shaded content represents code bytes; heavily shaded content represents invariant bytes.
Signature Classes

• Conjunction Signatures
  • consists of a set of tokens, and matches a payload if all tokens in the set are found in it, in any order.

• Token-Subsequence Signatures
  • consists of an ordered set of tokens.

• Bayes Signatures
  • consists of a set of tokens, each of which is associated with a score, and an overall threshold.
PolyGraph Architecture
Design Goals

• Signature quality
• Efficient signature generation
• Efficient signature matching
• Generation of small signature sets
• Robustness against noise and multiple worms
• Robustness against evasion and subversion
Signature Generation

- Token Extraction
  - Extract distinct substrings

- Conjunction Signature
  - Select a set of tokens extracted

- Token-Subsequence Signature
  - Find a subsequence of tokens

- Bayes Signature
  - From a set of tokens calculate empirical probabilities
  - Define a threshold
Flow Clustering

- for the token subsequence and conjunction algorithms clustering must be performed
- the suspicious flow pool is divided into several clusters, where each cluster contains similar flows
- Whenever two clusters are merged, the signature generation algorithm being used is run again on the combined set of samples to produce a new, more sensitive signature for the new cluster
- Merging stops when the signature resulting from merging any two clusters would result in an unacceptably high false positive rate, or when there is only one cluster remaining.
Experimental Setup

• Token-extraction threshold $k = 3$
• minimum token length $a = 2$
• minimum cluster size = 3

• Flow Pool Cases
  • One Worm
  • One Worm — With innocuous requests in suspicious flow (flow classifier is imperfect)
  • Multiple Worms & False Labels

• 1.4 GHz Intel R Pentium R III processors, running Linux kernel 2.4.20.
## Apache-Knacker Signatures

<table>
<thead>
<tr>
<th>Class</th>
<th>False +</th>
<th>False −</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longest Substring</td>
<td>92.5%</td>
<td>0%</td>
<td>HTTP/1.1\r\n</td>
</tr>
<tr>
<td>Best Substring</td>
<td>.008%</td>
<td>0%</td>
<td>xFF\xBF</td>
</tr>
<tr>
<td>Conjunction</td>
<td>.0024%</td>
<td>0%</td>
<td>‘GET ’, ‘ HTTP/1.1\r\n’, ‘: ’, ‘\r\nHost: ’, \r\n’, ‘: ’, ‘\r\nHost: ’, ‘xFF\xBF’, ‘\r\n’</td>
</tr>
<tr>
<td>Token Subsequence</td>
<td>.0008%</td>
<td>0%</td>
<td>GET .* HTTP/1.1\r\n.<em>.</em> \r\nHost: .* \r\n.<em>.</em> \r\nHost: .* xFF\xBF.* \r\n</td>
</tr>
<tr>
<td>Bayes</td>
<td>.008%</td>
<td>0%</td>
<td>‘\r\n’: 0.0000, ‘: ’: 0.0000, ‘\r\nHost: ’: 0.0022, ‘GET ’: 0.0035, ‘ HTTP/1.1\r\n’: 0.1108, ‘xFF\xBF’: 3.1517. Threshold: 1.9934</td>
</tr>
</tbody>
</table>

Succeeded with at least 3 worm samples
BIND-TSIG Signatures

<table>
<thead>
<tr>
<th>Class</th>
<th>False +</th>
<th>False -</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longest Substring</td>
<td>.3279%</td>
<td>0%</td>
<td>\x00\x00\xFA</td>
</tr>
<tr>
<td>Best Substring</td>
<td>.0023%</td>
<td>0%</td>
<td>\xFF\xBF</td>
</tr>
<tr>
<td>Conjunction</td>
<td>0%</td>
<td>0%</td>
<td>‘\xFF\xBF’, ‘\x00\x00\xFA’</td>
</tr>
<tr>
<td>Token Subsequence</td>
<td>0%</td>
<td>0%</td>
<td>\xFF\xBF.*\x00\x00\xFA</td>
</tr>
<tr>
<td>Bayes</td>
<td>.0023%</td>
<td>0%</td>
<td>‘\x00\x00\xFA’: 1.7574, ‘\xFF\xBF’: 4.3295</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Threshold: 4.2232</td>
</tr>
</tbody>
</table>
• Single polymorphic worm & noise
  • conjunction & token subsequence signatures remain the same
  • Bayes signatures are not affected by noise until it grows beyond 80%

• Multiple polymorphic worms & noise
  • conjunction & token subsequence signatures are generated for each type of worm.
  • only one bayes signature is generated that matches all the worms.
Possible Attacks on PolyGraph

• Overtraining Attacks
  • Flood the pool to cause overtrain

• Innocuous Pool Poisoning
  • Send harmful packets to result generating signatures which lead innocuous packets to be labeled as harmful

• Long-tail Attack
  • Attack can already be performed
Strengths

• Automation
• Robust
• Practical with multiple worms
• Applicable
• Low false positive and false negatives
• High Quality Signatures
Weaknesses

• Wide attack area must be searched
• Can be evaded
• Host can be already affected (not preventative)
Questions =(
Thanks =)