

Anonymization of Real Data for IDS Benchmarking

Vidar Evenrud Seeberg

June 8, 2006

Outline

- 1 Outline
- 2 Introduction
- 3 The New Anonymization Methodology
- 4 Experimental
- 5 Conclusions
- 6 Further Work

Background

- Topic: Generation of test data for IDS benchmarking
- Most IDS evaluations use test data based on artificially generated traffic
 - Artificially generated network traffic is not realistic
- A better solution?
 - Test data sets based on real network traffic

Problem

- Recorded network data may contain sensitive information!
 - Cannot be distributed

Research Questions

- How can recorded application layer data be anonymized?
- How can traffic properties needed for intrusion detection be preserved when application layer data is anonymized?

Classification System

Informational objects¹ classified according to identifiable potential

- Must ... be anonymized
- Should ... be anonymized
- Could ... be anonymized
- No ... anonymization necessary

¹e.g. header fields

Additional Aspects

- "Filter-in" methodology
 - Mark objects to be kept in clear
 - Other objects automatically anonymized
 - Prevents accidental release of sensitive information
- Packet lengths retained
- Header *values* substituted, names retained
- Certain values treated specifically
 - script-tags
 - /etc/passwd

Examples

HTTP header fields:

Referer:	Must
Accept-Language:	Should
User-Agent:	Could
Accept:	No

Examples

HTTP Request-URI:

- Must:** The domain part
- Should:** Until the last two levels of the URI
- Could:** The entire URI

Anonymator, the Prototype

Anonymization schemes:

Weak

... implements Must class

Strong

... implements Must+Should classes

Strongest

... implements Must+Should+Could classes

Customizable

... implements Must class for header fields.
Request-URI, Should and Could headers chosen
by the operator

Example 1

Ethereal screenshot of original packet:

```
Hypertext Transfer Protocol
+ GET /img/background.jpg HTTP/1.1\r\n
  User-Agent: Opera/9.00 (X11; Linux i686; U; en)\r\n
  Host: www.infosikring.no\r\n
  Accept: text/html, application/xml;q=0.9, application/xhtml+xml,
  Accept-Language: no-bok,en;q=0.9\r\n
  Accept-Encoding: deflate, gzip, x-gzip, identity, *;q=0\r\n
  Referer: http://www.infosikring.no/\r\n
  If-Modified-Since: Mon, 27 Feb 2006 14:50:03 GMT\r\n
  If-None-Match: "3b24a8-1f39-d3635cc0"\r\n
  Connection: Keep-Alive, TE\r\n
  TE: deflate, gzip, chunked, identity, trailers\r\n
\r\n
```

Example 2

Ethereal screenshot after *Weak* anonymization:

```
[-] Hypertext Transfer Protocol
[+] GET /img/background.jpg HTTP/1.1\r\n
  User-Agent: opera/9.00 (X11; Linux i686; U; en)\r\n
  Host: hosthosthosthostho\r\n
  Accept: text/html, application/xml;q=0.9, application/xhtml+xml,
  Accept-Language: no-bok,en;q=0.9\r\n
  Accept-Encoding: deflate, gzip, x-gzip, identity, *;q=0\r\n
  Referer: http://www.foofoofoofo.bar\r\n
  If-Modified-Since: Mon, 27 Feb 2006 14:50:03 GMT\r\n
  If-None-Match: "3b24a8-1f39-d3635cc0"\r\n
  Connection: Keep-Alive, TE\r\n
  TE: deflate, gzip, chunked, identity, trailers\r\n
\r\n
```

Example 3

Ethereal screenshot after *Strong* anonymization:

```
[-] Hypertext Transfer Protocol
[+] GET /img/background.jpg HTTP/1.1\r\n
  User-Agent: Opera/9.00 (X11; Linux i686; U; en)\r\n
  Host: hosthosthosthosto\r\n
  Accept: text/html, application/xml;q=0.9, application/xhtml+xml,
  Accept-Language: lllllllllllllllll\r\n
  Accept-Encoding: deflate, gzip, x-gzip, identity, *;q=0\r\n
  Referer: http://www.foofoofoofo.bar\r\n
  If-Modified-Since: Mon, 27 Feb 2006 14:50:03 GMT\r\n
  If-None-Match: ifnonematchifnonematch\r\n
  Connection: Keep-Alive, TE\r\n
  TE: deflate, gzip, chunked, identity, trailers\r\n
  \r\n
```

Example 4

Ethereal screenshot after *Strongest* anonymization:

```
Hypertext Transfer Protocol
+ GET /nnnnnnnnnnnnnnnnnnnn HTTP/1.1\r\n
  User-Agent: browserbrowserbrowserbrowserbrowser\r\n
  Host: hosthosthosthostho\r\n
  Accept: text/html, application/xml;q=0.9, application/xhtml+xml,
  Accept-Language: 111111111111111\r\n
  Accept-Encoding: deflate, gzip, x-gzip, identity, *;q=0\r\n
  Referer: http://www.foofoofoofo.bar\r\n
  If-Modified-since: Mon, 27 Feb 2006 14:50:03 GMT\r\n
  If-None-Match: ifnonematchifnonematch\r\n
  Connection: Keep-Alive, TE\r\n
  TE: deflate, gzip, chunked, identity, trailers\r\n
  \r\n
```

Experiments

- Different levels of anonymization applied to the data set:
 - No anonymization
 - Predefined: *Weak, Strong, Strongest*
 - Customized: Different combinations of Request-URI and header anonymization

Results

- Snort used for counting positives:
 - *Weak* related schemes retain 100% of positives
 - *Strong* related schemes retain about 80% of positives²
 - *Strongest* related schemes retain about 40% of positives²
 - For the current data set, drop in number of positives caused solely by altered Request-URI

²false and true

Conclusions

- Methodology devised and implemented
- *Weak* anonymization retains all positives
not recommended in production systems
- *Strong* anonymization retains 80% of positives
reasonable level of anonymity in most cases
- *Strongest* anonymization retains 40% of positives
highest level of anonymity
- After anonymization most positives must be considered false
 - tradeoff between anonymity and realism

Further Work

- Add more application layer protocols
- Include lower layer protocols for a complete and comprehensive methodology
- Implement added protocols in Anonymator
- Work towards increased number of retained attacks

and now...

Opponent and Questions...