HTTP Header Analysis

Roland Zegers

System and Network Engineering

01 July 2015

UNIVERSITY OF AMSTERDAM

FOX IT
HTTP: used for communication of webtraffic
Headers provide information about the source system, the software and the content that is transferred.
HTTP communication also extensively used by malware.
Exploit Kits: launch platform, easy to use, much options
Research questions

- Is it possible to determine from which source certain HTTP traffic comes, when analyzing and correlating the HTTP header ordering?
- Is it possible to create reliable fingerprints from the analysed results?
- Is it possible to determine if malware is present by analyzing outliers in the HTTP header ordering?
- Can fingerprints be created that match on the outliers?
Figure: HTTP header structure
Method

- Retrieve header order from pcap files from uninfected systems
- Get header order from infections
- Overlay infection headers over uninfected systems
- Calculate probability, uncertainty and occurrence of header order before and after infection
- Match results with unknown samples from Fox-IT
Approach

1. Parse HTTP traffic from pcap to .json format
2. Structure the format
3. Split into separate flows
4. Split into separate request headers (strip other headers)
5. Strip content of Cookie, URI and Referer headers
6. Add line numbers
7. Count line numbers of headers for further calculations

"ua": "Mozilla5.0 (Windows NT 6.3; WOW64; Trident7.0; rv:11.0)"
Used Shannon’s entropy theory to calculate and compare the header position uncertainty of uninfected and infected systems.

**Shannon’s Entropy Theory**

\[
H(X) = - \sum_{i=1}^{n} p_i \log_2(p_i)
\]

<table>
<thead>
<tr>
<th>Systems</th>
<th>Entropy before infection</th>
<th>Entropy after infection</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC1</td>
<td>4,07</td>
<td>4,95</td>
</tr>
<tr>
<td>PC2</td>
<td>4,00</td>
<td>4,87</td>
</tr>
<tr>
<td>PC3</td>
<td>4,19</td>
<td>4,73</td>
</tr>
</tbody>
</table>
Results - Fox-IT systems

<table>
<thead>
<tr>
<th>Header order positions - system-1 Fox-IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>req</td>
</tr>
<tr>
<td>-----</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Header order positions - system-2 Fox-IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>req</td>
</tr>
<tr>
<td>-----</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Header order positions - system-3 Fox-IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>req</td>
</tr>
<tr>
<td>-----</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fox-IT systems</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>System 1</td>
<td>4.98</td>
</tr>
<tr>
<td>System 2</td>
<td>4.45</td>
</tr>
<tr>
<td>System 3</td>
<td>4.60</td>
</tr>
</tbody>
</table>

Roland Zegers
HTTP Header Analysis
Figure: Uninfected headers
Results - example

**Figure:** Infected headers (Fiesta Exploit Kit)
From the header order, profiles (and thus fingerprints) can be created for individual systems.

No distinction between similar systems: cloned systems will have about the same fingerprint.

Some malware will have a distinct profile that can be fingerprinted.

(Re-)Calculating entropy levels can indicate an infection.

Results probably less obvious when using worst-case systems (systems with lots of user-agents or malware with a low disturbance profile).
Future work

- Testing on a larger scale, incorporating worst-case systems and infections
- Developing an automated header order fingerprinting program
Thank you for your attention!
Questions?