HTTP Header Analysis

Roland Zegers

System and Network Engineering

01 July 2015

UNIVERSITY OF AMSTERDAM

FOX IT

Introduction

- 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?

HTTP header structure

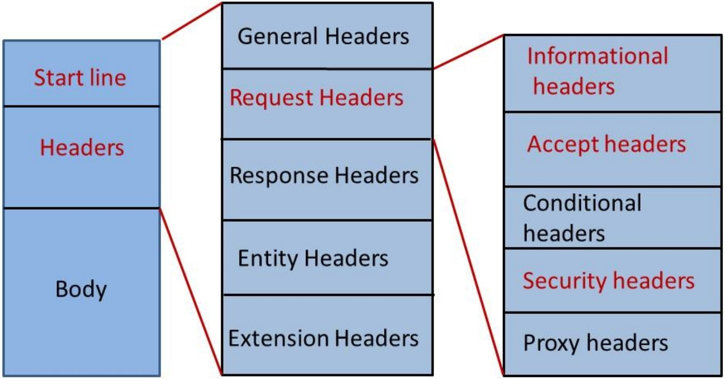


Figure: HTTP header structure

Method

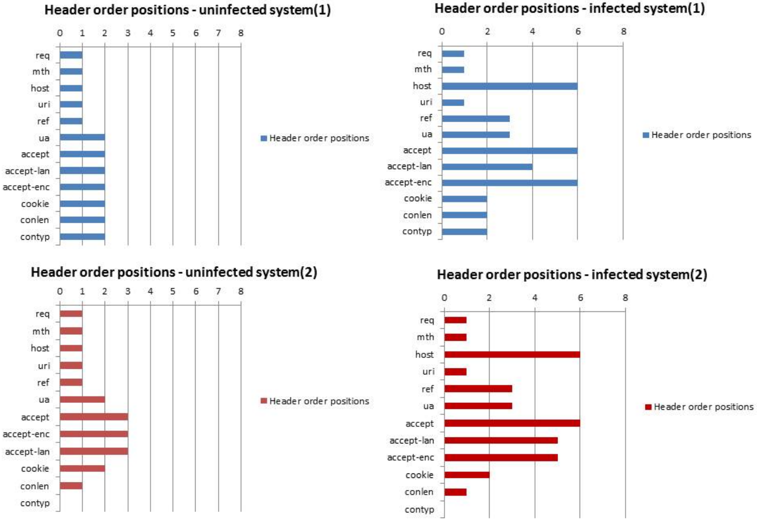
- Retrieve header order from pcap files from uninfected systems
- Get header order from infections
- Overlay infection headers over uninfected systems
- Calculate probability, uncertainty and occurrence of header order before and after infection
- Match results with unknown samples from Fox-IT

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 an Referer headers
6. Add linenumbers
7. Count linenumbers of headers for further calculations

```
"ua": "Mozilla5.0 (Windows NT 6.3; WOW64; Trident7.0; rv:1
```

Results


Header order positions - uninfected system(1)
0 1 2 3 4 5 6 7 8
req
mth
host
uri
ref
ua
accept
accept-lan
accept-enc
cookie
conlen
contyp
Header order positions
Header order positions - infected system(1)
0 2 4 6 8
req
mth
host
uri
ref
ua
accept
accept-lan
accept-enc
cookie
conlen
contyp
Header order positions
Header order positions - uninfected system(2)
0 1 2 3 4 5 6 7 8
req
mth
host
uri
ref
ua
accept
accept-enc
accept-lan
cookie
conlen
contyp
Header order positions
Header order positions - infected system(2)
0 2 4 6 8
req
mth
host
uri
ref
ua
accept
accept-lan
accept-enc
cookie
conlen
contyp
Header order positions

Results - Entropy calculation

Used Shannon's entropy theory to calculate and compare the header position uncertainty of uninfected and infected systems.

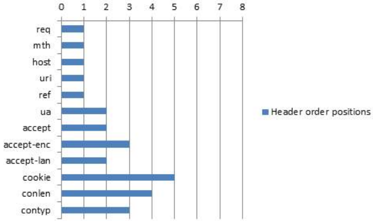
Shannon's Entropy Theory

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

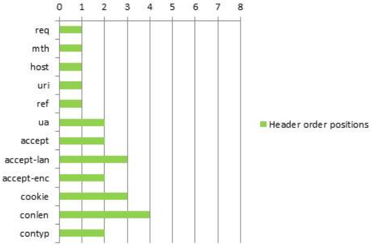
Systems	Entropy before infection	Entropy after infection
PC1	4,07	4,95
PC2	4,00	4,87
PC3	4,19	4,73

Results - Fox-IT systems

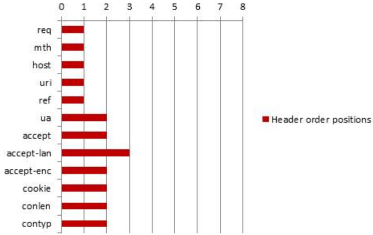
Header order positions - system-1 Fox-IT



Header order positions - system-3 Fox-IT

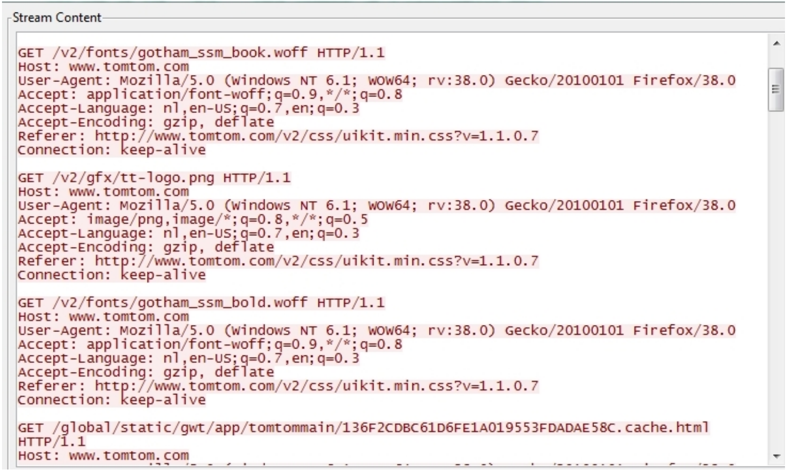


Header order positions - system-2 Fox-IT



Fox-IT systems	Entropy
System 1	4,98
System 2	4,45
System 3	4,60

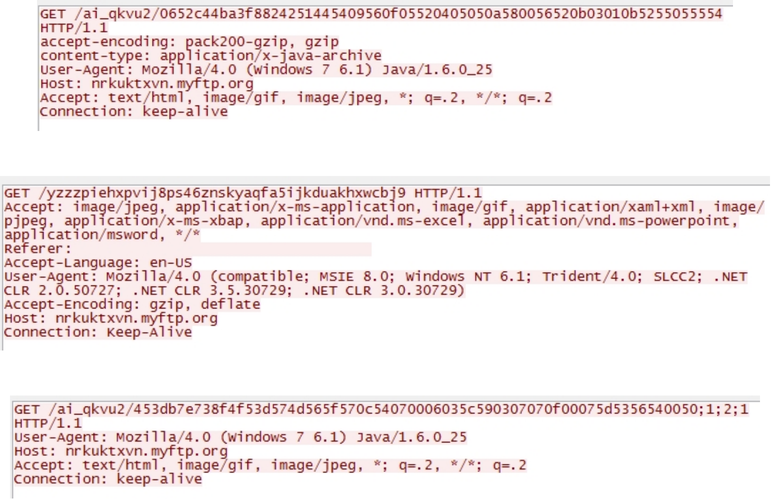
Results - example



```
GET /v2/fonts/gotham_ssm_book.woff HTTP/1.1
Host: www.tomtom.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:38.0) Gecko/20100101 Firefox/38.0
Accept: application/font-woff;q=0.9,*/*;q=0.8
Accept-Language: nl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://www.tomtom.com/v2/css/uikit.min.css?v=1.1.0.7
Connection: keep-alive

GET /v2/gfx/tt-logo.png HTTP/1.1
Host: www.tomtom.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:38.0) Gecko/20100101 Firefox/38.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: nl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://www.tomtom.com/v2/css/uikit.min.css?v=1.1.0.7
Connection: keep-alive

GET /v2/fonts/gotham_ssm_bold.woff HTTP/1.1
Host: www.tomtom.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:38.0) Gecko/20100101 Firefox/38.0
Accept: application/font-woff;q=0.9,*/*;q=0.8
Accept-Language: nl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://www.tomtom.com/v2/css/uikit.min.css?v=1.1.0.7
Connection: keep-alive

GET /global/static/gwt/app/tomtommain/136F2CDBC61D6FE1A019553FDADAE58C.cache.html
HTTP/1.1
Host: www.tomtom.com
```

Figure: Uninfected headers

Results - example



```
GET /ai_qkvu2/0652c44ba3f8824251445409560f05520405050a580056520b03010b5255055554
HTTP/1.1
accept-encoding: pack200-gzip, gzip
content-type: application/x-java-archive
User-Agent: Mozilla/4.0 (Windows 7 6.1) Java/1.6.0_25
Host: nrkuktxvn.myftp.org
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
```

```
GET /yzzzpiehxpvij8ps46znskyaqfa5ijkduakhxwcbj9 HTTP/1.1
Accept: image/jpeg, application/x-ms-application, image/gif, application/xaml+xml, image/
pjpeg, application/x-ms-xbap, application/vnd.ms-excel, application/vnd.ms-powerpoint,
application/msword, */*
Referer:
Accept-Language: en-US
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET
CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729)
Accept-Encoding: gzip, deflate
Host: nrkuktxvn.myftp.org
Connection: Keep-Alive
```

```
GET /ai_qkvu2/453db7e738f4f53d574d565f570c54070006035c590307070f00075d5356540050;1;2;1
HTTP/1.1
User-Agent: Mozilla/4.0 (Windows 7 6.1) Java/1.6.0_25
Host: nrkuktxvn.myftp.org
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
```

Figure: Infected headers (Fiesta Exploit Kit)

Conclusion

- 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 a automated header order fingerprinting program

End

Thank you for your attention!
Questions?