



Network Monitoring using MMT:

An application based on the User-Agent field in HTTP headers

Vinh Hoa LA †

Raul FUENTES †

PhD Student

Prof. Ana CAVALLI †‡

Supervisor

† Telecom SudParis, IMT

‡ Montimage France





IDOLE project

■ IDOLE:

- 3-year French project on “Investigation and Operated Detection in Large Scale”
- Passive tools of detection, high-speed correlation, and investigation after incidents.
- Started since late 2014





Contents

- **Motivation**
- **Network Monitoring**
 - **Montimage Monitoring Tool (MMT)**
- **User-Agent field case study**
 - **Problem statement**
 - **Methodology**
- **Experimental results**
- **Discussions**
- **Conclusion & perspectives**



Motivation

■ Network monitoring by examining metadata

- Metadata: data about data, an abstract (**structural/descriptive**) of data, a piece of data...
- Example: A book ~ data
A library ~ data

The position of the book in the library (which room, which shelf) ~ metadata

■ IMT's role in IDOLE project: Advanced monitoring techniques for detection and investigation using metadata.




■ Why metadata?

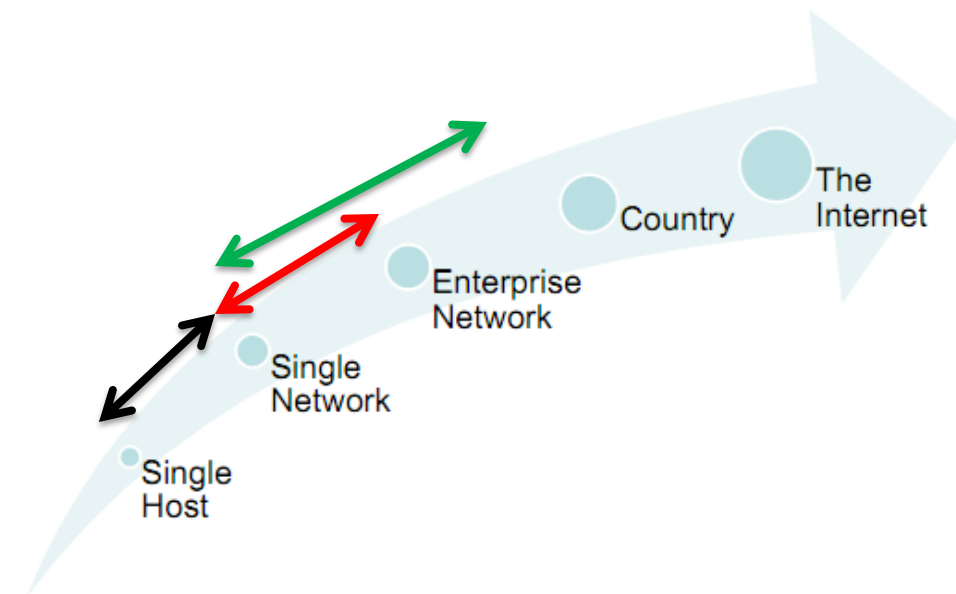
- Velocity

■ First step: Monitoring using User- Agent Field in HTTP's headers?



Network Monitoring

- **The range of Network Monitoring:**
- **Full Packet Capture:** 
 - Capture “everything” that goes across the network
 - Typically used on a single network
 - Example: PCAP
- **Meta Data Capture:** 
 - Capture data associated with a particular network activity
 - Typically in the form of logs
 - Examples:
 - For email traffic capture: from, to, subject, date, attachments
 - For web traffic capture: Source IP, destination IP, URL, User Agent String
- **NetFlow:** 
 - NetFlow aggregates related packets into unidirectional flows
 - The flow records are collected and stored for later analysis
 - Examples: SiLK , Argus





Network Monitoring Using MMT (Montimage Monitoring Tool)

MMT-Extract

- C library.
- Enable the extraction of wanted-attributes (protocol fields, application exchanged messages or logs...).

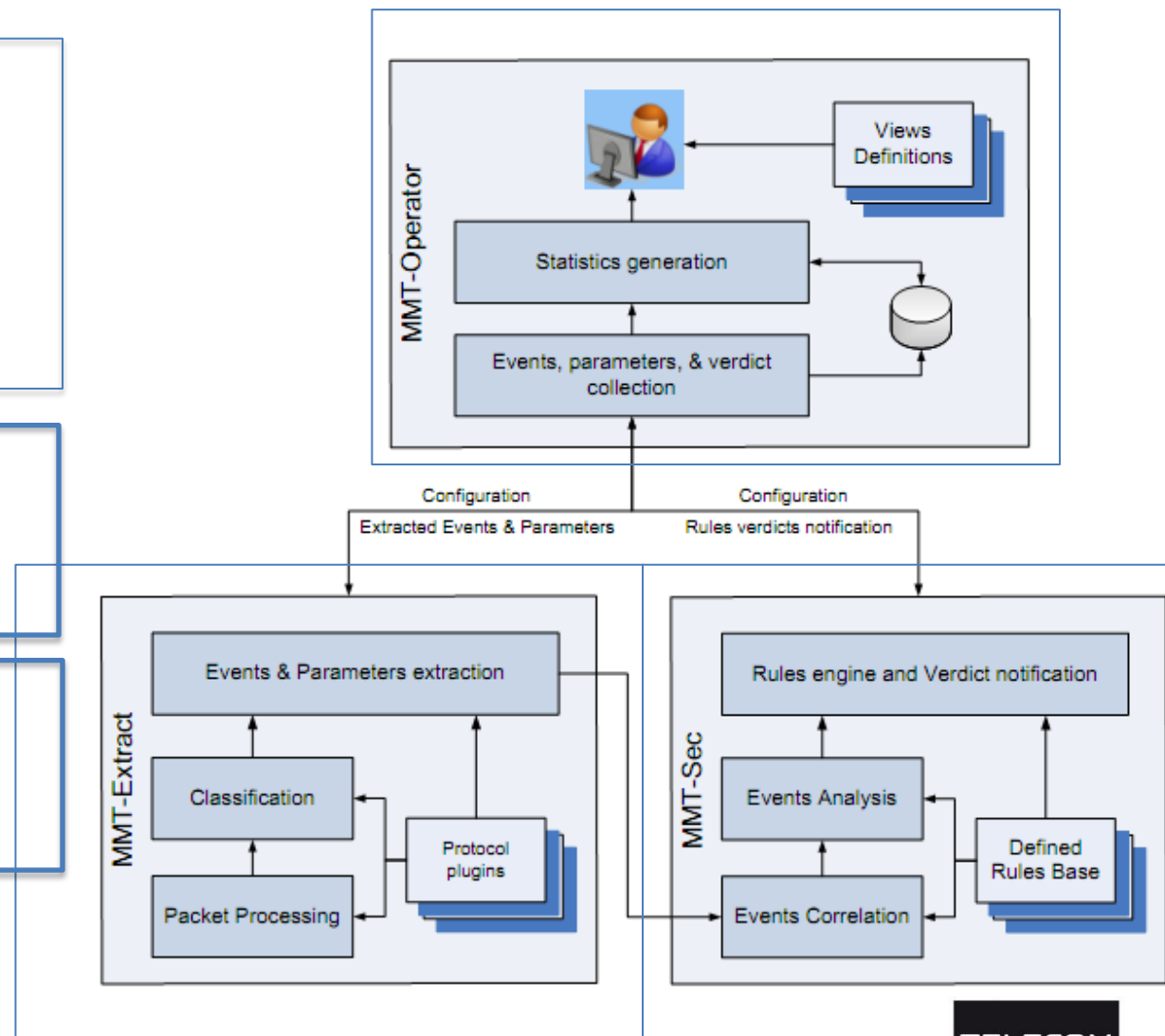
MMT-Sec

- Security rules written in XML referring to both expected and unexpected behaviors.

MMT-Operator

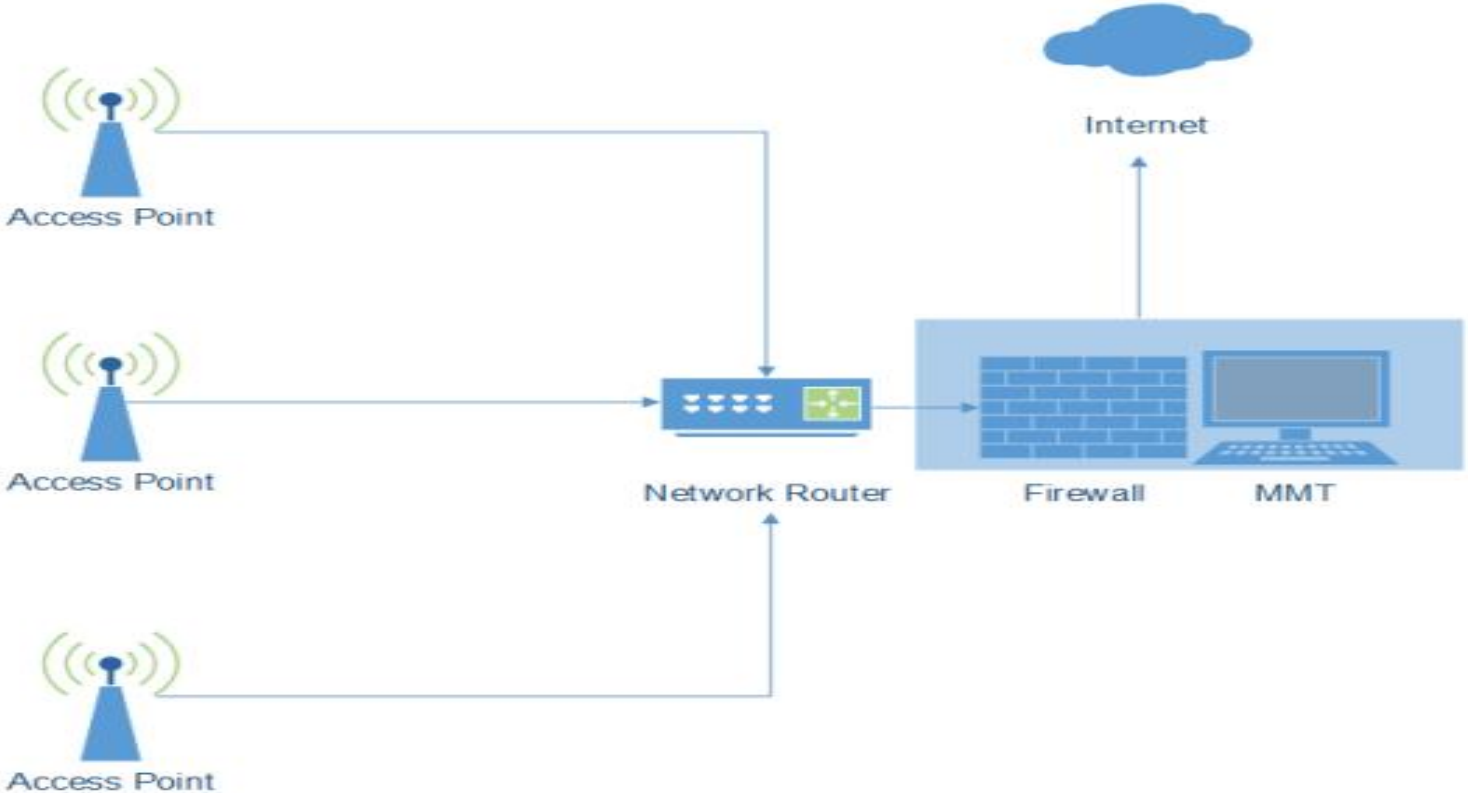
- Allow a customizable graphical user interface to display the result (still under development)

MMT is a DPI tool able to run in real time or with traces files.





Network Monitoring Using MMT (Montimage Monitoring Tool)

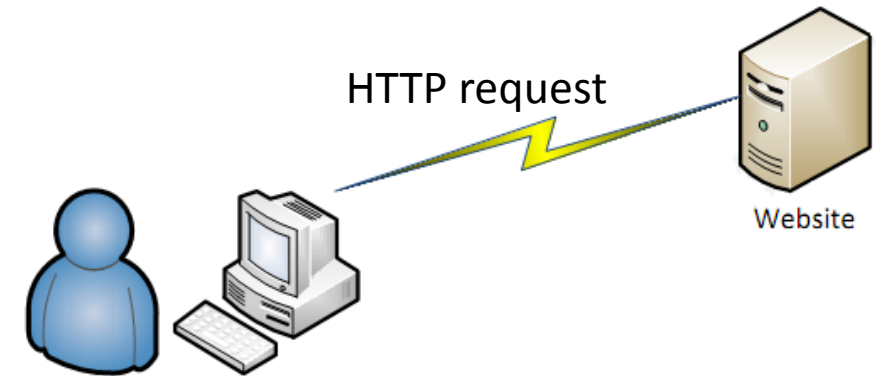


MMT's position to listen to live traffic



User-Agent field case study: Problem statement

- ❑ What is “user agent field”?
 - Statistical purposes
 - The tracing of protocol violations
 - Automated recognition of user agents for the sake of tailoring responses.
- ❑ Example of a HTTP header:



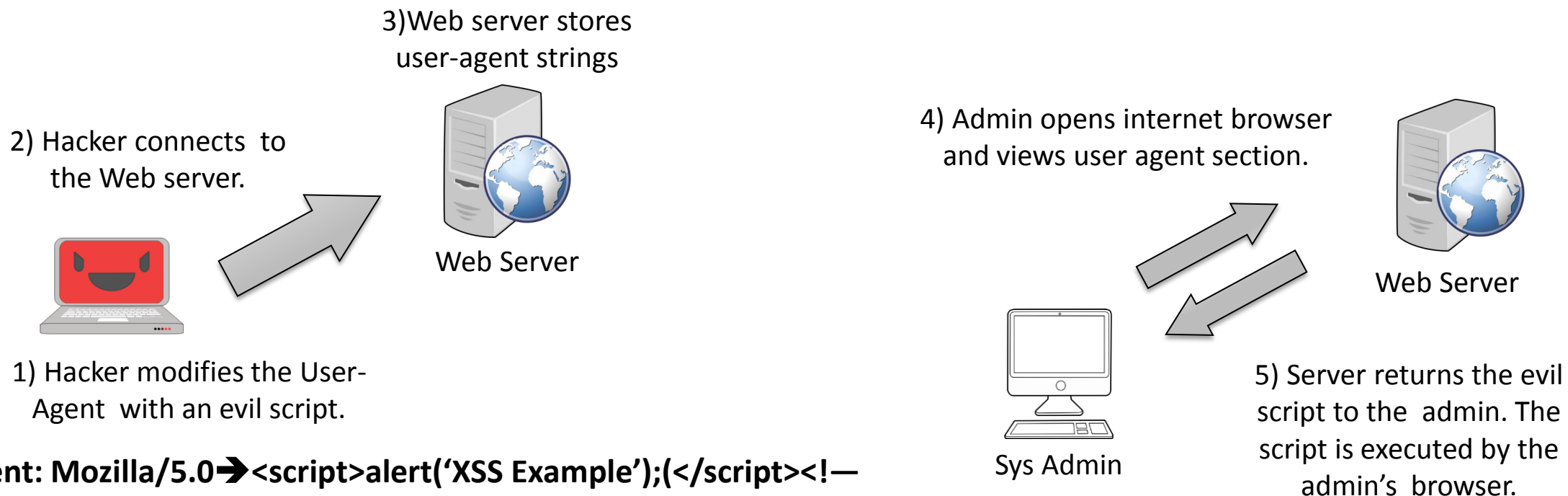
```
GET / HTTP/1.1
Accept: image/jpeg, application/x-ms-application, image/gif, application/xaml+xml,
image/png, application/x-ms-xbap, application/vnd.ms-excel, application/vnd.ms-
powerpoint, application/msword, */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; windows NT 6.1; Trident/4.0;
SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC
6.0; InfoPath.3)
Accept-Encoding: gzip, deflate
Host: www.sans.edu
Connection: Keep-Alive
```

User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0;
SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center
PC 6.0; InfoPath.3)



User-Agent field case study: Problem statement

■ Stored and Reflected XSS (cross-site scripting)

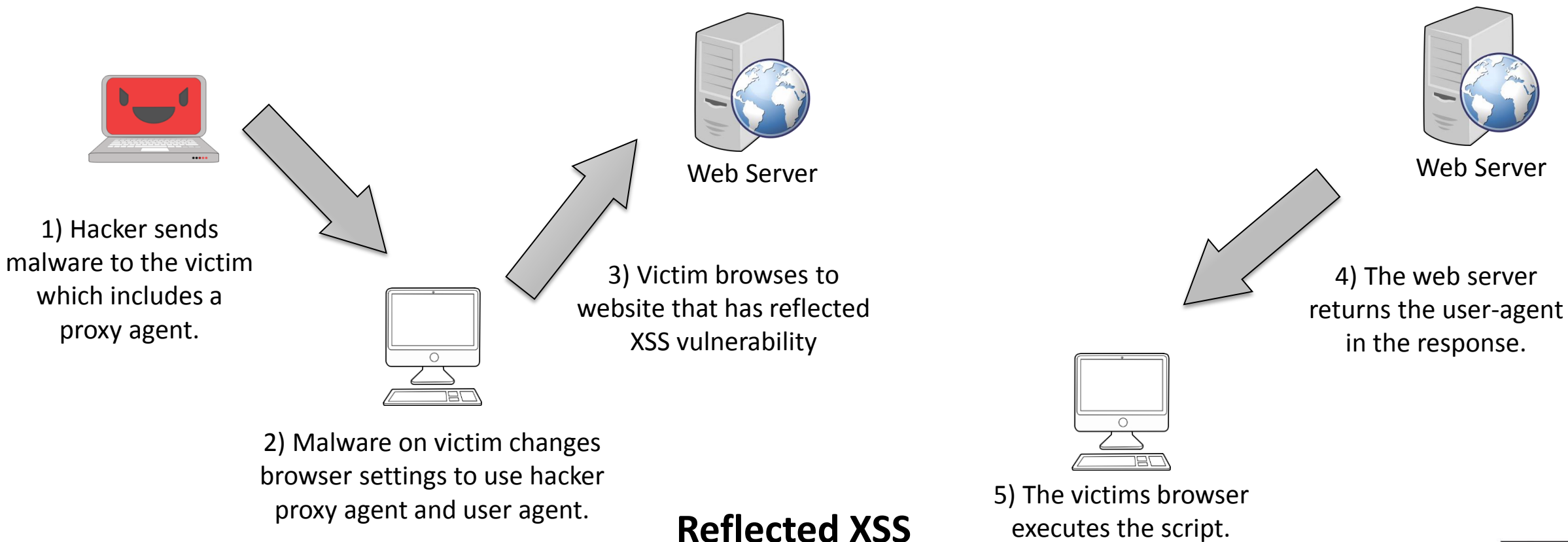


Stored XSS



User-Agent field case study: Problem statement

■ Stored and Reflected XSS (cross-site scripting)





User-Agent field case study: Problem statement

■ SQL injection via user agent field

Example 1



Web Server



Database server

1) Hackers creates a manual http request with an SQL injection in the user agent field.

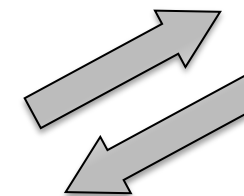
2) Web analytics collects user agent fields for marketing.

3) Database reads user agent data and executes SQL injection.

Example 2



1) Hacker modifies user agent to include an SQL query, ""

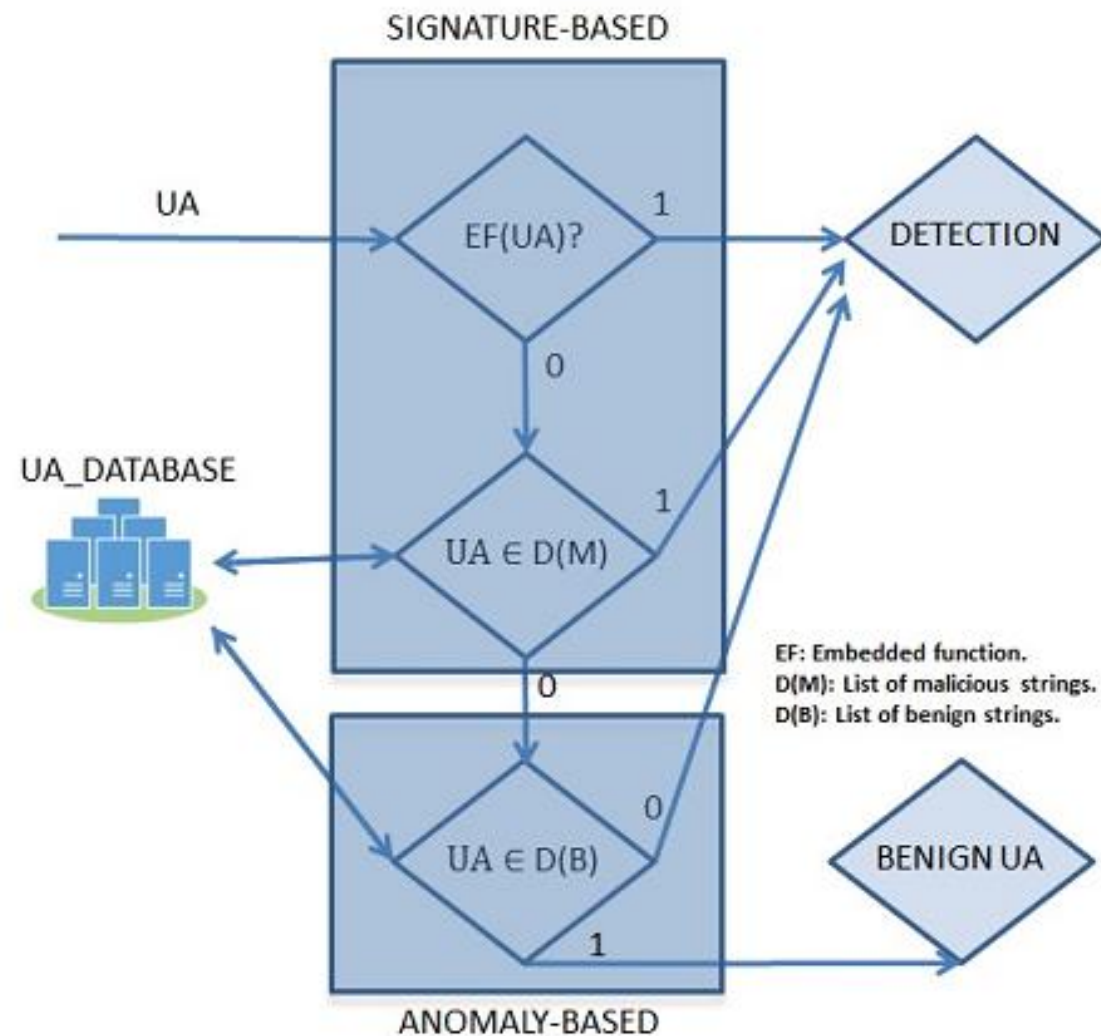
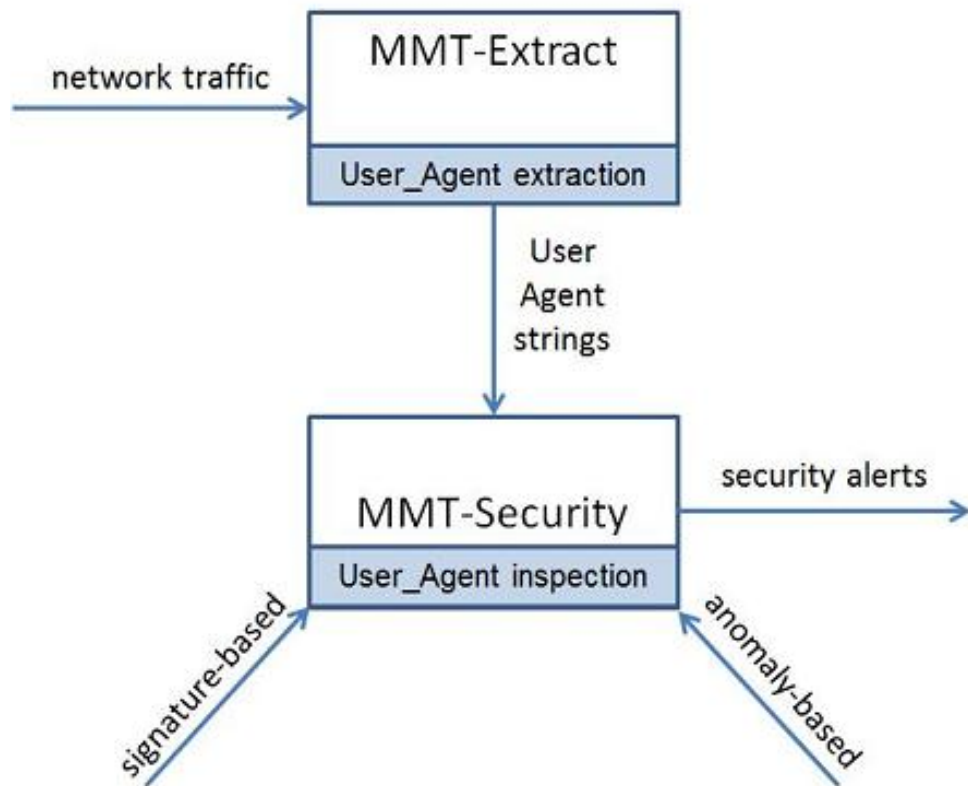


Web Server

2) Server returns an SQL error in its response page.



User-Agent field case study: Methodology





Experimental results

■ Experiments with offline traffic:

- Input: PCAP files
- Ex1 (Tab.I): **Rather small traffic.**
 - PCAP files contain different malware traffic within normal one (214 036 HTTP GET packets).
 - The packet loss rate is calculated as follow: $packet_loss_rate = \frac{number_of_packets_lost}{number_of_packets_sent}$
 - We noticed not only the deficiencies of SNORT in terms of detection but also a slight dominance of MMT regarding extraction issue.
 - Reason: SNORT utilizes only rules identifying blacklisted User-Agent strings, in other words, only a signature-based technique. Therefore, SNORT is incapable against new abnormal behavior.

	MMT	SNORT
Number of packets	214036	214036
Number of extractions	213978	213794
Packet loss rate	0.03%	0.11%
Number of detections	83209	585

Table I

MMT AND SNORT IN CASE OF OFFLINE TRAFFIC



Experimental results

■ Experiments with offline traffic:

- Ex2 (Tab.II): **Huge traffic**

- Input: a data-set consists of 80 files PCAP containing 83,850,638 packets with total volume of 39.2 GB.

- Only read and extract

- In the first five tests, we ran MMT, SNORT

and TCPdump all alone (limited in maximum parallel programs that could consume CPU/RAM resource or network bandwidth)

- In later five tests, we ran several applications at the same time.

Test N^0	MMT [s]	SNORT [s]	TCPdump [s]
1	807	1010	858
2	835	1004	862
3	743	1219	862
4	783	1006	860
5	720	1003	863
6	739	1005	2181
7	758	1143	2227
8	730	1283	2013
9	740	1307	2574
10	807	1212	2304
Average	766.2	1119.2	1638.4
Processing rate (Mbps)	419	287	196

Table II
EXECUTION TIME AND PROCESSING RATE OF MMT, SNORT AND
TCPDUMP IN READING PCAP FILES



Experimental results

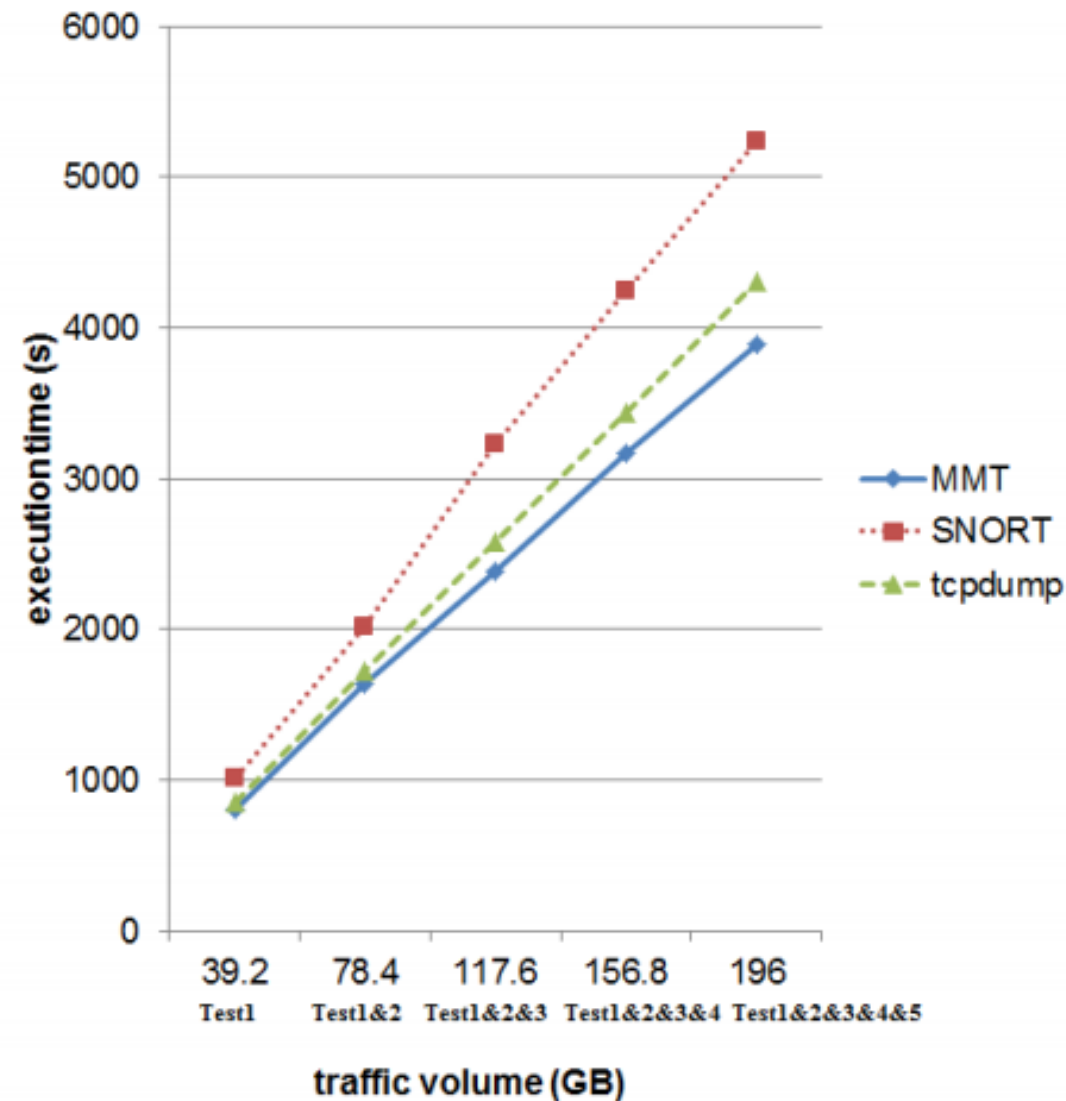
■ Experiments with offline traffic:

- Ex2 (cont):

	MMT	SNORT	TCPdump
CPU usage	3.4%	4.5%	6%
Memory consumption	12.8%	13.3%	13%

Table III

AVERAGE RESOURCE CONSUMPTION OF MMT, SNORT AND TCPDUMP



Execution time of MMT, SNORT and TCPdump in function of traffic volume



Experimental results

■ Experiments with live traffic:

- Ex3 (Tab.IV): **Automatically**

- A simple C application that enables reading normal/abnormal User-Agent strings prepared in a text file and passing the HTTP requests containing them to a web-server.

Test N ^o	SQL injection		DoS		Random UA		Known malicious UA	
	MMT [ms]	SNORT [ms]	MMT [ms]	SNORT [ms]	MMT [ms]	SNORT [ms]	MMT [ms]	SNORT [ms]
Test 1	0.901	–	0.735	–	0.868	–	0.776	0.920
Test 2	0.790	–	0.655	–	0.773	–	0.938	0.939
Test 3	0.700	–	0.555	–	0.704	–	0.881	0.942
Test 4	0.590	–	0.443	–	0.645	–	1.118	0.967
Test 5	0.482	–	0.192	–	0.988	–	1.116	–
Test 6	0.334	–	0.109	–	0.934	–	1.117	0.927
Test 7	0.167	–	0.978	–	0.870	–	1.052	0.959
Test 8	1.002	–	0.874	–	1.109	–	0.851	0.989
Test 9	0.895	–	0.783	–	1.136	–	0.944	0.993
Test 10	0.810	–	0.695	–	1.142	–	0.906	–
Average	0.667	–	0.602	–	0.917	–	0.970	0.955

Table IV

DETECTION LATENCY OF MMT AND SNORT



Experimental results

■ Experiments with live traffic:

- Ex4 (Tab.V): **Manually**
 - Mozilla Firefox's Add-on named TAMPER DATA is used to edit manually the User-Agent field and thus, to generate malicious HTTP requests.

	MMT	SNORT
Number of extractions	212	212
Number of detections	40	8
False positive	0	0
False negative	0	32

Table V

FALSE POSITIVE AND FALSE NEGATIVE OF OUR SOLUTION AND SNORT



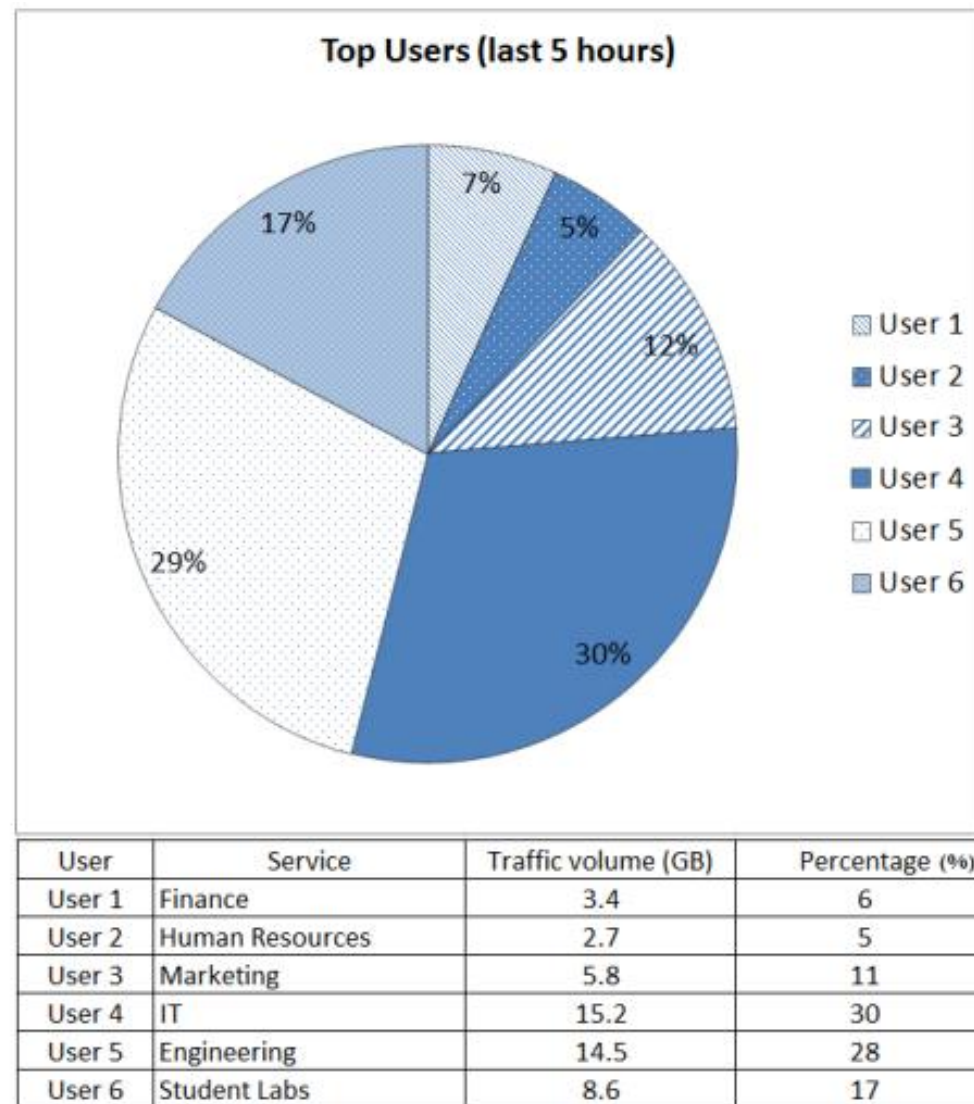
Discussions

■ MMT's strength:

- Heterogeneous intrusion detection approach
- High-speed extraction and real-time detection
- Attribute extraction and legal problems

■ MMT is more than a network security monitor:

- In the presented case study, we concentrate only on security issues. In practice, MMT can also monitor user activities and troubleshoot the network.





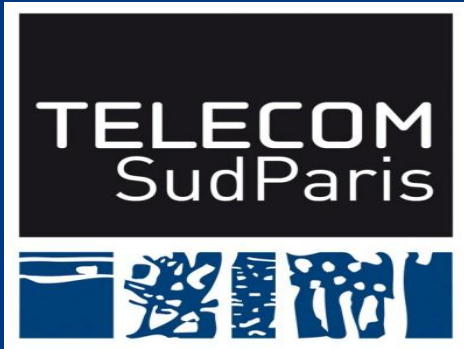
Conclusion & perspectives

- **MMT is an extensible and flexible monitoring tool:**
 - Applicable as a real-time automated detection of malicious User-Agent strings
 - Applicable for large scale networks, not limited in security but even for other network issues.
- **Our detection approach covers two kinds of threats:**
 - attacks in which attackers modify intentionally the User-Agent field in order to perform their evil intention (e.g., SQL injection, Stored and Reflected XSS, and DoS)
 - malicious traffic corresponding to suspicious threats (e.g., malware, botnets or virus) generated intentionally or unintentionally by infected users or proxies.



Conclusion & perspectives

- **Detecting a malicious User-Agent string is NOT ENOUGH to determine a harmful user agent.**
 - A good starting point of network traffic inspection.
 - The related IP address and/or domain, payload data sent and received by this host and other correlated hosts should be investigated.
- **Perspectives:**
 - Broaden our research over total HTTP headers including other field (e.g., cookies) as well as other network protocols (e.g., SMTP).
 - Correlate different rules and extractions in order to detect more complicated intrusions or attacks (e.g., heart-bleed bug, botnets, etc.)



Thank you!

