Network Security Testing using MMT:

A case study in IDOLE project

Vinh Hoa LA
PhD Student

Telecom SudParis
Institut Mines Telecom
France

Prof. Ana CAVALLI
Supevisor

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 2014

CASSIDIAN CYBERSECURITY

OPPIDA
EXPERT EN SÉCURITÉ
DES SYSTÈMES D'INFORMATION

SkyRecon®
Intelligent Client Security

INSTITUT
Mines-Télécom

QOSMOS

Bull

TELECOM
SudParis

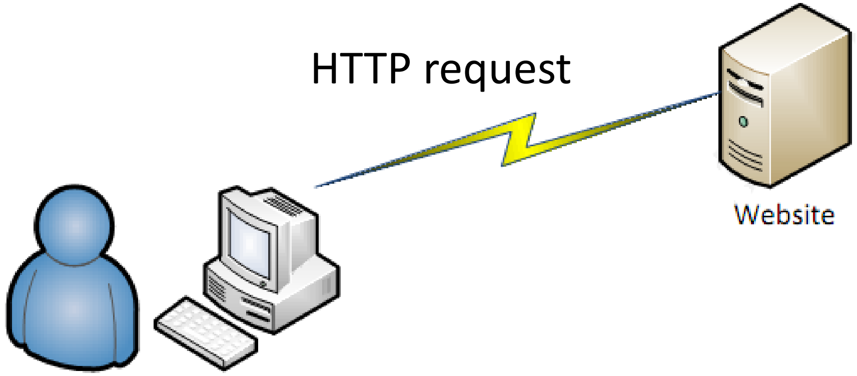
Motivation

- **Network monitoring by examining metadata**
  - Metadata: data about data, an abstract (**structural/descriptive)** of data, a piece of data...
  - Example: A book ~ data

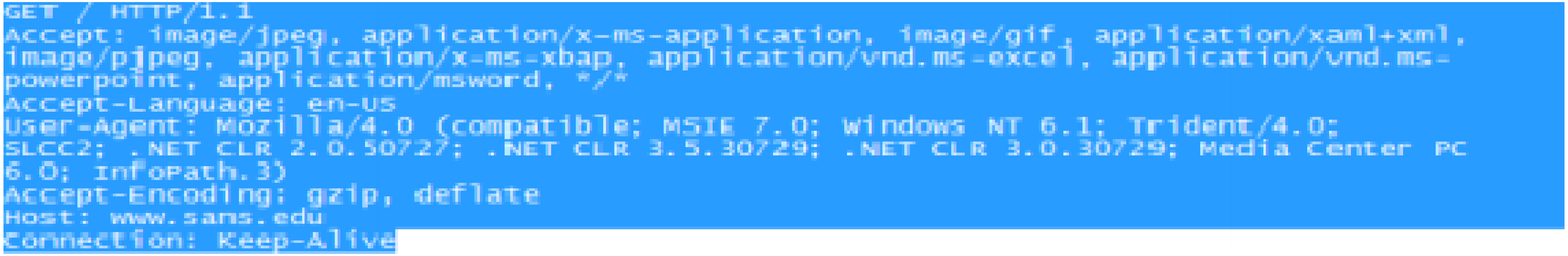
    A library ~ data

    The position of the book in the library (which room, which shelf) ~ metadata
- **IMT's role:** Advanced monitoring techniques for detection and investigation using metadata.
- **Why metadata?**
  - Velocity
- **First step: Monitoring using User- Agent Field in HTTP's headers?**

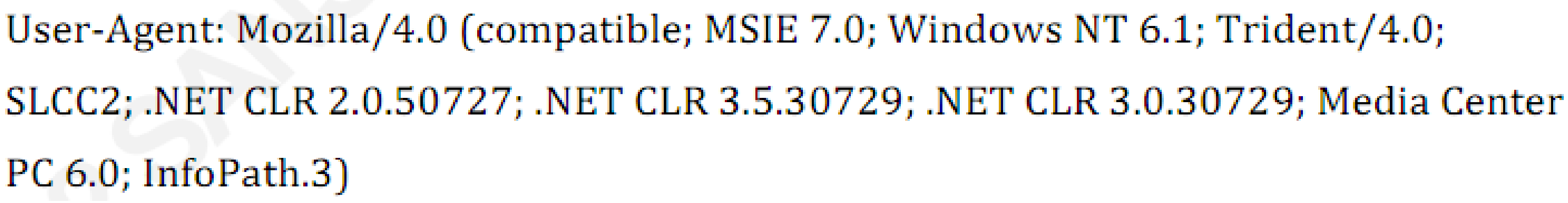
Metadata: User -Agent field



- ❑ **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:**



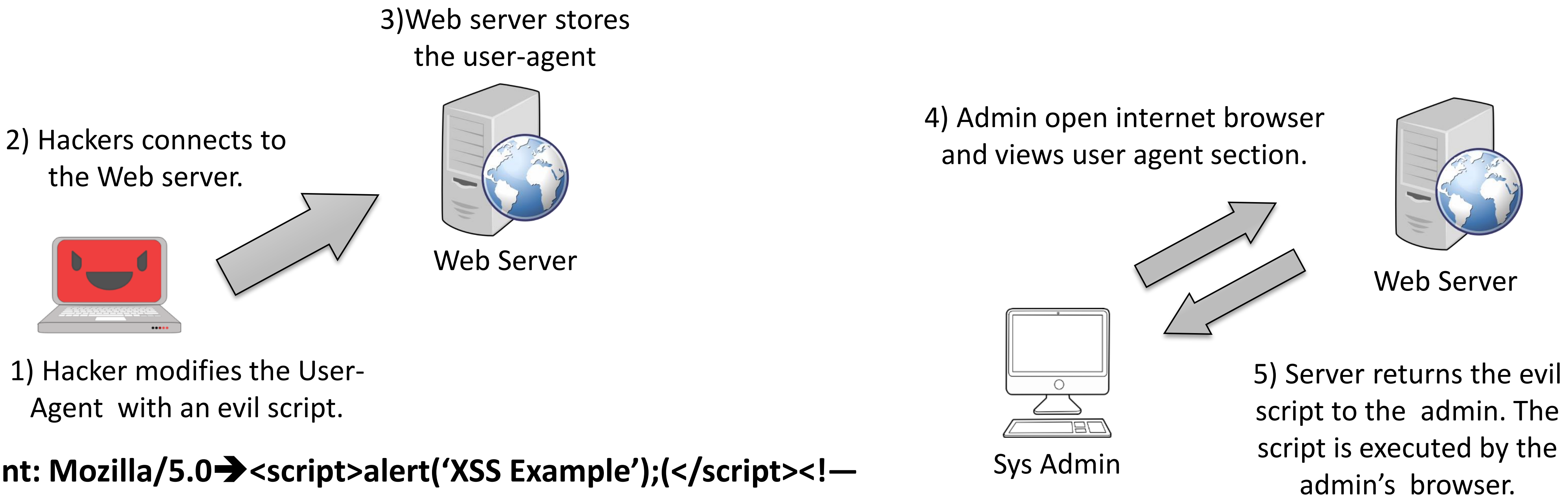
```
GET / HTTP/1.1
Accept: image/jpeg, application/x-ms-application, image/gif, application/xaml+xml,
image/pjpeg, application/x-ms-xbap, application/vnd.ms-excel, application/vnd.ms-
powerpoint, application/msword, */*
Accept-Language: en-US
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 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)

Vulnerabilities based on user-agent-field (1)

- **Stored and Reflected XSS (cross-site scripting)**



1) Hacker modifies the User-Agent with an evil script.

2) Hackers connects to the Web server.

3)Web server stores the user-agent

4) Admin open internet browser and views user agent section.

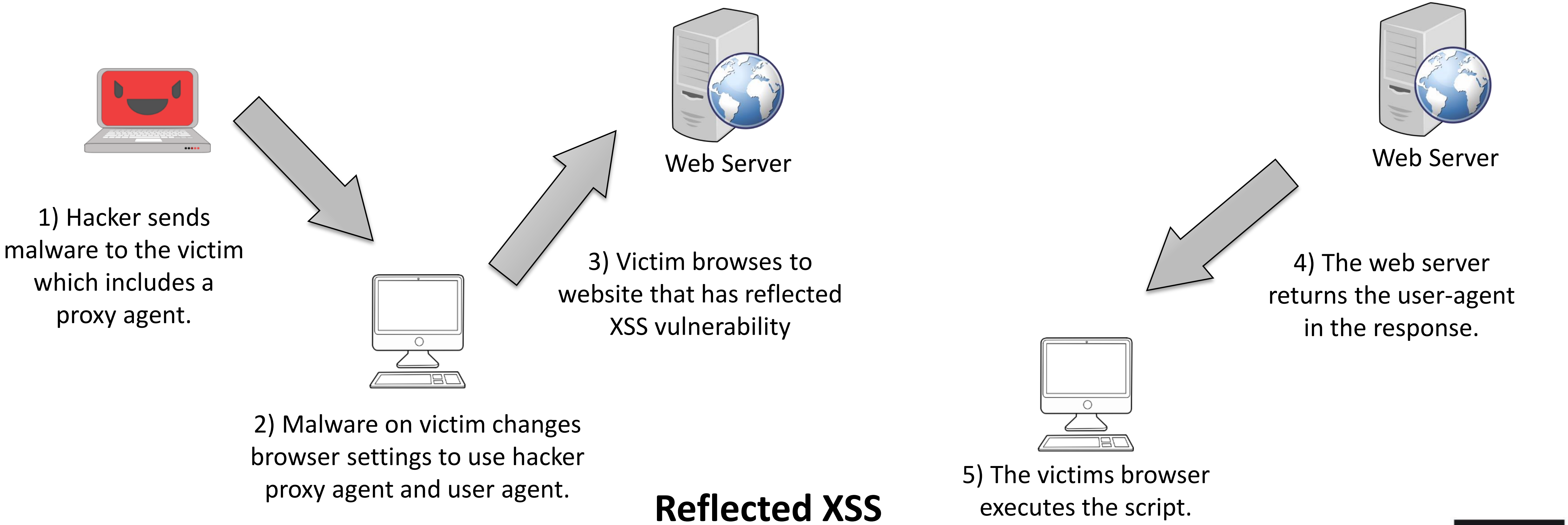
5) Server returns the evil script to the admin. The script is executed by the admin’s browser.

User-agent: Mozilla/5.0➔<script>alert(‘XSS Example’);(</script><!—

Stored XSS

Vulnerabilities using user-agent-field (2)

- **Stored and Reflected XSS (cross-site scripting)**



1) Hacker sends malware to the victim which includes a proxy agent.

2) Malware on victim changes browser settings to use hacker proxy agent and user agent.

3) Victim browses to website that has reflected XSS vulnerability

Web Server

4) The web server returns the user-agent in the response.

Web Server

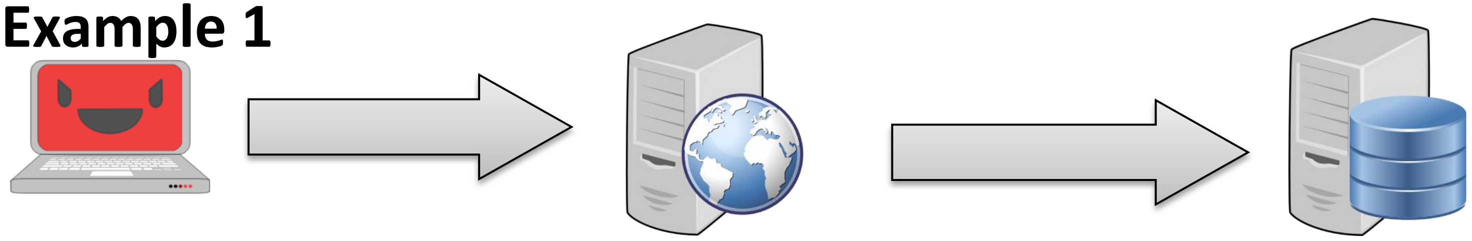
5) The victims browser executes the script.

Reflected XSS

Vulnerabilities using user-agent-field (3)

- SQL injection via user agent field

Example 1



Web Server

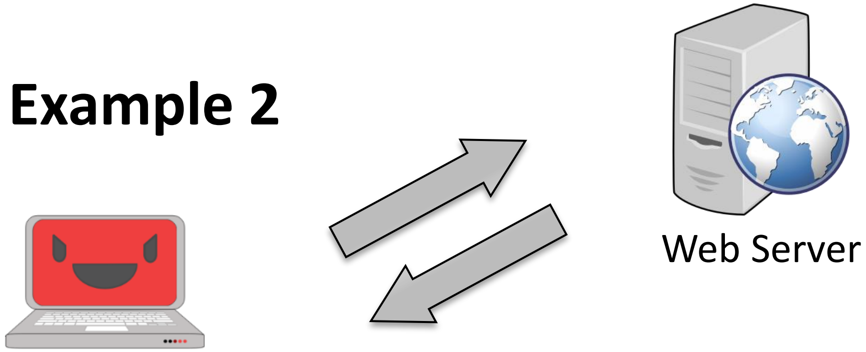
Database server

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



Web Server

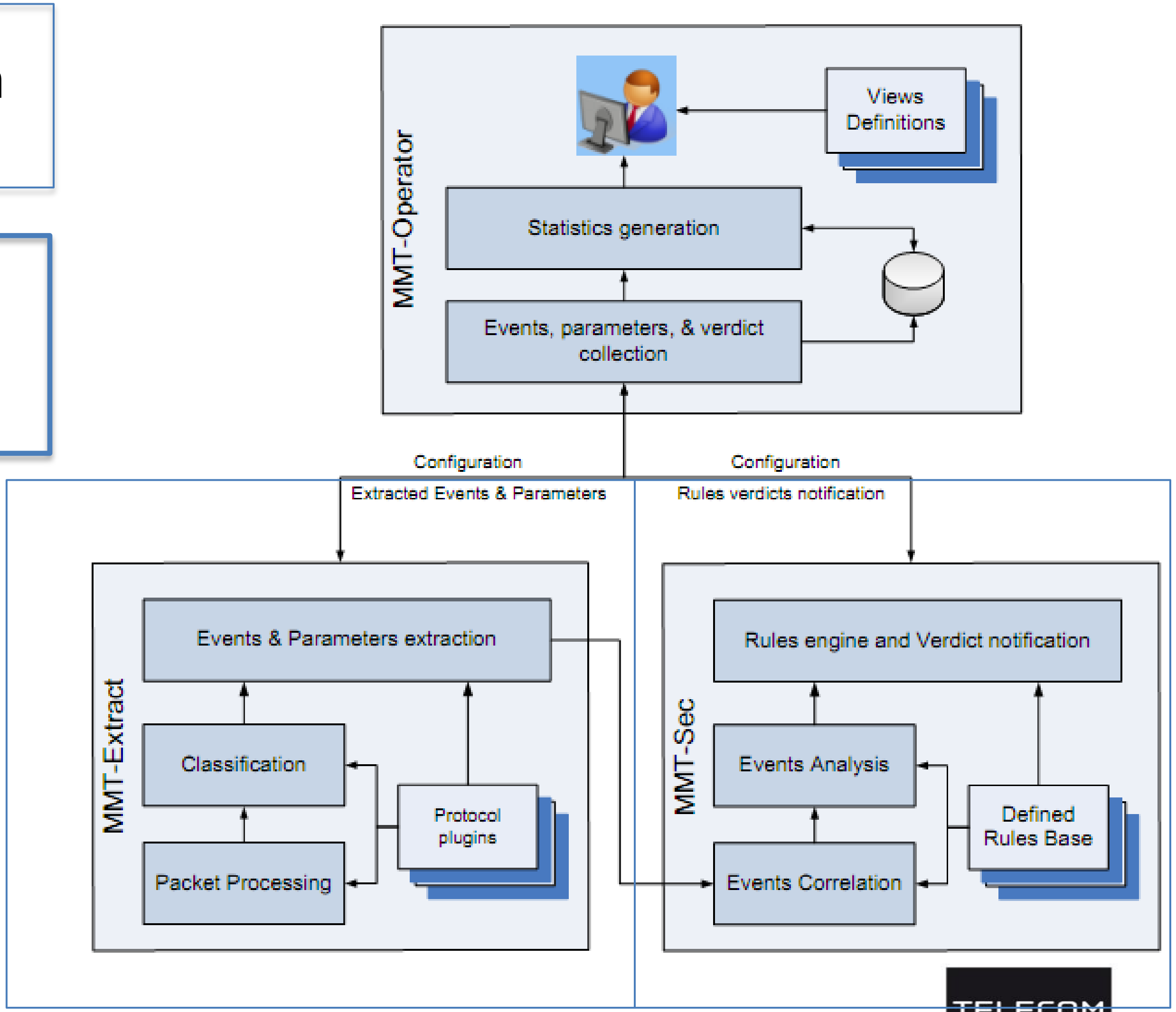
1) Hacker modifies user agent to include an SQL query, “”

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

Using MMT to detect vulnerabilities based on User Agent Field (1)

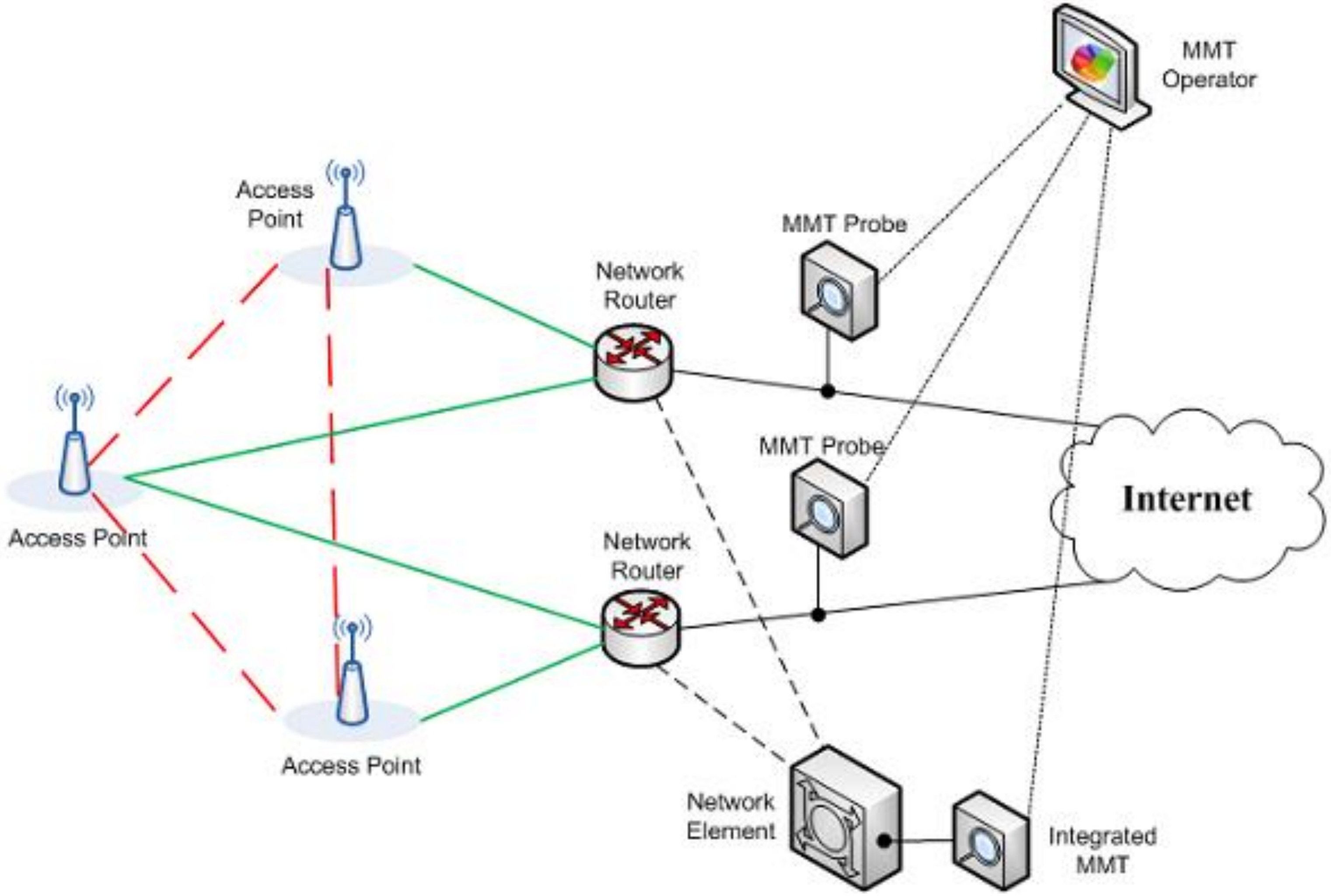
MMT-Extract: Extract the User Agent Fields from HTTP requests.

MMT-Sec: Define the rules to detect HTML, SQL and other malicious scripting code in User Agent Fields.

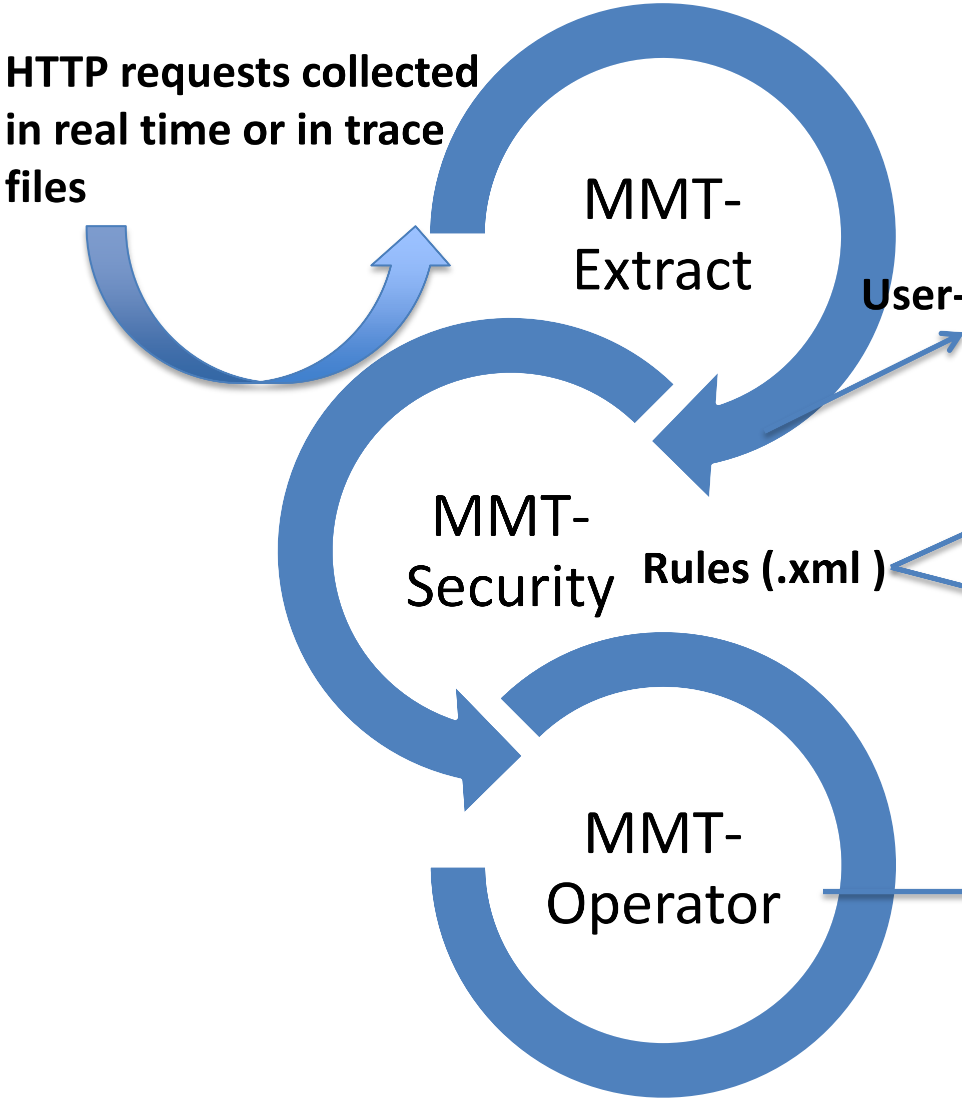


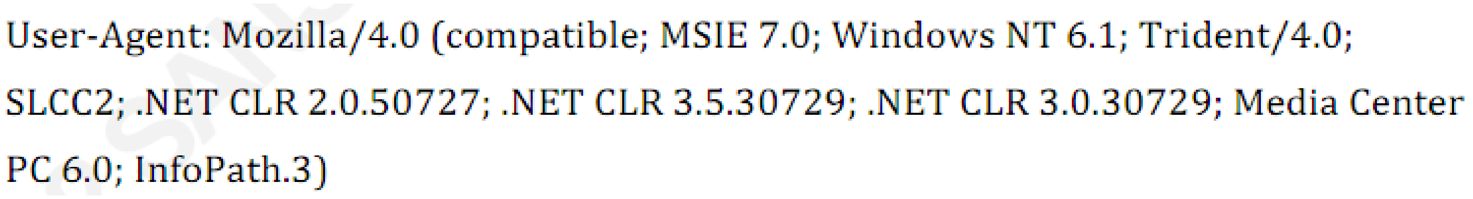
MMT is a DPI tool able to run in real time or with traces files.

Using MMT to detect vulnerabilities based on User Agent Field (2)



Using MMT to detect vulnerabilities based on User Agent Field (3)




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 Fields

- expected behaviour of the application or protocol under-test
- malicious behaviour: an attack, a vulnerability or a misbehaviour

Using MMT to detect vulnerabilities based on User Agent Field (4)

- **MMT's strength:**
  - MMT properties: Rules can describe both wanted and unwanted behavior of application or protocol under-test.
  - MMT allows combining active and passive approaches.
  - MMT allows combining centralized and distributed analysis to detect 0-day attacks.

- **Concerns to be considered:**
  - Possibility of the passage to large scale.
  - Possibility to correlate with other rules and extractions to detect more complicate intrusions or attacks (e.g., heartbleed bug, BYOD- Bring Your Own Device, Botnet…)

Thank you!