How to Break Web Application Security

Erwin Geirnaert
Director European Operations
Security Innovation
egeirnaert@securityinnovation.com
+32478289466

OWASP

Copyright © 2004 - The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License.

Agenda

- Security Test Checklist
- Threat Modeling
- Tools
- Some examples

Security Test Checklist

- You need an EXPERIENCED TESTER
- Create a threat model and a test plan
- Web application testing <> penetration testing
- Do not rely ONLY on automated web application security scanners
- Source code of the web application HELPS
- Have a Security Tester Toolbox
- Log everything

Threat modeling

“You cannot test a system until you understand the threats”

Threat modeling is the design activity to discover the threats that your application is susceptible to.

Threat modeling yields both threats and vulnerabilities and provides ways to perform security testing in order to prioritize the security fixes needed.

Threat modeling - Definitions

Threats are possible attacks. Vulnerabilities are security related software errors:

- A *threat* is what an attacker might try to do to an asset or through an entry point
- A *vulnerability* is a specific security exploit due to an unmitigated threat path

Threat modeling - STRIDE

Threats can be classified using the STRIDE classification:

- **S**poofing – lying about identity
- **T**ampering – Destroying data
- **R**epudiation – Cleaning the steps of an attack/Denying a transaction
- **I**nformation Disclosure – Stealing valuable private data
- **D**enial of Service – Stopping an application from providing its basic functionality
- **E**scalation of Privileges – Executing code with stolen high privileges

- Whenever discovering threats the analyst will always think about STRIDE elements

Threat modeling – Example of threats

Some threats for Web Video Recoding System

- Attacker **tampers** with central video storage
- Attacker **sends** malicious input to overrun the video recording client
- Attacker **deletes** temporary recordings
- Attacker remotely **executes** code in Video web service box

Threat modeling - DREAD

DREAD:

- Damage potential – what's the extent of the damage if this vulnerability was to be exploited
- Reproducibility – how well can the finder reproduce the issue
- Exploitability – difficulty of taking advantage of the flaw for malicious purpose
- Affected users – how many or what type of users are affected by the flaw
- Discoverability – how fast can it be publicly be discovered

- DREAD is used to analyze the risk of discovered **vulnerabilities**

Attack vectors for web applications

- Parameter Tampering
- Cookie Tampering
- Cross-site Scripting
- SQL Injection
- Script Injection
- Command Injection
- Encoding Attacks
- Buffer Overflows
- Format-string attacks
- Harvesting User IDs
- Brute-forcing Accounts
- Path Truncation Attacks
- Hidden Path Discovery
- Application Directory and File Mapping
- Forceful Browsing
- Source Code Disclosure
- Web server vulnerability exploitation

Security Tester Toolbox

- Tools are just a way of manipulating web applications
- They are no silver bullet, a lot of false positives can be the result of automated scan
- They can be really expensive
- They can be useful
- You need to learn how to use them and what the limitations are
- Internet Explorer can do the job and for free ☺

Tools in the past

- 4 years ago, a limited list of free tools:
  - Achilles: local proxy
  - @Stake WebProxy: local proxy& fuzzer, in Java ☺
  - WebSleuth: plugin for IE, raw requests
  - Whisker: vulnerability scanner
  - Nikto: vulnerability scanner
  - Nessus: didn't include web vulnerabilities yet

  - But they did the job, only it required more time....

Commercial Fault Injection Test Tools

1. SPI Dynamics WebInspect
2. Sanctum now Watchfire AppScan
3. Kavado Scando
4. AppSecInc AppDetective for Web Apps
5. Cenzic Hailstorm
6. Security Innovation Holodeck
7. NT Objectives NTOSpider
8. Acunetix Web Vulnerability Scanner 2
9. Compuware DevPartner Fault Simulator
10. Fortify Pen Testing Team Tool
11. @stake Web Proxy 2.0
12. Burp Intruder
13. Sandsprite Web Sleuth
14. MaxPatrol 7
15. Syhunt Sandcat Scanner & Miner
16. TrustSecurityConsulting HTTPExplorer
17. Ecyware BlueGreen Inspector
18. NGS Typhon
19. Parasoft WebKing (more QA-type tool)

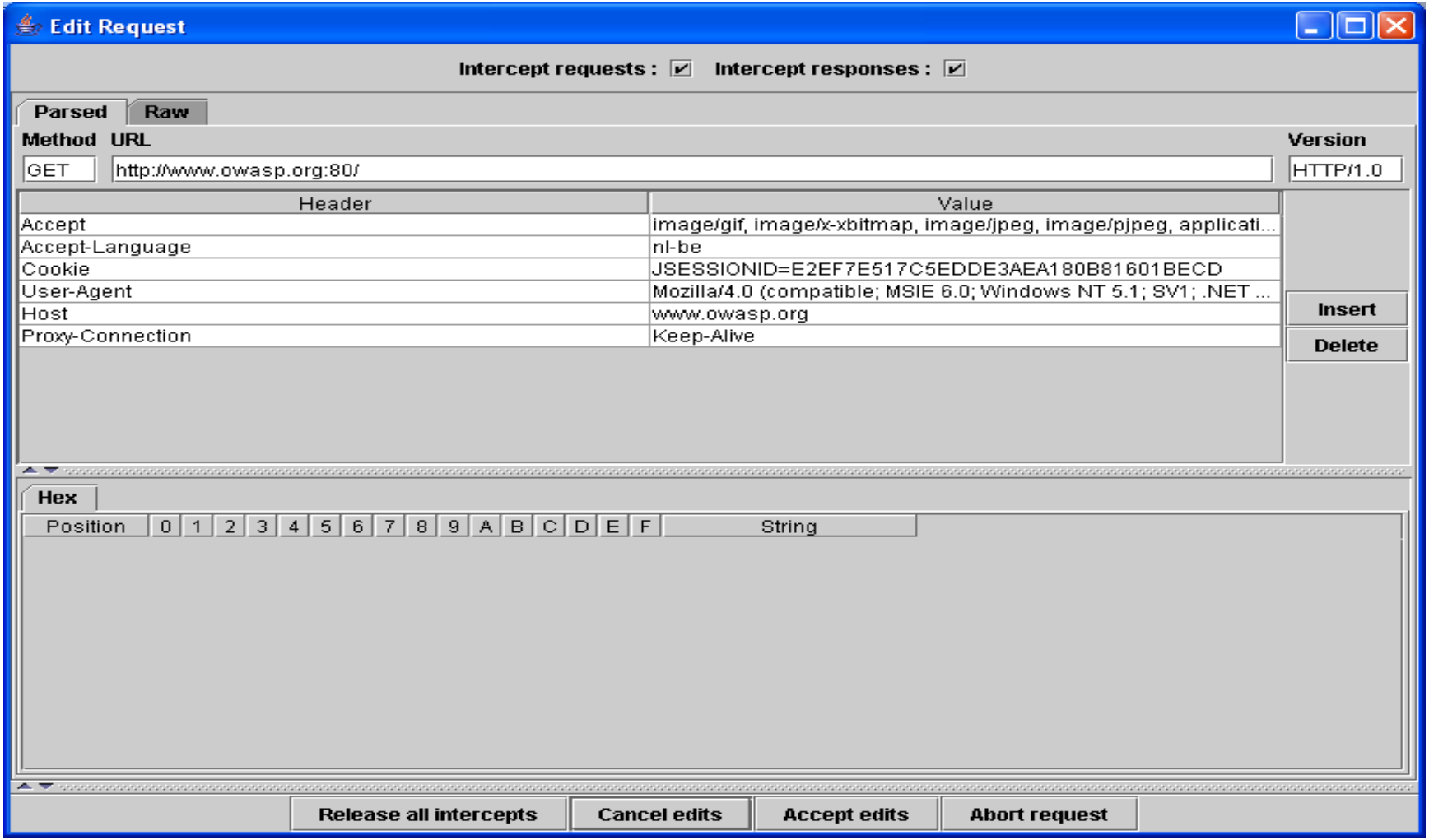
Open Source or Freeware Fault Injection Test Tools

1. WebScarab (HTTPush, Exodus)
2. Paros Proxy
3. Burp Spider
4. Burp Proxy
5. SPIKE Proxy
6. SPIKE
7. Achilles Proxy
8. Odysseus Proxy
9. Webstretch Proxy
10. Absinthe 1.1 (formerly SQLSqueal)
11. NGS SQL Injection Inference Tool (BH Europe 2005)
12. Internet Explorer HTMLBar Plugin
13. Firefox LiveHTTPHeaders and Developer Tools
14. Sensepost Wikto (Google cached fault-finding)
15. Foundstone Sitedigger (Google cached fault-finding)

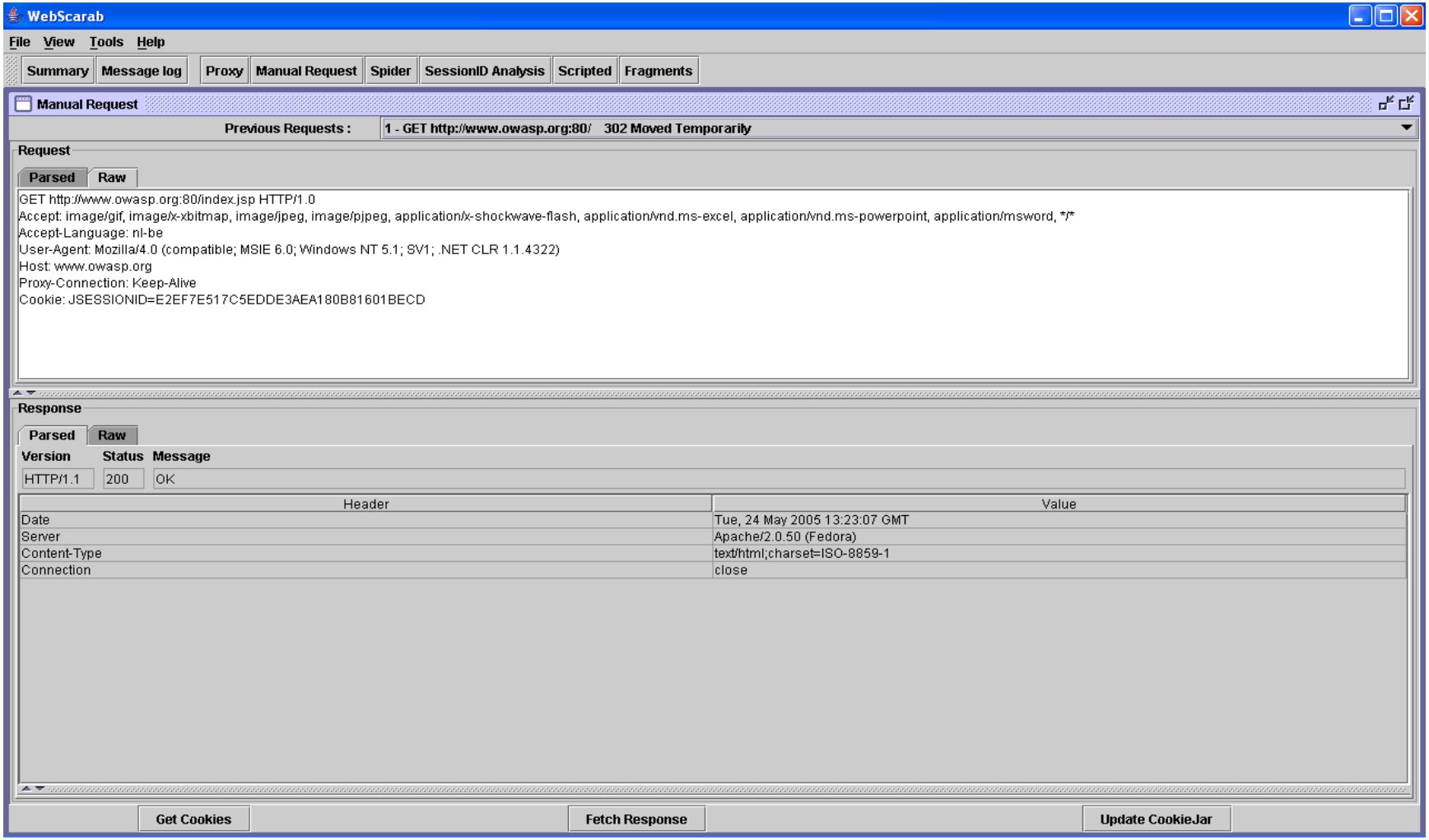
OWASP - WebScarab

- Java based: download stand-alone JAR and runtime
- HTTP Proxy
- Client-certificates
- Session analysis
- Raw request
- Spider
- Custum plugins: BeanShell

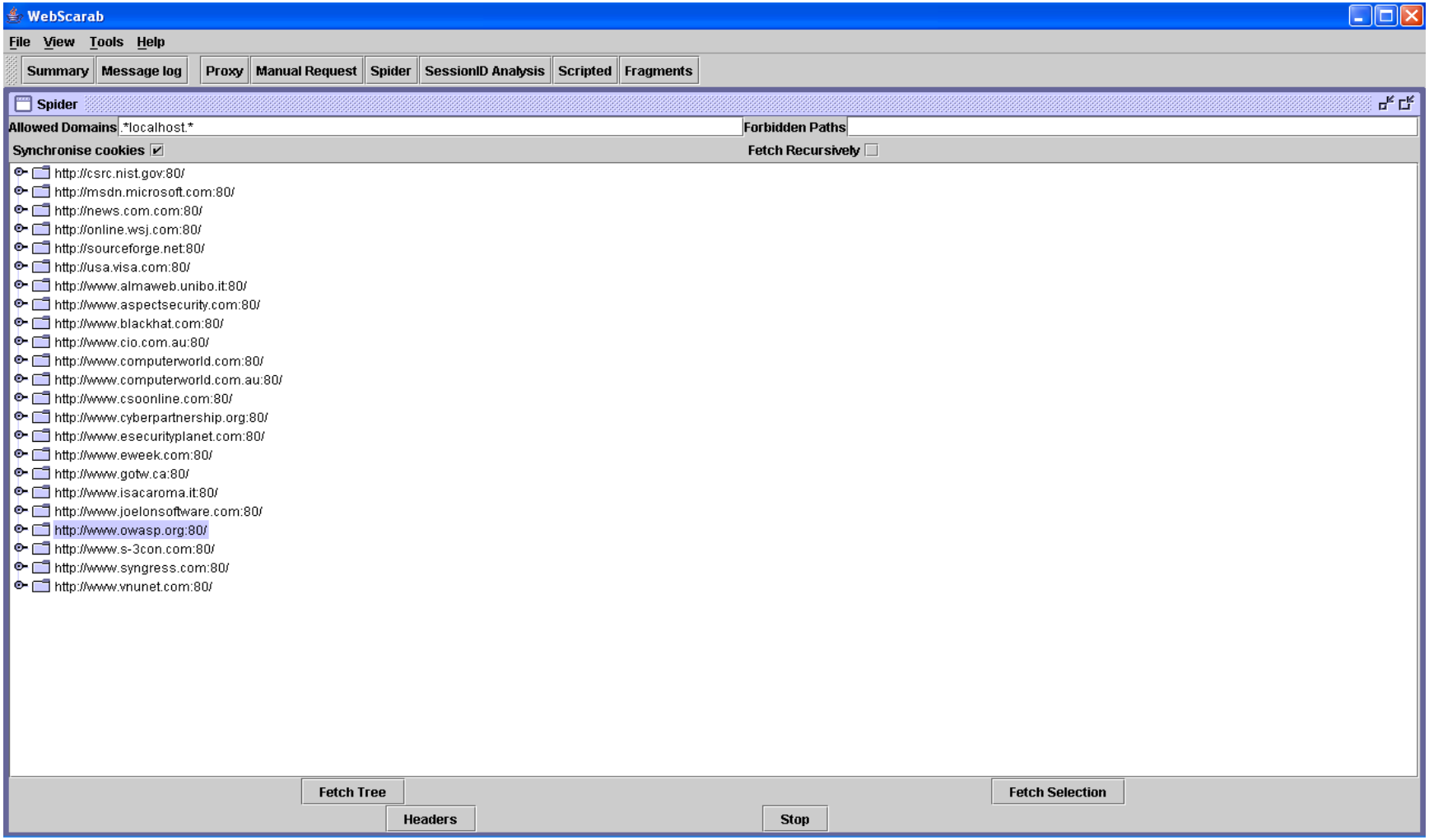
OWASP – WebScarab - Interceptor



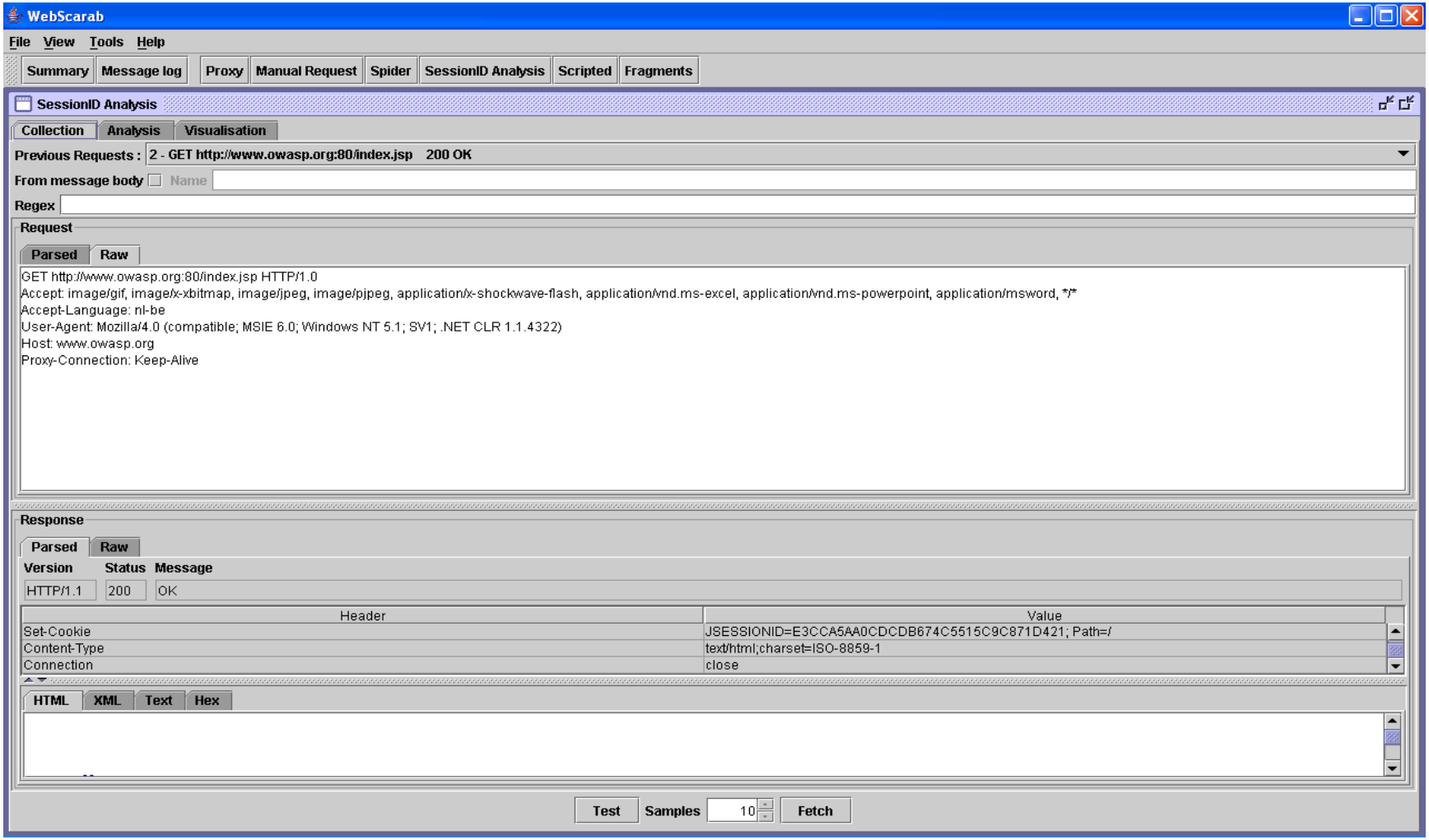
OWASP – WebScarab – Raw Request



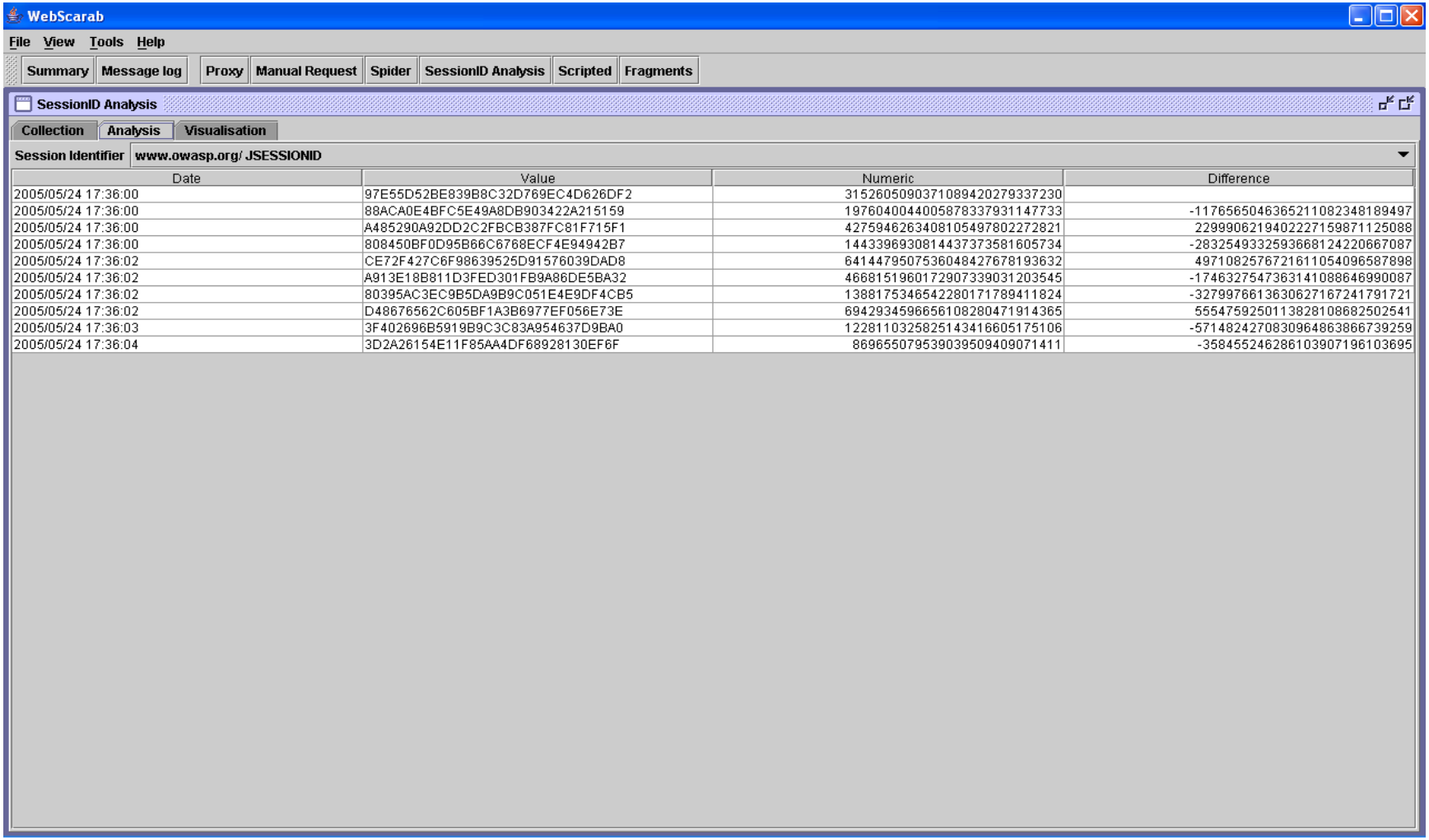
OWASP – WebScarab - Spider



OWASP – WebScarab – SessionID Analysis

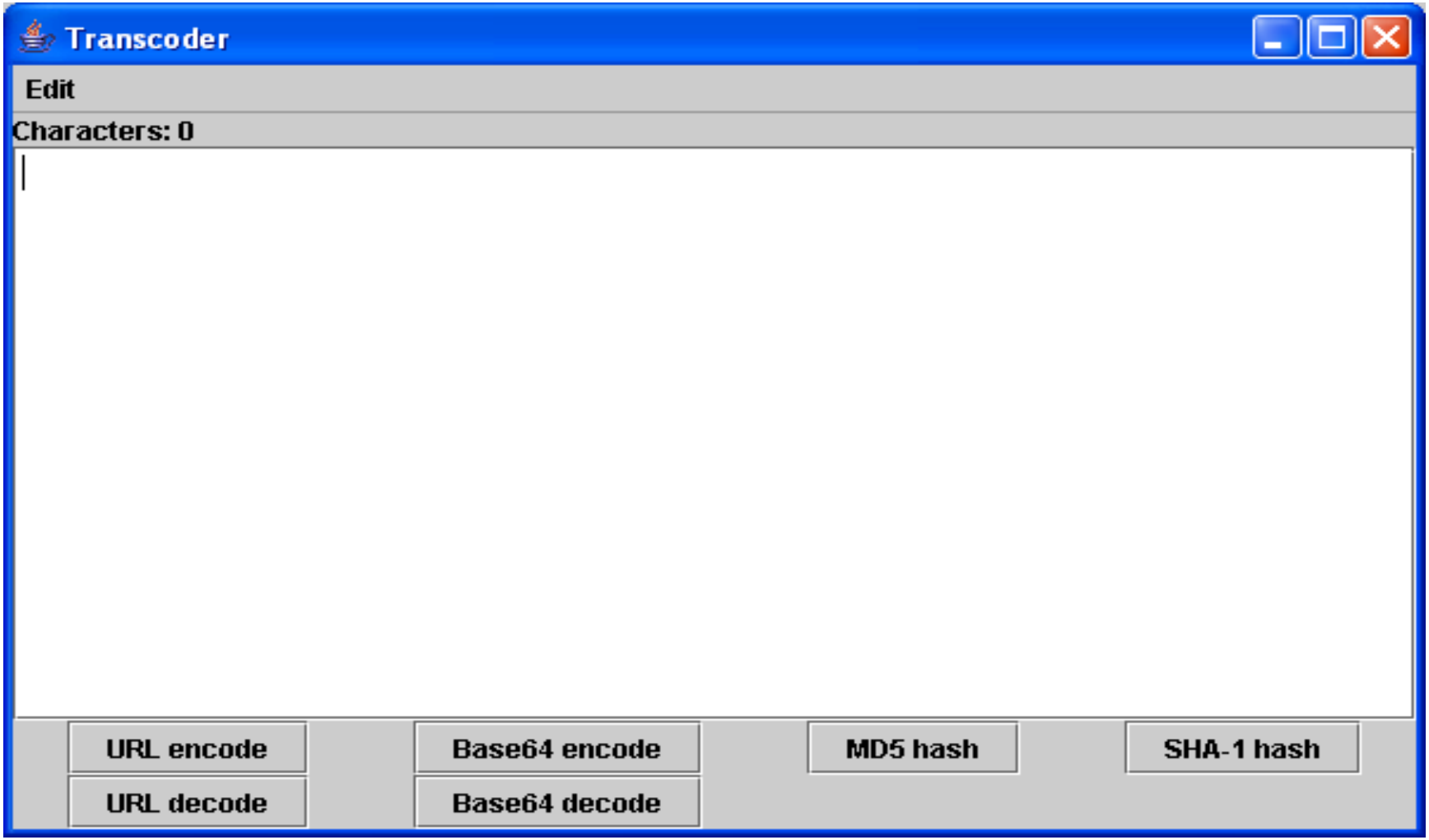


OWASP – WebScarab – SessionID Analysis



Date	Value	Numeric	Difference
2005/05/24 17:36:00	97E55D52BE839B8C32D769EC4D626DF2	315260509037108942027933723O	
2005/05/24 17:36:00	88ACA0E4BFC5E49A8DB903422A215159	197604004400587833793114773 3	-11765650463652110823481894 97
2005/05/24 17:36:00	A485290A92DD2C2FBCB387FC81F715F1	427594626340810549780227282 1	22999062194022271598711250 88
2005/05/24 17:36:00	808450BF0D95B66C6768ECF4E94942B7	144339693081443737358160573 4	-28325493325936681242206670 87
2005/05/24 17:36:02	CE72F427C6F98639525D91576039DAD8	641447950753604842767819363 2	49710825767216110540965878 98
2005/05/24 17:36:02	A913E18B811D3FED301FB9A86DE5BA32	466815196017290733903120354 5	-17463275473631410886469900 87
2005/05/24 17:36:02	80395AC3EC9B5DA9B9C051E4E9DF4CB5	138817534654228017178941182 4	-32799766136306271672417917 21
2005/05/24 17:36:02	D48676562C605BF1A3B6977EF056E73E	694293459665610828047191436 5	55547592501138281086825025 41
2005/05/24 17:36:03	3F402696B5919B9C3C83A954637D9BA0	122811032582514341660517510 6	-57148242708309648638667392 59
2005/05/24 17:36:04	3D2A26154E11F85AA4DF68928130EF6F	869655079539039509409071411	-35845524628610390719610369 5

OWASP – WebScarab – Transcoder



Some examples

- Parameter tampering
- Cross-site-scripting
- Hidden fields
- SQL Injection
- Error messages
- Google ☺

That’s it…

- Any Questions?

Thank you!