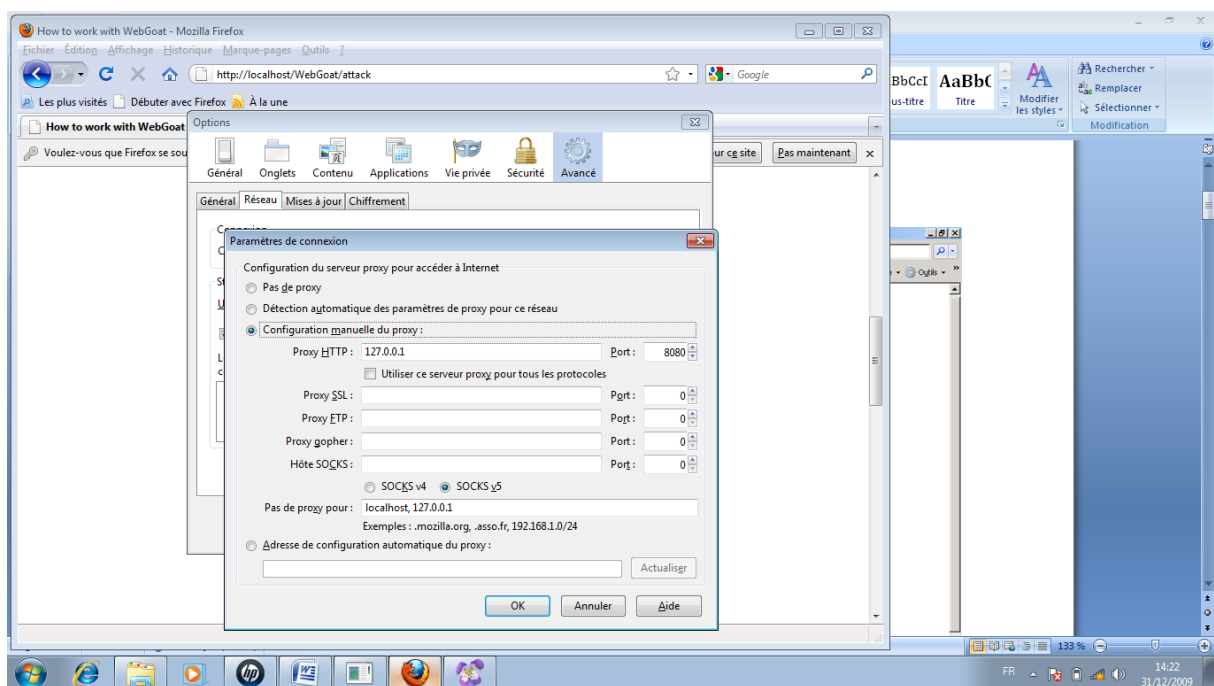


TP1 : Utilisation de WebGoat pour tester des attaques sur les applications Web

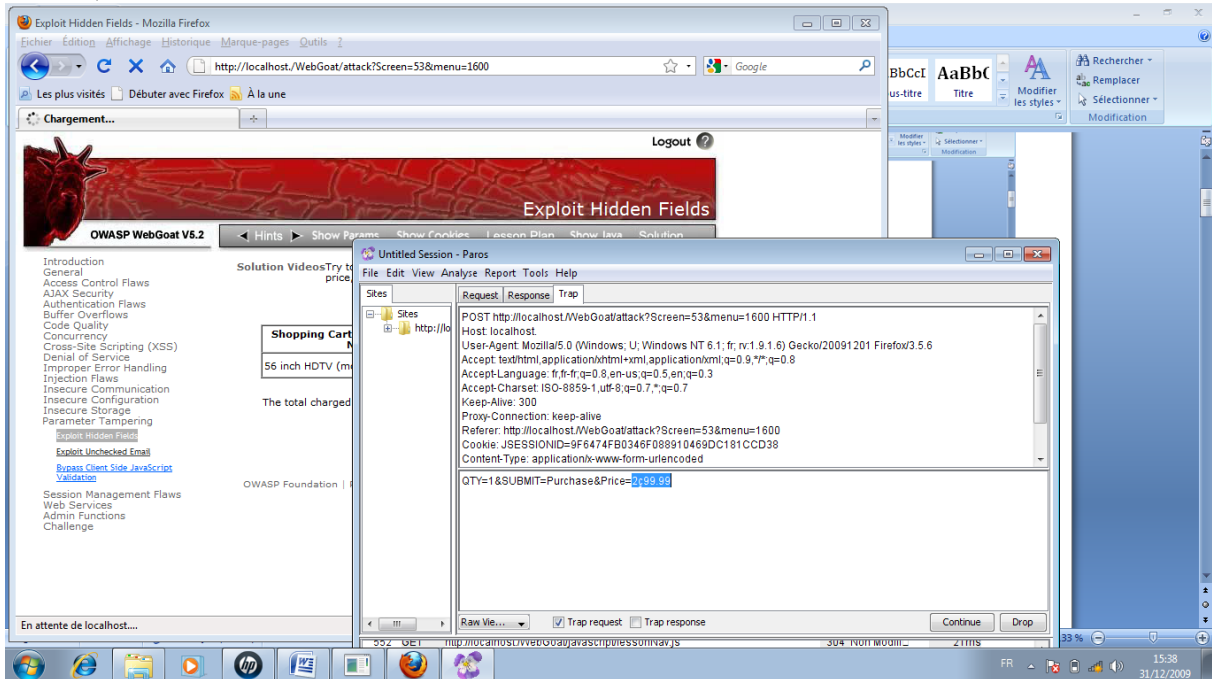
Etapes initiales pour faire les TPs WebGoat:

1. Déconnexion du réseau (pour éviter des risques de sécurité)
2. Lancer la machine virtuelle (WinXp) (pour avoir des droits d'admin)
3. Installation de WebGoat5.2 (fichier webgoat.bat)
4. Installation du proxy Paros (on peut à la place de Paros, utiliser WebScarab de l'OWASP)
5. Via un navigateur on se connecte au site web local installé par WebGoat en tapant l'URL suivante : <http://localhost/WebGoat/attack>
6. Pour accéder on tape **guest** dans login et Mot de pass
7. Dans le navigateur (options/./Param Réseaux (IE) ou Avancé/Réseau(Firefox)) on déclare notre proxy (IP : 127.0.0.1 et port : 8080)
8. Parfois pour indiquer qu'on passe par le proxy on ajoute un point après localhost (ça dépend des cas)

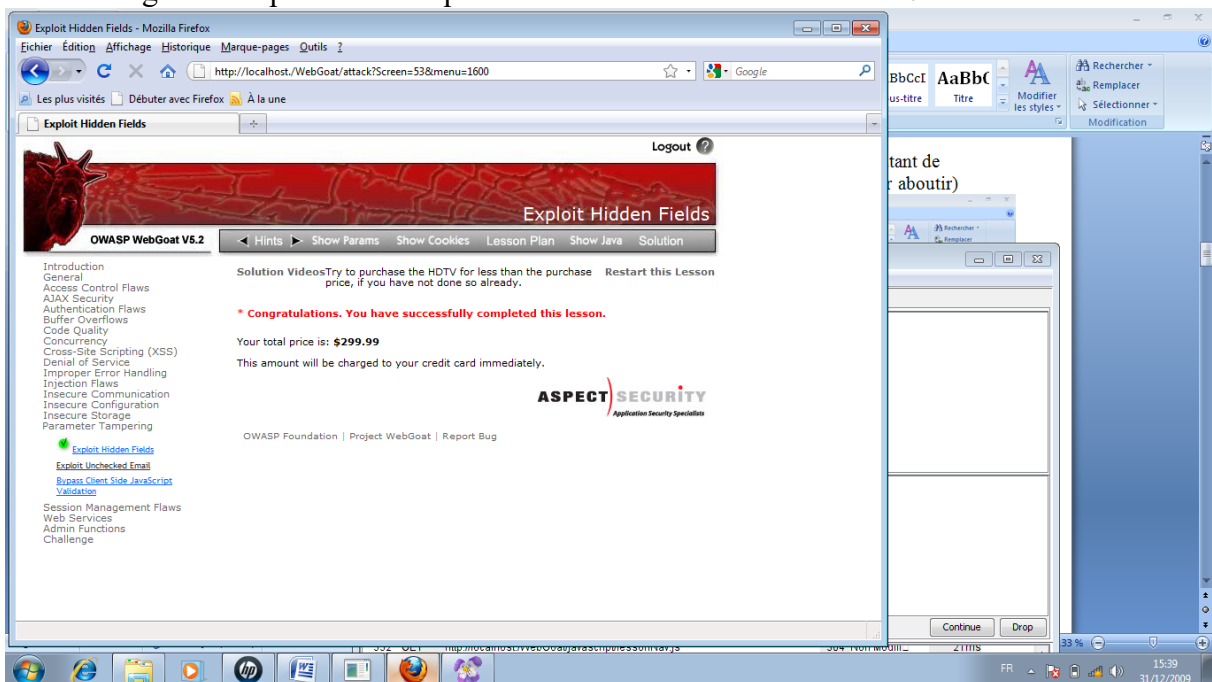


Ex 1 : L'exploitation des champs cachés d'un formulaire

1. Dans le site WebGoat on choisit la rubrique « [Parameter Tampering](#) »/ l'attaque « [Exploit Hidden Fields](#) »
2. Dans Paros (après lancement) on coche TrapRequest (en bas) dans l'onglet Trap
3. Dans le site on clique sur Purchase
4. Dans Paros, la requête Post correspondante est interceptée et **on modifie le montant** et on clique autant de fois que nécessaire sur 'Continue' (pour laisser les autres requêtes http du navigateur aboutir)

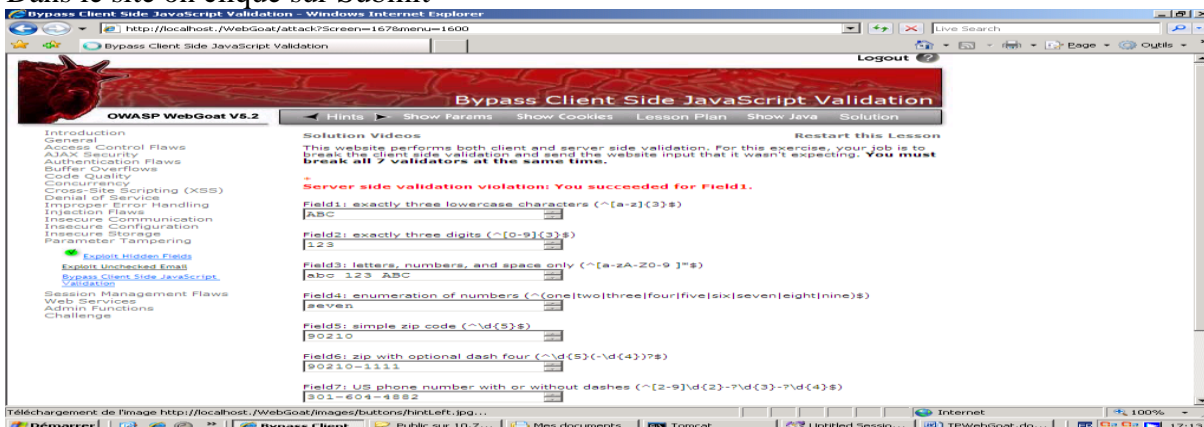


5. Sur le navigateur on peut vérifier qu'on a réussi à modifier le montant !

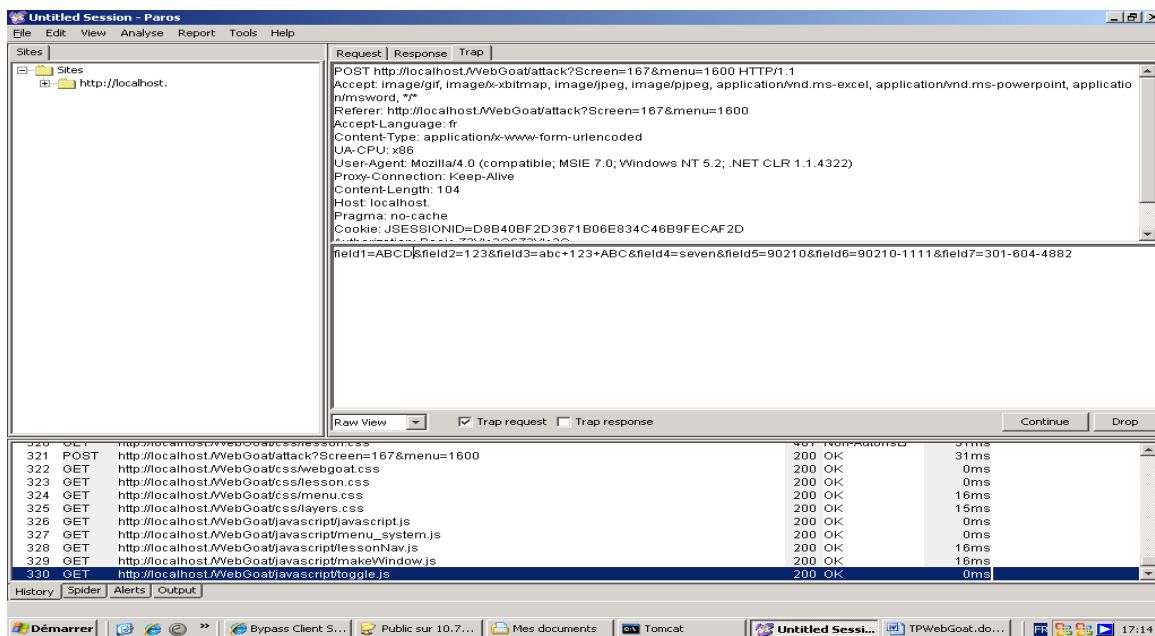


Ex 2 : Modification des règles de validation des champs d'un formulaire

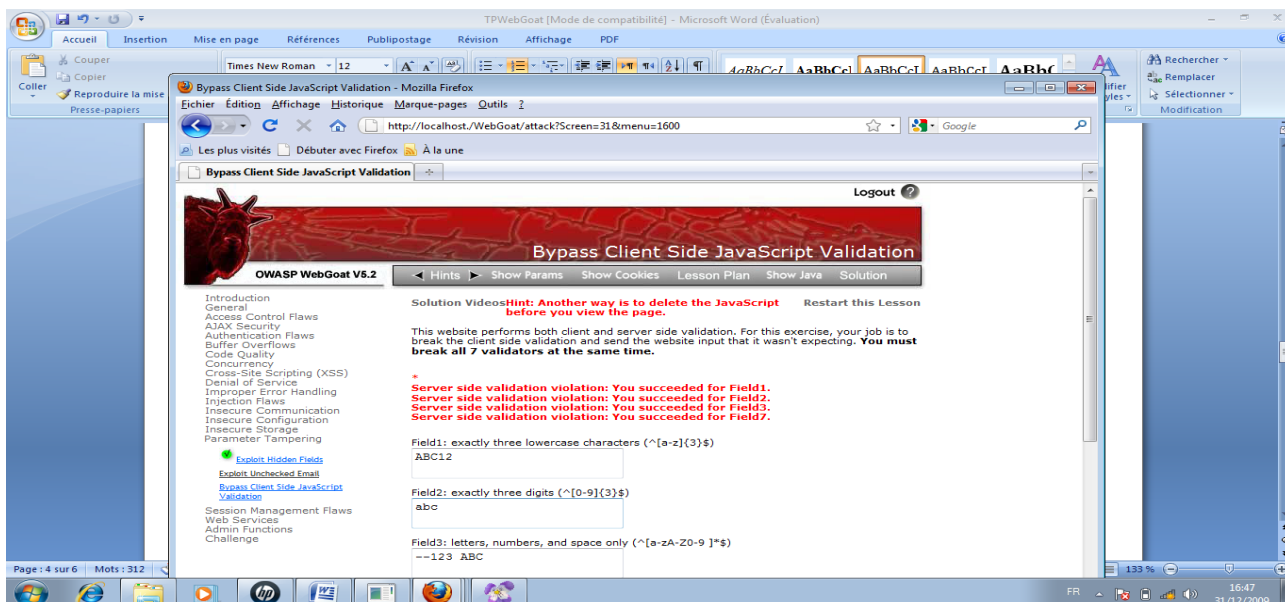
1. Dans le site WebGoat on choisit la rubrique « Parameter Tampering »/ l'attaque « Bypass Client Side JavaScript Validation »
2. Dans Paros (après lancement) on coche TrapRequest (en bas) dans l'onglet Trap
3. Dans le site on clique sur Submit



4. Dans Paros, la requête Post correspondante est interceptée et **on modifie les règles de validation** et on clique autant de fois que nécessaire sur 'Continue' (pour laisser les autres requêtes http du navigateur aboutir)

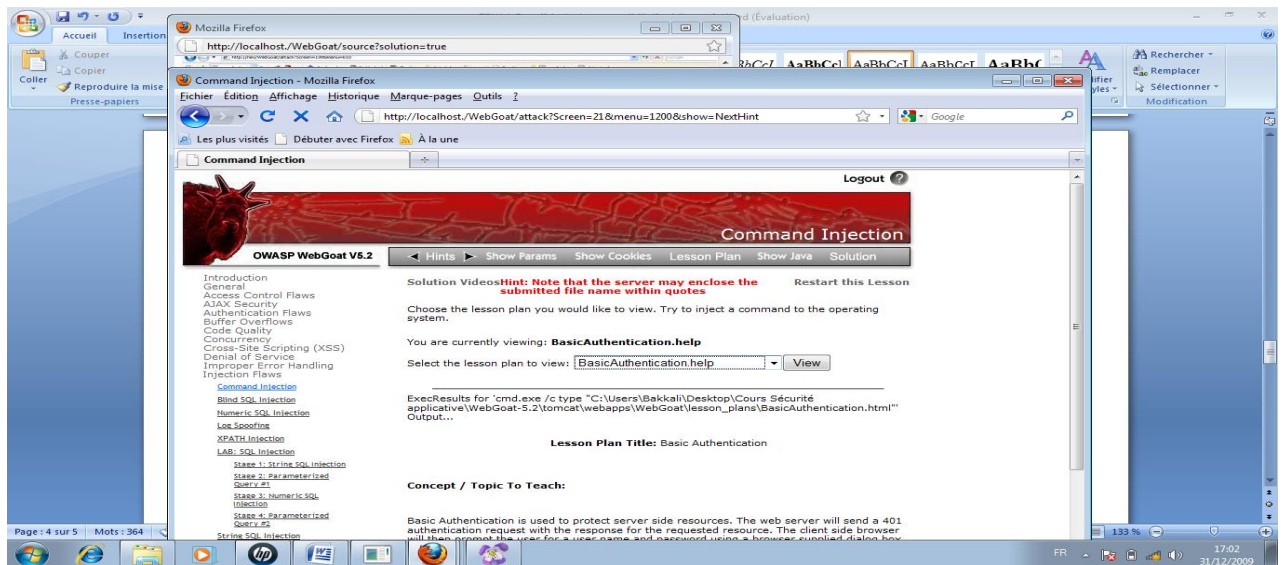


5. Dans le site on vérifie qu'on a réussi.

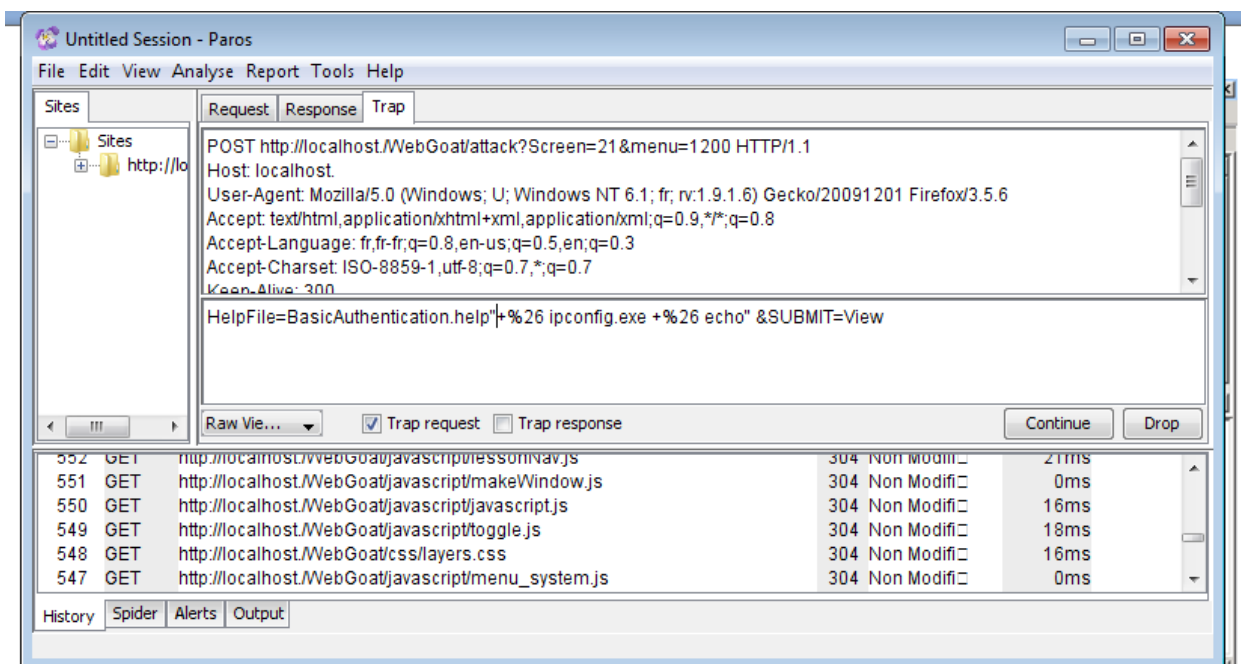


Ex 3 : Injection de commandes

1. Dans le site WebGoat on choisit la rubrique « Injection Flaws »/ l'attaque «Command Injection »
2. On choisit une lesson dans la liste (exp : BasicAuthentication)



3. Dans Paros (après lancement) on coche TrapRequest (en bas) dans l'onglet Trap
4. Dans le site on clique sur View
5. Dans Paros on modifie la variable passé à la commande en ajoutant une autre commande comme: "+%26 ipconfig.exe +%26 echo" comme suit (puis on clique sur Continue plusieurs fois comme d'hab) (Rq:%26 est le code HTML de &)



6. Résultat sur le site :

Command Injection - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ?

http://localhost/WebGoat/attack?Screen=21&menu=1200

Les plus visités Débuter avec Firefox À la une

Command Injection

Code Quality
Concurrency
Cross-Site Scripting (XSS)
Denial of Service
Improper Error Handling
Injection Flaws

[Command Injection](#)
[Blind SQL Injection](#)
[Numeric SQL Injection](#)
[Log Spoofing](#)
[XPath Injection](#)
[LDAP: SQL Injection](#)
[Stage 1: String SQL Injection](#)
[Stage 2: Parameterized Query #1](#)
[Stage 3: Numeric SQL Injection](#)
[Stage 4: Parameterized Query #2](#)
[String SQL Injection](#)
[Database Backdoors](#)
Insecure Communication
Insecure Configuration
Insecure Storage
Parameter Tampering
Session Management Flaws
Web Services
Admin Functions
Challenge

Select the lesson plan to view: View

ExecResults for 'cmd.exe /c type "C:\Documents and Settings\Administrateur\Bureau\WebGoat-5.2\tomcat\webapps\WebGoat\lesson_plans\BasicAuthentication.html" & ipconfig.exe & echo ""'
Output...

Lesson Plan Title: Basic Authentication

Concept / Topic To Teach:

Basic Authentication is used to protect server side resources. The web server will send a 401 authentication request with the response for the requested resource. The client side browser will then prompt the user for a user name and password using a browser supplied dialog box. The browser will base64 encode the user name and password and send those credentials back to the web server. The web server will then validate the credentials and return the requested resource if the credentials are correct. These credentials are automatically resent for each page protected with this mechanism without requiring the user to enter their credentials again.

General Goal(s):

For this lesson, your goal is to understand Basic Authentication and answer the questions below.

Configuration IP de Windows

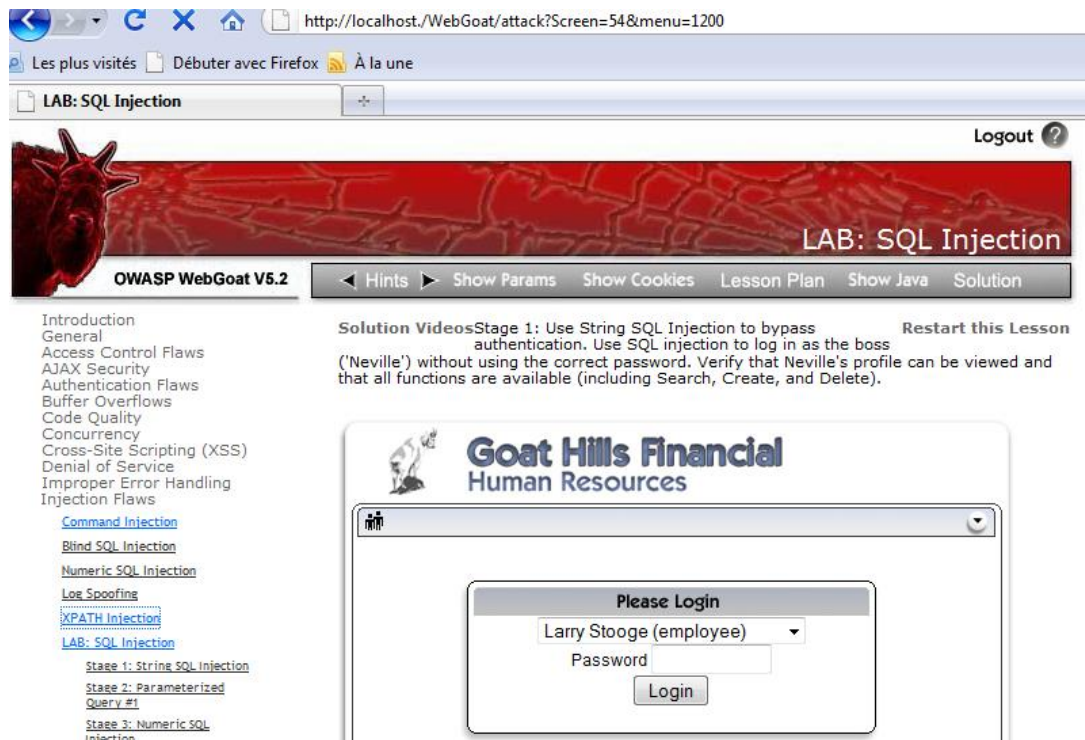
Carte Ethernet Connexion au réseau local :
Suffixe DNS propre à la connexion :
Adresse IP : 10.7.1.4
Masque de sous-réseau : 255.255.0.0
Passerelle par défaut : 10.0.0.1
Returncode: 1
Bad return code (expected 0)

OWASP Foundation | Project WebGoat | Report Bug

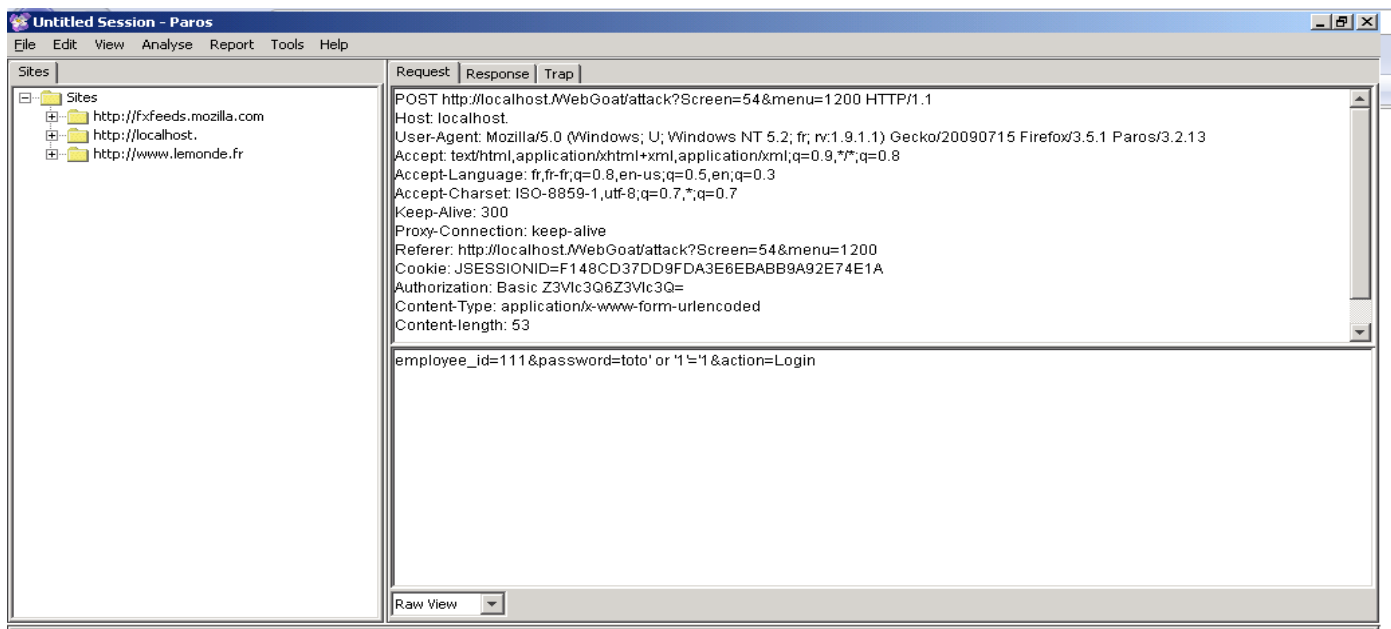
Démarrer Public sur 10.7... Mes documents Tomcat Command Inj... Untitled Sessio... TPWebGoat.do... 09:45

Ex 4 : Injection SQL

1. Dans le site WebGoat on choisit la rubrique « Injection Flaws »/ Lab SQL Injection
2. On saisi n'importe quel mot de pass



3. Dans Paros on fait la modif de la requête comme suit (on vise à passer dans la requête SQL un nouveau login/pasword permettant une intrusion) puis le fameux clic sur Continuer et le décochage de Trap request :



4. Résultat réussi sans connaissance du mot de passe :

http://localhost./WebGoat/attack?Screen=54&menu=1200

Les plus visités Débuter avec Firefox À la une

LAB: SQL Injection

Voulez-vous que Firefox se souvienne de ce mot de passe pour http://localhost. ?

Retenir Jamais pour ce site Pas maintenant

LAB: SQL Injection

OWASP WebGoat V5.2

< Hints > Show Params Show Cookies Lesson Plan Show Java Solution Restart this Lesson


Introduction
General
Access Control Flaws
AJAX Security
Authentication Flaws
Buffer Overflows
Code Quality
Concurrency
Cross-Site Scripting (XSS)
Denial of Service
Improper Error Handling
Injection Flaws

[Command Injection](#)
[Blind SQL Injection](#)
[Numeric SQL Injection](#)
[Log Spoofing](#)
[XPath Injection](#)
[LAB: SQL Injection](#)

Stage 1: String SQL Injection
Stage 2: Parameterized Query #1

Solution VideosStage 2: Block SQL Injection using a Parameterized Query.
Implement a fix to block SQL injection into the fields in question on stage 1. Verify that the attack is no longer effective.

*** You have completed String SQL Injection.**
*** Welcome to Parameterized Query #1**



Goat Hills Financial Human Resources

Welcome Back **Neville** - Staff Listing Page

Select from the list below

Larry Stooze (employee)
Moe Stooze (manager)