

TECHNISCHE
UNIVERSITÄT
WIEN

VIENNA
UNIVERSITY OF
TECHNOLOGY

Abgabe 2: WEB GOAT

Version 1.0

Dokumentenverlauf

| Datum | Änderungen | Autor |
|--------------|----------------------|---------------|
| 21.Nov. | Initialversion | Gerald Haider |
| 5. Dez. | Erweiterung | Gerald Haider |
| 6. Dez. | Finale Überarbeitung | Gerald Haider |

Inhaltsverzeichnis

| | | |
|-----------|---|-----------|
| 1 | GENERAL | 3 |
| 1.1 | HTTP BASICS | 3 |
| 1.2 | THREAD SAFETY | 3 |
| 2 | CODE QUALITY | 3 |
| 2.1 | HTML CLUES | 3 |
| 3 | UNVALIDATED PARAMETERS | 4 |
| 3.1 | HIDDEN FIELD TAMPERING | 4 |
| 3.2 | UNCHECKED EMAIL..... | 5 |
| 3.3 | JAVASCRIPT VALIDATION | 6 |
| 4 | BROKEN ACCESS CONTROL | 6 |
| 4.1 | PATH BASED ACCESS CONTROL..... | 6 |
| 4.2 | ROLE BASED ACCESS CONTROL..... | 6 |
| 5 | BROKEN AUTHENTICATION AND SESSION MANAGEMENT | 7 |
| 5.1 | WEAK AUTHENTICATION COOKIE | 7 |
| 6 | CROSS-SITE SCRIPTING (XSS) FLAWS | 8 |
| 6.1 | STORED XSS | 8 |
| 6.2 | REFLECTED XSS | 8 |
| 7 | INJECTION FLAWS | 9 |
| 7.1 | COMMAND INJECTION | 9 |
| 7.2 | NUMERIC SQL INJECTION | 9 |
| 7.3 | BLIND SQL INJECTION | 10 |
| 7.4 | STRING SQL INJECTION..... | 10 |
| 7.5 | ADVANCED SQL INJECTION (SCHWIERIG) | 11 |
| 8 | IMPROPER ERROR HANDLING | 11 |
| 8.1 | FAIL OPEN AUTHENTICATION | 11 |
| 9 | DENIAL OF SERVICE | 11 |
| 9.1 | DOS MULTIPLE LOGIN..... | 11 |
| 10 | WEB SERVICES (ZUSATZPUNKTE) | 12 |
| 10.1 | SOAP REQUEST..... | 12 |
| 10.2 | WSDL SCANNING..... | 12 |
| 10.3 | WEB SERVICE SQL INJECTION | 12 |

1 General

1.1 Http Basics

1.1.1 - Machen Sie sich mit der WebGoat Umgebung vertraut.

Erledigt ☺

1.2 Thread Safety

1.2.1 Was ist unter dem Begriff „Thread Safety“ zu verstehen?

“Thread-safety is a computer programming concept applicable in the context of multi-threaded programs. A piece of code is thread-safe if it functions correctly during simultaneous execution by multiple threads. In particular, it must satisfy the need for multiple threads to access the same shared data, and the need for a shared piece of data to be accessed by only one thread at any given time.” (Wikipedia, <http://en.wikipedia.org/wiki/Thread-safety> , Zugriff 6.12.2005)

1.2.2 Wie können Sie den Fehler in diesem Fall ausnutzen? Wie macht er sich bemerkbar? Beschreiben Sie Ihr Vorgehen.

Das Ausnutzen des Fehlers ist möglich durch schnelles Aufrufen der Seite mit jeweils anderen Usernamen, sprich 2 Browserfenster, in einem ein Request mit User „Dave“ im anderen mit User „Jeff“. Ergebnis: in beiden Browserfenstern erhält man den Output von einem User.

1.2.3 Wo liegt der Fehler im Source Code („Show Java“ Funktion benutzen)?

Der Fehler liegt in der nicht threadsicheren Variable „currentUser“ deren Wert vom zweiten Thread überschrieben wird, während der erste Thread noch

```
66         Thread.sleep( 1500 );
```

ausführt. Diese sleep Anweisung vereinfacht das Ausnutzen der Lücke erheblich.

2 Code Quality

2.1 HTML Clues

2.1.1 Was ist unter „HTML Clues“ zu verstehen?

Das sind Codeteile im HTML die vielfach noch von den Programmieren, ursprünglich meist für Debugging Zwecke hinterlassen wurden, und dann vielfach noch in der „finalen“ Seite vorhanden sind, aber nicht auf Anhieb sichtbar sind, wie z.B.: Hinweise in HTML Kommentaren.

2.1.2 Wie können Sie in diesem Beispiel ausgenutzt werden?

die folgenden Codezeilen im HTML Sourcecode verraten den korrekten Login.

```
103 <!--  
104 FIXME admin:adminpw  
105 --><!--
```

3 Unvalidated Parameters

3.1 Hidden Field Tampering

3.1.1 Was versteht man unter „Hidden Field Tampering“?

Das Verändern von Informationen die in versteckten HTML Eingabefeldern übergeben wird.

3.1.2 Beschreiben Sie, wie Sie vorgehen, um den HDTV im Beispiel um \$10 zu zerstören.

Aufrufen der Seite über einen Browser mit zwischengeschaltetem WebScarab, welcher auf „Intercept requests“ eingestellt ist.

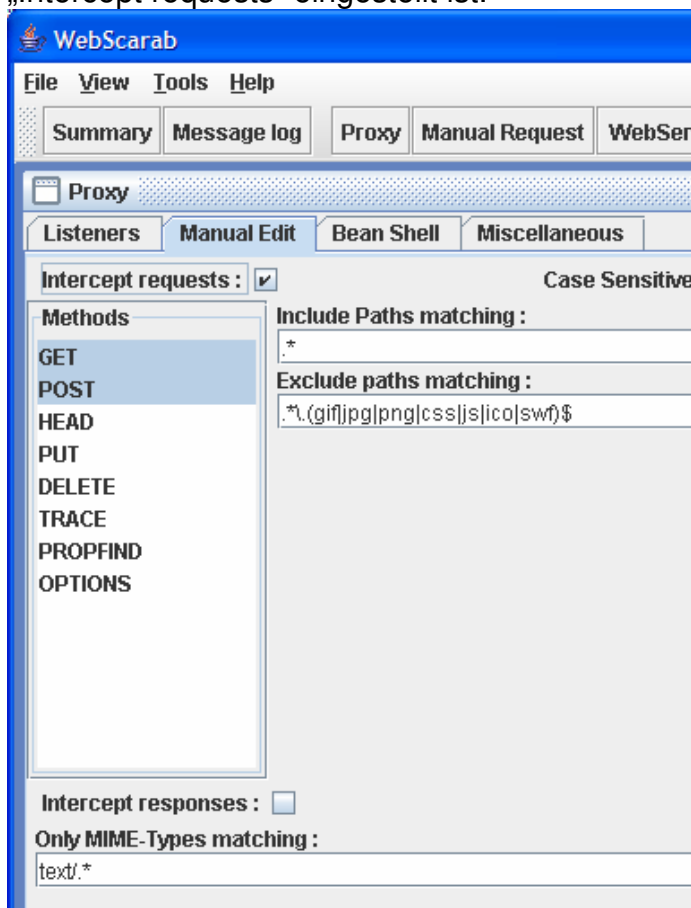


Abbildung 1 - WebScarab Einstellungen

Damit ist es möglich den Wert des Hidden Field in dem der Preis des HDTV gespeichert wird zu verändern und somit das Gerät billiger zu entstehen.

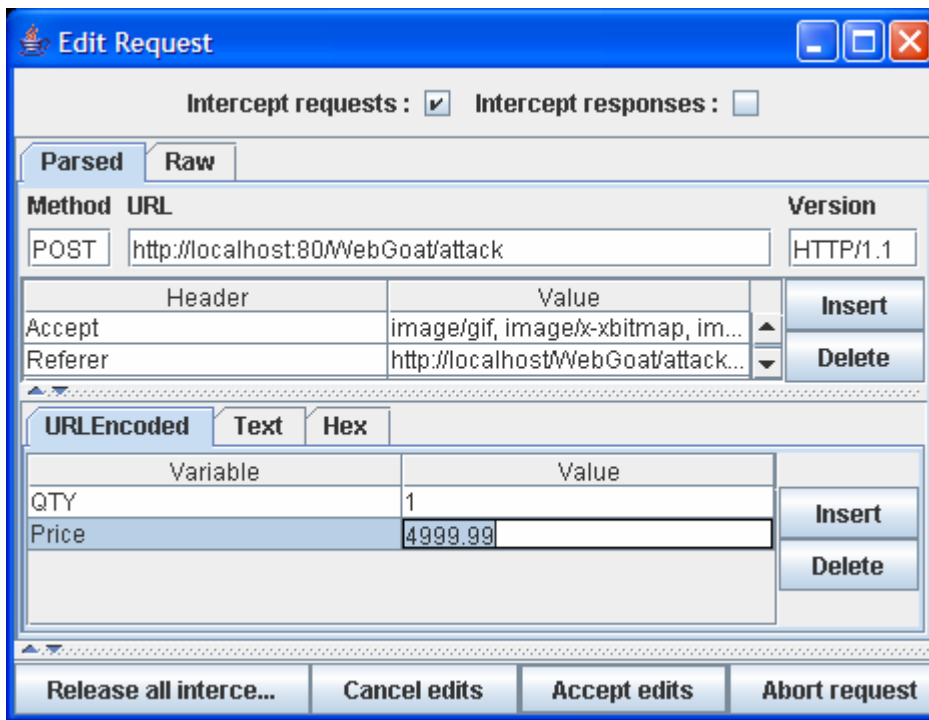


Abbildung 2 - WebScarab Intercept request

3.2 Unchecked Email

3.2.1 Was ist mit „Unchecked Email“ gemeint?

Ein Email Formular bei welchem die Usereingaben nicht auf potentiell gefährliche Inhalte hin überprüft wird, sprich es werden keine Scripts, bzw. sonstige spezielle (potentiell gefährliche) Zeichen ausgefiltert.

3.2.2 Beschreiben Sie Ihr Vorgehen beim Lösen der gestellten Aufgaben.

Zuerst wurde folgender „böser“ Skript in das Comments Feld eingetragen:

```
<script>alert("Bad Stuff");</script>
```

dann wurde wiederum mittels WebScarab „Intercept requests“ der „To“ Parameter modifiziert zwecks übermitteln an eine andere Email Adresse.

3.2.3 Welche beiden verschiedenen potentiellen Sicherheitslöcher können Sie ausmachen und wie ließen sie sich verhindern?

Die Empfängeradresse wird in einem für den Angreifer leicht manipulierbaren Hidden Field gespeichert.

Der Inhalt des Comments Fields wird nicht auf potentiell gefährlichen Code gefiltert.

3.2.4 Beschreiben Sie an einem Beispiel, welchen maximalen Schaden ein Angreifer anrichten könnte und wie er dabei vorgehe.

Der Angreifer führt über das Email Formular Code mit den Rechten des Webservers aus, und kann so event. Files löschen/überschreiben (z.B.: bei PHP basierendem Mailformular

und schlechter Serverkonfiguration. Das Mailformular könnte außerdem zum versenden von Spam/Phishing Mails verwendet werden.

3.2.5 Welche Konsequenzen ergeben sich für den Betreiber der Website, auf der dieses Form läuft?

Das ganze System kann kompromittiert werden, was vom simplen Datenverlust bis zu rechtlichen Problemen führen kann.

3.3 JavaScript Validation

3.3.1 Was versteht man unter „JavaScript Validation“ und worin besteht das Sicherheitsrisiko?

Darunter versteht man das Clientseitige Überprüfen von Usereingaben, da JavaScript aber vom User nach belieben abgeschaltet werden kann, können diese Scripte laufen, müssen aber nicht, sprich es muss immer zur Sicherheit auch eine Serverseitige Überprüfung der Eingabeparameter erfolgen.

3.3.2 Beschreiben Sie Ihren Lösungsweg, um die Javascript Validation zu umgehen.

Wiederum erfolgte die Lösung mittels WebScarab und der „Intercept requests“ Funktion, womit die Parameter manuell in normalerweise vom Javascript beanstandete Werte geändert wurden.

4 Broken Access Control

4.1 Path Based Access Control

4.1.1 Was versteht man unter einer „Directory Traversal Attack“?

Das Ausnutzen von nicht überprüften Pfadangaben in einer Applikation, wie z.B.: die unvorhergesehene Verwendung von ../

4.1.2 Wie könnten Sie sie als Entwickler verhindern?

Filtern jeglicher Benutzereingaben, wobei der sicherere Ansatz ist nur gewisse Zeichen zu erlauben und alles andere zu verbieten.

4.2 Role Based Access Control

4.2.1 Was versteht man unter „Role Based Access Control“?

Die Vergabe von Berechtigungen basierend auf vorgegebenen Rollen, anstatt die Berechtigungen direkt an den jeweiligen User zu vergeben, wird dem User eine gewisse Rolle und damit Berechtigungen zugewiesen.

4.2.2 Welche User haben außer dem Administrator noch Zugriff auf die Ressource „Account Manager“ und welche eine Zeile im Java Source Code ist dafür verantwortlich?

Der User „Larry – Manager“ hat ebenfalls Zugriff. Verantwortlich dafür ist folgendes Code-segment:

```
169         if ( rl.contains( roles[1] ) )
170         {
171             list.add( resources[1] );
172             list.add( resources[5] );
173         }
```

Die Zeile 172 gewährt die zusätzlichen Admin Rechte.

5 Broken Authentication and Session Management

5.1 Weak Authentication Cookie

5.1.1 Beschreiben Sie, was unter „Weak Authentication Cookie“ zu verstehen ist.

Jegliche Art von Cookie die sich umkehren lassen.

5.1.2 Beschreiben Sie Ihr Vorgehen beim Ausnutzen der Schwachstelle.

Mittels Einloggen in die 2 vorhandenen Accounts wurden die folgenden AuthCookie Strings gefunden:

```
65432ubphcfx
65432udfqtb
```

Und daraus ist nicht schwer zu erraten das es sich um einen simplen Cäsar Cipher mit Verschiebung 1 handelt, um das ganze zu erschweren wird der String vorher noch umgekehrt.

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

```
65432ubphcfx
54321taogbew <- webgoat12345 verkehr herum
```

```
65432udfqtb
54321tcepsa <- aspect vekehrt herum
```

```
alice12345 <- ausgangsstring
54321ecila <- umkehren
65432fdjmb <- "verschlüsseln"
```

Nach dieser kleinen Verschlüsselungsaufgabe wird das Cookie mit Webscarab modifiziert und refresh gedrückt → Fertig!

5.1.3 Wie können Sie als Entwickler diese Sicherheitslücke verhindern?

Verwendung von sicheren Hashingalgorithmen (SHA-265) für solche Cookies und einbauen eines wirklichen Zufallswertes.

6 Cross-Site Scripting (XSS) Flaws

6.1 Stored XSS

Hinweis: offenbar existiert in WebGoat bei dieser Aufgabe ein kleiner Fehler, so dass die Aufgabe auch nach korrekter Lösung nicht grün angezeigt wird.

6.1.1 Was versteht man unter „Cross Site Scripting (XSS)“?

Cross-Site Scripting (XSS) bezeichnet das Ausnutzen einer Computersicherheitslücke, indem Informationen aus einem Kontext, in dem sie nicht vertrauenswürdig sind, in einen anderen Kontext eingefügt werden, in dem sie als vertrauenswürdig eingestuft sind. Aus diesem vertrauenswürdigen Kontext kann dann ein Angriff gestartet werden. (Wikipedia, <http://de.wikipedia.org/wiki/XSS> , Zugriff 6.12.2005)

6.1.2 Was unter Database XSS?

Selbes Prinzip wie bei 6.1.1, nur wird der Schadcode am Server dauerhaft z.B.: in einer Datenbank gespeichert.

6.1.3 Worauf sollten Sie als Entwickler achten, um XSS zu verhindern?

Filtern bzw. Escapen jeglicher möglicher Benutzereingabedaten. Benutzereingaben niemals direkt im Programmcode weiterverwenden.

6.1.4 Eine Website stellt von Usern eingegebene Kommentare einfach in `<PRE>...</PRE>` Tags und setzt sonst keine weiteren Maßnahmen – reicht das als Schutzmaßnahme aus (Begründung)?

Nein, weil der Benutzer kann ja auch mit einem eigenen `</pre>` in seiner Eingabe das schließen des `<PRE>` tags verursachen, und danach seinen, beliebigen Code einfügen.

6.2 Reflected XSS

6.2.1 Was versteht man unter „Reflected XSS“?

Der Schadcode wird in den Request an den Server (z.B.: den URL der aufgerufen wird) eingebettet und wird beim Anzeigen des Responds ausgeführt.

6.2.2 Ändern Sie den „Show Lessons Plan“-Button oben rechts und lassen Sie ihn stattdessen „XSS“ anzeigen. Dokumentieren Sie Ihr Vorgehen.

Dazu wird der folgende Code in das „digit access code“ Feld eingegeben:

```
<script>document.getElementsByTagName("INPUT")[5].value = "XSS";</script>
```

Fertig! ☺

7 Injection Flaws

7.1 Command Injection

Hinweis: Sie können Tools-Transcoder von WebScarab [1] verwenden, falls Sie URLEncoding benötigen sollten.

7.1.1 Was ist unter Parameter Injection / Command Injection zu verstehen?

Das Einfügen von zusätzlichen Parameter bzw. Commandos in Aufrufe die von einem Programm gemacht werden.

7.1.2 Wie können Sie zusätzlich zur „type“-Anweisung einen anderen Befehl starten?

Unter Windows durch eingabe des „&“ Zeichens.
z.B.:

```
ExecResults for 'cmd.exe /c type "D:\PAS\ WebGoat
\tomcat\webapps\WebGoat\lesson_plans\"BasicAuthentication.html &
netstat -a & ipconfig'
```

Unter Unix wäre das ganze mit „;“ zu erreichen.

7.1.3 Mit welcher Eingabe können Sie herausfinden, welche Version von Windows auf dem Rechner installiert ist?

Kurz und bündig: `ver`

Unter manchen Windowsversionen erhält man genauere Infos mit dem Kommando:

```
systeminfo
```

7.2 Numeric SQL Injection

7.2.1 Was ist unter „SQL Injection“ zu verstehen?

SQL injection is a security vulnerability that occurs in the database layer of an application. Its source is the incorrect escaping of dynamically-generated string literals embedded in SQL statements. It is in fact an instance of a more general class of vulnerabilities that can occur whenever one programming or scripting language is embedded inside another. (Wikipedia, http://en.wikipedia.org/wiki/SQL_Injection , Zugriff 6.12.2005)

7.2.2 Wie können Sie im konkreten Beispiel die gesamte Tabelle auslesen?

Durch eingeben des folgenden Texts in das Eingabefeld:

101 or 1=1

7.2.3 Bisher operieren Sie nur auf einer bestimmten Tabelle. Wie könnte man mittels SQL Injection auch Daten von anderen Tabellen auslesen?

Entweder durch die Verwendung von JOIN Operatoren oder das absetzen von mehreren SQL Kommandos nacheinander z.B.: SELECT * FROM db1; SELECT * FROM db2;

7.3 Blind SQL Injection

7.3.1 Was ist unter „Blind SQL Injection“ zu verstehen?

When an attacker executes SQL Injection attacks sometimes the server responds with error messages from the database server complaining that the SQL Query's syntax is incorrect. Blind SQL injection is identical to normal SQL Injection except that when an attacker attempts to exploit an application rather than getting a useful error message they get a generic page specified by the developer instead. This makes exploiting a potential SQL Injection attack more difficult but not impossible. An attacker can still steal data by asking a series of True and False questions through sql statements. (Cgisecurity.com, <http://www.cgisecurity.com/questions/blindsqli.shtml> , Zugriff 6.12.2005)

7.3.2 Erklären Sie Ihr konkretes Vorgehen, um den Wert des Feldes „first_name“ für den User mit „userid“ = 15613 herauszufinden.

Hints lesen spart viel Arbeit! Danach schrittweise Annäherung durch solche Abfragen (feststellen ob der ASCII Code des erste Zeichens kleiner 77 ist:

```
101 AND (asc( mid((SELECT first_name FROM user_data WHERE
userid=15613) , 1 , 1) ) < 77 );
101 AND (asc( mid((SELECT first_name FROM user_data WHERE
userid=15613) , 2 , 1) ) < 111 );
101 AND (asc( mid((SELECT first_name FROM user_data WHERE
userid=15613) , 3 , 1) ) = 101 );
```

Ergebnis:

```
74 -> J
111 -> o
101 -> e
```

7.4 String SQL Injection

7.4.1 Erklären Sie Ihr Vorgehen zur Lösung der Aufgabe.

Folgende Eingabe führt zur ausgabe aller Daten:

```
smith' or 1=1 or last_name = 'bla
```

7.5 Advanced SQL Injection (schwierig)

7.5.1 Da Sie Einblick in den Source einer alten Version der Software haben, wissen

Sie, dass sie auch einen Table „product_system_data“ beinhaltet, welcher Daten zu Produkten bereitstellt. Diese Tabelle „product_system_data“ beinhaltet 3 Spalten: „productid“, „product_name“ und „price“. Versuchen Sie, alle in dieser Tabelle enthaltenen Informationen über das Textfeld von Numeric oder String SQL Injection abzurufen (beides möglich) und beschreiben Sie Ihr Vorgehen detailliert.

8 Improper Error Handling

8.1 Fail Open Authentication

8.1.1 Beschreiben Sie, was unter einer „Fail Open“-Problematik zu verstehen ist.

Unter der „Fail Open“-Problematik versteht man das zurückfallen in eine unsicher default Einstellung im falle eines Fehlers.

8.1.2 Wie können Sie diesen Zustand in diesem Beispiel erreichen/ausnutzen? Beschreiben Sie Ihr Vorgehen.

Ins Feld „User Name“ wird ganz normal „webgoat“ eingegeben, danach auf Login geklickt, der Request wird mit WebScarab abgefangen und der URLEncoded Parameter „Password“ wird komplett gelöscht bevor der Request weitergeschickt wird. Fertig!

8.1.3 Wo im Source Code liegt der "Programmierfehler" und wie können Sie als Entwickler der Problematik vorbeugen?

Der Fehler liegt in dieser Zeile:

```
50      if ( !"webgoat".equals( username ) || !password.equals(
"webgoat" ) )
```

Bei fehlen des "PASSWORD" Parameters kommt es hier zu einer Exception und im anschlienden Catch Block wird jeder User mit gültigem Usernamen hineingelassen.

9 Denial of Service

9.1 DOS Multiple Login

9.1.1 Was ist eine DOS Attacke?

DOS = Denial of Service, das Überfluten eines Zieles mit einer so großen Anzahl von Anfrage so das dieses nicht mehr korrekt verarbeitet werden können.

9.1.2 Beschreiben Sie Ihren Lösungsweg, um die gesuchten Usernamen/Passwörter anzuzeigen.

Sowohl in das „User Name“ Feld als auch in das „Password“ Feld wurde folgendes eingetragen:

```
' or '1' ='1
```

Dadurch wurde die komplette Liste der Usernamen & Passwörter ermittelt.

10 Web Services (Zusatzpunkte)

Hinweis: Die Fragen dieses Abschnitts berühren mit Web Services ein Thema, das etwas Hintergrundwissen verlangt und sind daher optional. Durch Beantwortung können insgesamt 10 zusätzliche Prozentpunkte gewonnen werden.

10.1 Soap Request

10.1.1 Was ist SOAP (kurz)?

10.1.2 Was ist WSDL (kurz)?

10.1.3 Wie können Sie die vom Web Service angebotenen Methoden in Erfahrung bringen?

10.1.4 Welchen Typ hat die "getFirstNameRequest"-Methode?

10.1.5 Erklären Sie Ihren Lösungsweg zum Aufrufen einer der Operationen.

10.2 WSDL Scanning

10.2.1 Wie können Sie sich zusätzlich die Kreditkarteninformationen eines Users ausgeben lassen?

10.3 Web Service SQL Injection

10.3.1 Erklären Sie Ihren Lösungsweg zur Erfüllung der gestellten Aufgabe.