

Übung, Teil 2: **WebGoat**

Typ: Einzelarbeit
Deadline Bericht: 06.12.05 23:55
Abgabegespräch: 31.01.06

Für diesen Teil der Übung kommt das in Java geschriebene Security Lesson Framework [WebGoat](#) zum Einsatz. *WebGoat* ist eine J2EE Applikation, die der Lehre von Sicherheitsrelevanten Aspekten in Webapplikationen dient.

Unterschiedliche Gebiete sind dabei in übersichtliche Lektionen aufgegliedert, die gelöst werden sollen. Dabei sind Parameter der dynamischen Seite, Cookies und Source Code jeweils per Knopfdruck einsehbar, was eine große Erleichterung zum händischen Vorgehen darstellt und mit ein Entscheidungsgrund für *WebGoat* ist. Ein Hilfe-System, das aufgabenspezifische Tips anbietet, ist ebenfalls integriert.

WebGoat läuft auf einem lokalen Webserver auf Ihrem Computer, alles Nötige ist im Paket enthalten.

Aufgabenstellung:

- Downloaden und Installieren (Entpacken) der aktuellen *WebGoat* Version von <http://www.owasp.org/software/webgoat.html>. Am besten die Version mit Java nehmen, in ein Rootverzeichnis entpacken (zu tiefe Verzeichnisse/Verzeichnisse mit Sonderzeichen könnten Probleme bereiten). README Datei lesen. Nicht vergessen, eventuelle andere am selben Port laufende lokale Webserver vorher zu beenden.
- Absolvieren der einzelnen Lektionen unter Berücksichtigung des Fragenkataloges. Im Fragenkatalog nicht erwähnte Lessons müssen NICHT bearbeitet werden (z.B. „Buffer Overflow“, „Forced Browsing“ aber auch die Challenge!).
- Erstellen eines detaillierten Berichts mit Antworten/Lösungen zu im Fragenkatalog aufgeführten Fragestellungen inklusive Lösungsweg, Links zu externen Seiten, die im Zuge der Lösung in Anspruch genommen wurden, Screenshots von relevanten Inhalten.

Deliverables:

06.12.05 23:55

- Lösungsbericht: Beschreibung der Lösung und des Vorgehens bezüglich der Beantwortung der Fragen, Erklärung, was warum wie gemacht wurde und welche Sicherheitsrisiken sich daraus ergeben. Links zu eventuell verwendeten externen Seiten und Programmen angeben, Screenshots von wichtigen Details.
- Bitte verwenden Sie die gleiche Nummerierung und Reihenfolge wie im Fragenkatalog, um eine Zuordnung zu erleichtern.
- Format: .pdf

Abgabegespräch:
31.01.06

- Beim Abgabegespräch müssen Sie Ihre Lösungswege erklären können sowie über die entsprechenden Grundlagen Bescheid wissen. Was bedeuten die aufgezeigten Sicherheitsprobleme für Sie als Entwickler?
 - o Beispielfragen: „Wie haben sie Aufgabe xyz gelöst?“ – „Was versteht man unter SQL Injection? Wie geht man vor, falls man eine Seite auf SQL Injection Probleme untersuchen möchte?“ – „Welche Probleme ergeben sich aus XSS? Beispiele? Was sollten Entwickler (bzw. User) daher beachten?“
 - o NICHT: „Wie lauten sämtliche Command Line Switches von cmd.exe?“ – „Wie lautet die MD5 von ‚Das wird sicher nicht gefragt‘?“
- Falls Sie eine besonders elegante / ungewöhnliche Lösung gefunden haben, weisen Sie bitte darauf hin.
- Bringen Sie bitte einen Ausdruck Ihres Berichts zum Abgabegespräch mit!

Punkteschema:

Insgesamt gibt es **15** Punkte zu erreichen. Da die verschiedenen Aufgaben sehr unterschiedliche Schwierigkeitsgrade aufweisen, wird zur besseren Gewichtung mit 100 Prozentpunkten gerechnet, wobei 10 zusätzliche Prozentpunkte durch Beantwortung der letzten Fragengruppe über Web Services erworben werden können.

Die Endpunktezahlgibt sich dann z.B. folgender Maßen:

- 75 Prozentpunkte erreicht (siehe Fragenkatalog für Verteilung)
- 75% von **15** Punkten entsprechen **11,25**
- Auf-/Abrunden auf halbe Punkte → Endpunktezahlg = **11** Punkte

Eine detaillierte Aufschlüsselung findet sich auf der folgenden Seite.

Detaillierte Aufschlüsselung der zu erreichenden Prozentpunkte je Aufgabe:

Kapitel	Unterabschnitt	Prozentpunkte
1 General		
	1.1 Http Basics	0
	1.2 Thread Safety	3
2 Code Quality		
	2.1 HTML Clues	2
3 Unvalidated Parameters		
	3.1 Hidden Fields	5
	3.2 Unchecked Email	5
	3.3 Javascript Validation	8
Broken Access		
4 Control		
	4.1 Path Based Access Control	5
	4.2 Role Based Access Control	7
5 Broken Authentication and Session Management		
	Weak Authentication	
	5.1 Cookie	8
6 Cross-Site Scripting (XSS)		
	6.1 Stored XSS	8
	6.2 Reflected XSS	8
7 Injection Flaws		
	7.1 Command Injection	6
	7.2 Numeric SQL Injection	6
	7.3 Blind SQL Injection	8
	7.4 String SQL Injection	2
	7.5 Advanced SQL Injection	10
8 Improper Error Handling		
	8.1 Fail Open Authentication	5
9 Denial of Service		
	9.1 DOS Multiple Login	4
10 Web Services		
	10.1 Soap Request	4
	10.2 WSDL Scanning	3
	10.3 Web Service SQL Injection	3
		110

FRAGENKATALOG

Vorwort

Die verschiedenen Lessons von WebGoat verfügen über sehr unterschiedliche Schwierigkeitsgrade, auch dieser Fragenkatalog setzt Schwerpunkte. Für manche der Aufgaben wird das Programm **WebScarab** [1] empfohlen (Kurzanleitung bei [1]). Sie müssen nur jene Teile absolvieren, die in diesem Fragenkatalog erwähnt sind.

1 General

1.1 *Http Basics*

- Machen Sie sich mit der WebGoat Umgebung vertraut. Probieren Sie die Funktionen durch, lassen Sie sich die Parameter anzeigen, die an die Seite übergeben werden. Schauen Sie sich HTML und Java Source Code an. „Show Lesson Plan“ zeigt an, was in der jeweiligen Lesson nähergebracht werden soll. Benutzen Sie die „Hint“-Funktion mehrmals, um Tips angezeigt zu bekommen.

1.2 *Thread Safety*

- Was ist unter dem Begriff „Thread Safety“ zu verstehen?
- Wie können Sie den Fehler in diesem Fall ausnutzen? Wie macht er sich bemerkbar? Beschreiben Sie Ihr Vorgehen.
- Wo liegt der Fehler im Source Code („Show Java“ Funktion benutzen)?

2 Code Quality

2.1 *HTML Clues*

- Was ist unter „HTML Clues“ zu verstehen?
- Wie können Sie in diesem Beispiel ausgenutzt werden?

3 Unvalidated Parameters

3.1 *Hidden Field Tampering*

- Was versteht man unter „Hidden Field Tampering“?
- Beschreiben Sie, wie Sie vorgehen, um den HDTV im Beispiel um \$10 zu erstehen.

3.2 *Unchecked Email*

- Was ist mit „Unchecked Email“ gemeint?
- Beschreiben Sie Ihr Vorgehen beim Lösen der gestellten Aufgaben.
- Welche beiden verschiedenen potentiellen Sicherheitslöcher können Sie ausmachen und wie ließen sie sich verhindern?

- Beschreiben Sie an einem Beispiel, welchen maximalen Schaden ein Angreifer anrichten könnte und wie er dabei vorgeht.
- Welche Konsequenzen ergeben sich für den Betreiber der Website, auf der dieses Form läuft?

3.3 JavaScript Validation

- Was versteht man unter „JavaScript Validation“ und worin besteht das Sicherheitsrisiko?
- Beschreiben Sie Ihren Lösungsweg, um die Javascript Validation zu umgehen.

4 Broken Access Control

4.1 Path Based Access Control

- Was versteht man unter einer „Directory Traversal Attack“?
- Wie könnten Sie sie als Entwickler verhindern?

4.2 Role Based Access Control

- Was versteht man unter „Role Based Access Control“?
- Welche User haben außer dem Administrator noch Zugriff auf die Ressource „Account Manager“ und welche eine Zeile im Java Source Code ist dafür verantwortlich?

5 Broken Authentication and Session Management

5.1 Weak Authentication Cookie

- Beschreiben Sie, was unter „Weak Authentication Cookie“ zu verstehen ist.
- Beschreiben Sie Ihr Vorgehen beim Ausnutzen der Schwachstelle.
- Wie können Sie als Entwickler diese Sicherheitslücke verhindern?

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.

- Was versteht man unter „Cross Site Scripting (XSS)“?
- Was unter Database XSS?
- Worauf sollten Sie als Entwickler achten, um XSS zu verhindern?
- Eine Website stellt von Usern eingegebene Kommentare einfach in `<PRE>...</PRE>` Tags und setzt sonst keine weiteren Maßnahmen – reicht das als Schutzmaßnahme aus (Begründung)?

6.2 Reflected XSS

- Was versteht man unter „Reflected XSS“?
- Ändern Sie den „Show Lessons Plan“-Button oben rechts und lassen Sie ihn stattdessen „XSS“ anzeigen. Dokumentieren Sie Ihr Vorgehen.

7 Injection Flaws

7.1 Command Injection

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

- Was ist unter Parameter Injection / Command Injection zu verstehen?
- Wie können Sie zusätzlich zur „type“-Anweisung einen anderen Befehl starten?
- Mit welcher Eingabe können Sie herausfinden, welche Version von Windows auf dem Rechner installiert ist?

Hinweis zu SQL Injection

Hinweis: haben Sie Numeric oder String SQL Injection gelöst, wechselt WebGoat in einen anderen Modus, in dem Sie aufgefordert werden, „parameterized queries“ zu bearbeiten – diese Aufgaben sind NICHT zu lösen. Durch Eingabe von „restart“ ins Textfeld können Sie die Aufgaben zurücksetzen, um Ihre Lösung erneut durchzugehen oder Aufgabe 7.5 zu probieren.

7.2 Numeric SQL Injection

- Was ist unter „SQL Injection“ zu verstehen?
- Wie können Sie im konkreten Beispiel die gesamte Tabelle auslesen?
- Bisher operieren Sie nur auf einer bestimmten Tabelle. Wie könnte man mittels SQL Injection auch Daten von anderen Tabellen auslesen?

7.3 Blind SQL Injection

- Was ist unter „Blind SQL Injection“ zu verstehen?
- Erklären Sie Ihr konkretes Vorgehen, um den Wert des Feldes „first_name“ für den User mit „userid“ = 15613 herauszufinden.

7.4 String SQL Injection

- Erklären Sie Ihr Vorgehen zur Lösung der Aufgabe.

7.5 Advanced SQL Injection (schwierig)

- 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

- Beschreiben Sie, was unter einer „Fail Open“-Problematik zu verstehen ist.
- Wie können Sie diesen Zustand in diesem Beispiel erreichen/ausnutzen? Beschreiben Sie Ihr Vorgehen.

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

9 Denial of Service

9.1 DOS Multiple Login

- Was ist eine DOS Attacke?
- Beschreiben Sie Ihren Lösungsweg, um die gesuchten Usernamen/Passwörter anzuzeigen.

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

- Was ist SOAP (kurz)?
- Was ist WSDL (kurz)?
- Wie können Sie die vom Web Service angebotenen Methoden in Erfahrung bringen?
- Welchen Typ hat die "getFirstNameRequest"-Methode?
- Erklären Sie Ihren Lösungsweg zum Aufrufen einer der Operationen.

10.2 WSDL Scanning

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

10.3 Web Service SQL Injection

- Erklären Sie Ihren Lösungsweg zur Erfüllung der gestellten Aufgabe.

[1] WebScarab: <http://www.owasp.org/software/webscarab.html>

Framework zur Analyse von Anwendungen, die über HTTP and HTTPS Protokolle kommunizieren.

Kurzanleitung: Sie brauchen insbesondere den Proxy Teil, „Manual Edit“ und „Miscellaneous“. Der Proxy lauscht an einer vorgegebenen Adresse am Localhost, Sie müssen Ihren Browser entsprechend konfigurieren, um über den Proxy auf Webseiten zuzugreifen. Wichtig: der Proxy muss auch bei lokalen Seiten zur Anwendung kommen.