Webseiten- & Datenbanksicherheit (Langfassung)

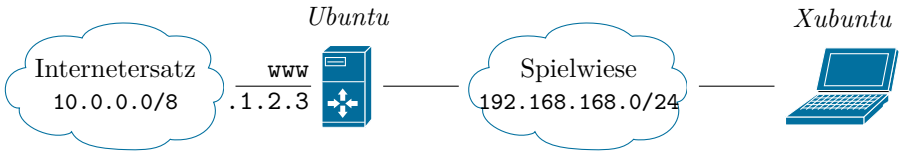
Lösen Sie die nachfolgenden Aufgaben und bereiten Sie diese bis zum nächsten Lehrveranstaltungstermin vor.

01.

a) Konfigurieren Sie den Webserver auf dem *Ubuntu*-Server so, dass die IP der in der Grafik unten abgebildeten entspricht und der Webserver alle Anfragen der Domäne bearbeitet. Folgen Sie den Hinweisen zur Einrichtung.

b) Kopieren Sie die zur Verfügung gestellten Beispieldateien ins Verzeichnis `/var/www` auf dem *Ubuntu*-Server. Überprüfen Sie anschließend die Funktionsfähigkeit der Webseite, indem Sie sie in der *Xubuntu*-VM mit *Firefox* öffnen und alle Links auf der Startseite einmal anklicken.
Hinweis: Sie können im Rahmen dieses Laborblockes den Befehl `scp` *auf der* Xubuntu-*VM verwenden, um Dateien auf den* Ubuntu-*Server zu kopieren. Beachten Sie allerdings die fehlenden Zugriffsrechte des Benutzers* `bruce` *auf das Zielverzeichnis. Kopieren Sie daher die Dateien zuerst nach* `/tmp` *und verschieben Sie sie von dort als Benutzer* `wartung` *auf dem* Ubuntu-*Sever mit dem Befehl* `mv` *ins Zielverzeichnis.*

c) Überprüfen Sie Möglichkeit von HTML bzw. JavaScript Injection, indem Sie in das Eingabefeld der eingerichteten Seite `top.html` in *Firefox* in der *Xubuntu*-VM `<em>Test</em>` bzw. `<script language="javascript"> alert("Hallo Welt");</script>` eingeben. Erläutern Sie die Sicherheitsimplikationen dieser beiden Angriffsvektoren.



Domäne `spielwiese-<Nachname>.tld.`, z.B. `spielwiese-unterweger.tld.`

02.

a) Erweitern Sie die JavaScript Injection aus der vorherigen Aufgabe derart, dass ein sich standardmäßig nicht verändernder Teil der angezeigten Webseite nach der Eingabe durch eine von Ihnen gewählte Zeichenkette ersetzt wird.
Hinweis: Sie können in Firefox *Objekte auf einer Webseite per Rechtsklick mit dem Kontextmenüeintrag* Inspect Element *untersuchen und über einen weiteren Rechtsklick auf die markierte Zeile in der Codeansicht mit dem*

Kontextmenüeintrag Copy\CSS Selector *eine Zeichenkette als Parameter für den Aufruf der Funktion* `document.querySelector` *erzeugen lassen.*

b) Führen Sie eine kombinierte HTML+JavaScript Injection durch, bei der Sie als Eingabe ein Bild (`img`-Tag) mit ungültigem Pfad angeben, das `0px` breit und hoch ist. Führen Sie im `onerror`-Ereignis des Bildes eine Funktion aus, die – wie oben – einen Teil der Webseite durch Text ersetzt. Diskutieren Sie die Sicherheitsimplikationen dieser Art von Angriff.

c) Wiederholen Sie die HTML+JavaScript Injection und ersetzen Sie bei Ihrer Eingabe alle möglichen Zeichen durch ihre jeweiligen Entity Numbers, z.B. `a` durch `&#97;` laut ASCII-Tabelle. Diskutieren Sie, welche Teile der Eingabe damit verschleiert werden können, ohne dass die Funktionsfähigkeit des Angriffes beeinträchtigt wird und diskutieren Sie die Implikationen dieser Eingabeform für Angriffe über Social Engineering.

03.

a) Erstellen Sie in der *Xubuntu*-VM eine HTML-Seite mit dem Dateinamen `malicious.html` in `/var/www/html`, die einem Nachbau der Ergebnisseite von `top.html` entspricht, aber eine zusätzliche Fehlermeldung sowie ein Passworteingabefeld enthält. Überprüfen Sie, ob die erzeugte HTML-Seite von anderen Rechnern (z.B. aus der *Windows*-VM) abrufbar ist und – falls nicht – starten Sie den Webserver auf der *Xubuntu*-VM mit dem Befehl `sudo service apache2 start`.

b) Führen Sie erneut eine JavaScript Injection über das Eingabefeld auf `top.html` durch, bei der nach dem Anzeigen der Ergebnisseite `malicious.html` vom Webserver der *Xubuntu*-VM geladen wird. Diskutieren Sie, welche URL in der Adresszeile angezeigt wird.

c) Wiederholen Sie das Ersetzen der Ergebnisseite per JavaScript Injection, laden Sie aber nur den `body` der Seite asynchron (d.h. per `XMLHttpRequest`) nach, um Cross-Site Request Forgery zu versuchen. Diskutieren Sie anhand der *Same-Origin-Policy*, warum dieser Angriff nicht funktioniert.
Hinweis: Verwenden Sie die Entwicklerwerkzeuge von Firefox*, um JavaScript-Fehlermeldungen anzuzeigen und die übertragenen HTTP-Header zu analysieren.*

d) Wiederholen Sie den XSS-Versuch per JavaScript Injection mit der zur Verfügung gestellten Datei `malicious.php` an Stelle von `malicious.html` auf dem Webserver in der *Xubuntu*-VM. Diskutieren Sie die Änderungen im HTTP-Header und die Sicherheitsimplikationen dieser Art von Angriff.

e) Erstellen Sie auf dem *Ubuntu*-Server eine Kopie von `customize_v1.php`, z.B. `customize_v2.php`, in der Sie `$_POST['name']` mit dem Aufruf der Funktion `htmlspecialchars` umschließen. Modifizieren Sie außerdem `top.html` entsprechend. Wiederholen Sie einige Ihrer bisherigen Angriffe und diskutieren Sie, inwieweit diese nun verhindert werden können.

04.

a) Loggen Sie sich mit dem Befehl `mysql -p<Passwort>` (z.B. `mysql -pwartung`) auf dem *Ubuntu*-Server in die SQL-Konsole ein und legen Sie eine Datenbank namens `Spielwiese` an. Legen Sie in dieser Datenbank eine Tabelle `products` an, die zumindest über die beiden Spalten `name` und `description` verfügt und überdies zumindest eine weitere Spalte mit nichtöffentlichen Zusatzinformationen (z.B. Einkaufspreis) beinhaltet. Befüllen Sie die Tabelle mit mindestens drei Datensätzen.

b) Überprüfen Sie die korrekte Einrichtung der Datenbank(-Tabelle) über das Suchformular auf der Beispielseite `search.html` in einem Browser.
Hinweis: Beachten Sie die Hinweise zur Verwendung von PHP, um temporär Fehlermeldungen bei Seitenaufrufen anzeigen zu lassen.

c) Überprüfen Sie Möglichkeit von SQL Injection, indem Sie in das Eingabefeld auf der eingerichteten Seite `search.html` in *Firefox* in der *Xubuntu*-VM `" OR 1=1 OR "` bzw. `" OR "*"; --` (beachten Sie das zusätzliche Leerzeichen am Ende!) eingeben. Beschreiben Sie, welche Datenbankabfrage im Hintergrund effektiv ausgeführt wird und erläutern Sie die Sicherheitsimplikationen dieses Angriffsvektors.

05.

a) Modifizieren Sie die SQL Injection aus der vorherigen Aufgabe derart, dass die ursprüngliche Abfrage mittels `UNION ALL` mit einem Tupel, bestehend aus dem Datenbanknamen (Funktion `database`) und dem Inhalt der Datei `/etc/passwd` (mit der Funktion `LOAD_FILE`), verknüpft wird. Diskutieren Sie die Sicherheitsimplikationen der Verfügbarkeit dieser Funktionen.

b) Führen Sie eine weitere SQL Injection durch, über die Sie zuerst die Namen aller (relevanten) Tabellen und anschließend deren Spalten ermitteln. Verwenden Sie dazu erneut `UNION ALL` sowie die (System-)Tabelle `INFORMATION_SCHEMA.TABLES` bzw. `INFORMATION_SCHEMA.COLUMNS`. Grenzen Sie den Datenbank- bzw. Tabellennamen (`TABLE_SCHEMA` bzw. `TABLE_NAME`) anhand der gesammelten Informationen mit `WHERE`-Klauseln ein.
Hinweis: Versuchen Sie zur Übung zuerst entsprechende Abfragen in der SQL-Konsole auf dem Ubuntu-*Server durchzuführen.*

c) Nutzen Sie die bisher erlangten Informationen, um per SQL Injection nichtöffentliche Informationen (der Tabelle `products`) auszugeben. Stellen Sie dabei sicher, dass diese den öffentlichen eindeutig zuordenbar sind. Diskutieren Sie die Sicherheitsimplikationen der gesamten Angriffskette.

d) Erstellen Sie auf dem *Ubuntu*-Server eine Kopie von `search_v1.php`, z.B. `search_v2.php`, in der Sie `$_POST['query']` mit dem Aufruf der Funktion `mysql_real_escape_string` umschließen. Modifizieren Sie außerdem `search.html` entsprechend. Wiederholen Sie einige Ihrer bisherigen Angriffe und diskutieren Sie, inwieweit diese nun verhindert werden können.

Hinweis zur Einrichtung des Webservers auf einem *Ubuntu*-Server

Stellen Sie sicher, dass Sie eingeloggt sind und führen Sie alle nachfolgenden Befehle als Superuser aus. Öffnen Sie die Port-Konfigurationsdatei des Webservers mit dem Editor `nano`, indem Sie

```
nano /etc/apache2/ports.conf
```

eingeben.
Ändern Sie in der Zeile

```
1 NameVirtualHost *:80
```

den Platzhalter `*` durch die IP-Adresse, die der Webserver bedienen soll. Entfernen Sie außerdem die Zeile

```
1 Listen 80
```

Speichern Sie die Datei und beenden Sie `nano`.
Öffnen Sie anschließend die Konfigurationsdatei `/etc/apache2/apache2.conf` mit dem Editor `nano` und fügen Sie am Ende die Zeile

```
1 Listen <IP-Adresse>:80
```

ein, wobei `<IP-Adresse>` die oben angegebene IP-Adresse des Webservers ist. Speichern Sie die Änderungen und beenden Sie `nano`.
Öffnen Sie anschließend die Datei `/etc/apache2/sites-enabled/000-default` zur Standardwebseitenkonfiguration mit `nano`. Ändern Sie `VirtualHost *:80` in der ersten Zeile wie oben beschrieben ab. Fügen Sie anschließend vor der letzten Zeile (`</VirtualHost>`) die folgenden beiden Zeilen ein:

```
1 ServerName <Servername>.<Domänenname>
2 ServerAlias <Domänenname>
```

`<Servername>` und `<Domänenname>` müssen dabei durch den Rechner- bzw. den Domänennamen ersetzt werden. Die spitzen Klammern sind **nicht** zu inkludieren. Beispiel:

```
1 ServerName www.spielwiese-unterweger.tld
2 ServerAlias spielwiese-unterweger.tld
```

In dieser Beispielkonfiguration bedient der Webserver alle Anfragen an den Servernamen `www.spielwiese-unterweger.tld`, bearbeitet aber auch Anfragen an die gesamte Domäne, d.h. ohne Angabe des Servernamens.
Speichern Sie die Datei und beenden Sie `nano`.
Öffnen Sie abschließend die HTTP-Konfigurationsdatei `/etc/apache2/httpd.conf` mit `nano` und fügen Sie die Zeile

```
1 ServerName localhost
```

hinzu, damit der Webserver seinen eigenen Namen nicht über ein Reverse DNS Lookup ermittelt, sondern fix auf dem lokalen Rechnernamen belässt. Speichern Sie die Datei und beenden Sie `nano`.
Nach der Konfiguration muss der Webserver neu gestartet werden:

```
/etc/init.d/apache2 restart
```

Abschließend kann mit dem Befehl

```
netstat -4at
```

überprüft werden, ob nur ein HTTP-Port auf der gewünschten IP-Adresse offen ist und keine weiteren IP-Adressen bedient werden.

Hinweis zur Verwendung von PHP auf einem als Webserver konfigurierten *Ubuntu*-Server

Um Fehlermeldungen bei der Erstellung von Seiten mit PHP anzeigen zu lassen, muss die Konfigurationsdatei `/etc/php5/apache2/php.ini` mit `nano` geöffnet werden und die Zeile

```
1 display_errors = Off
```

durch

```
1 display_errors = On
```

ersetzt werden. Nach dem Speichern der Datei und dem Beenden von `nano` muss der Webserver – wie oben beschrieben – neu gestart werden.