Software Security

Kryptographie und IT Sicherheit SS 2018

Dimitrii ,Manuel Klappacher

Universität Salzburg

Themen

1. Einleitung
2. Remote and Lokale Gefahren
3. Exploits
4. Open Source und Propertäre Software
5. Firmware Security

Einleitung

Einleitung

test frame

Remote and Lokale Gefahren

Exploits

Code Injection

Code Injection ist das ausnutzen von Bugs durch Eingabe von ungewollten Parametern, um dadurch die Ausfürung zu verändern. Kann folgende Auswirkungen haben:

- Daten in SQL Tabellen verndern
- Installieren von Malware durch Server-Scripting Code zB. PHP
- Root Privilegien bekommen, durch Shell Injection oder Windows Service
- Angtiff auf Web User durch Cross-Site-Scripting in HTML/JS

Code Injection - Masnahmen

Kann erschwert werden durch:

- API's benutzen, die sicher gegenüber allen Symbolen sind, indem der Eingabestring compiliert und gefiltert wird.
- Whitelisting von erwünschten Parametern

SQL Injection

Ausnutzen von Sicherheitslücken in Zusammenhang mit SQL-Datenbanken. Ziele:

- Daten auszuspähen oder zu verändern
- Kontrolle über Server zu erhalten

SQL Injection - Beispiel

Es wird zusätzlicher Code bei Aufruf eingeschleust, der die Bentzertabelle modifiziert.

	Erwarteter Aufruf
Aufruf	`http://webserver/cgi-bin/find.cgi?ID=42`
Erzeugtes SQL	`SELECT author, subject, text FROM artikel WHERE ID=42;`
	SQL-Injection
Aufruf	`http://webserver/cgi-bin/find.cgi?ID=42;UPDATE+USER+SET+TYPE="admin"+WHERE+ID=23`
Erzeugtes SQL	`SELECT author, subject, text FROM artikel WHERE ID=42;UPDATE USER SET TYPE="admin" WHERE ID=23;`

Cross Site Scripting

test frame

Cross Site Scripting - reflektierte Angriffe

Eine Benutzereingabe wird direkt vom Server wieder zurück gesendet. Wenn diese Eingabe Scriptcode enthält, die vom Browser des Nutzers interpretiert wird, kann dorrt Schadcode ausgefürt werden. Beispiel: Suchfunktion.

```
http://example.com/?suche=Suchbegriff
http://example.com/?suche=<script type="text/javascript">alert("XSS")</script>
<p>Sie suchten nach: <script type="text/javascript">alert("XSS")</script></p>
```

Ausgenutzt wird das dynamisch generierte Websiten ihren Ihnalt an übergebene Eingabewerte anpassen, durch HTTP-GET und HTTP-POST. Dieser Typ heisst auch nicht-persistent, da der Schadcode nur temprär bei der jeweiligen Generierung der Website eingeschleust wird.

Cross Site Scripting - persistente Angriffe

Unterscheidet sich von reflektierenden Angriffen nur dadurch, dass der Schadcode auf dem Server gespeichert wird, wodurch er bei jeder Anfrage asugefürt wird. Ist bei Webanwendungen möglich, die Benutzereingaben serverseitig ohne Prüfung speichern und diese später wieder ausliefert. Beispiel Posting auf Website:

```
Eine sehr gutes Produkt!<script type="text/javascript">alert("XSS")</script>
```

Cross Site Scripting - DOM-basierte Angriffe

Webapplikation auf dem Server ist hier nicht beteiligt, wird auch lokales XSS genannt. Somit auch statische HTML Seiten mit JavaScript unterstützung anfÏlig für diesen Angriff.

Cross Site Scripting - Schutzmaßnahmen

- Anstatt Blacklist mit bs̈en Eingaben zu führen, besser Whitelist mit buten Eingaben. Da die Anzahl der Angriffmethoden nicht bekannt ist.
- HTML-Metazeichen durch Zeichenreferenzen ersetzen, damit sie als normale Zeichen behandelt werden
- Sicher programmierte Anwendung sind Web Apllication Firewalls (WAF) vorzuziehen.

Directory Traversal Attack

Ein HTTP Angriff, bei dem ein Angreifer zugriff auf gesperrte Verzeichnisse gewinnt und Code auserhalb des root Verzeichnisses ausführt.

Directory Traversal Attack - Schutzmaßnahmen

test frame

Format String

test frame

Format String - Schutzmaßnahmen

test frame

Buffer Overflows

test frame

Buffer Overflows - Type-Safe Sprachen

Compiler stellt Typsicherheit her, indem Datentypen geprüft werden, damit keine Typverletzungen entstehen. Wenn Typverletzungen spätestens zur Laufzeit erkannt werden, spricht man von Typsicheren Programmiersprachen.

Beispiel String in Python, es reicht der Variable einen String zuzuweisen.

```
1    mystring = "This is my string"
2
```

Beispiel in C, es muss der Typ deklariert und auch der Speicher manuell reserviert werden.

```
1    char mystring[20] = "This is my string";
2
```

Wenn man in C nun einen 30 Byte String zuweist entsteht eine Overflow Situation.

Buffer Overflows - Schutzmaßnahmen

- Type-Safe Programmiersprachen verwenden, welche Memory Management zB Java, Python, Ruby,...
- Überprüfen auf Overflows bei User Eingaben
- in C sichere Methoden verwenden, *get_s* anstatt *get*.

Open Source und Propertäre Software

Firmware Security

Spezialfall Firmware

- Gerät wurde bereits verkauft, kein Interesse des Herstellers an Updates
- Zu viele verschiedene Geräte - Unmöglicher Verwaltungsaufwand
  - Alleine Samsung hat bis 2014 56 verscheidene Smartphones pro Jahr herausgebracht
- Firmware agiert in Schicht unter Betriebssystem - Angriffe können vom Benutzer nicht erkannt oder verhindert werden
- Firmware meist Closed Source - keine Weiterentwicklung der Community

Sicherheitslücken in Firmware - Beispiele

- BadUSB - Eingabegeräte, USB-Sticks, Speichermedien, Kameras, ...
- Intel ME - Betriebssystem im Prozessor (AMD PSP)
  - Funktionsweise undokumentiert
  - Kritische Lücke 2017 entdeckt
  - NSA und Google haben Intel ME abgeschaltet auf ihren Geräten
- Android
  - praktisch alle Android Geräte ohne Sicherheitupdates
- Router, Smart TV's, IoT-Devices - Millionen angreifbare Geräte in Haushalten, Firmen und Behörden