Semesterprüfung APSI 2018, FHNW 17.12.2018

Dozent: Dr. Arno Wagner

Kandidat

Name: _____

Vorname: _____

Studentennummer: _____

Aufgabe	1	2	3	4	5	6	7	Σ
Maximalpunkte	16	8	10	8	13	6	6	67
Erreichte Punkte								

Note: _____

1 Aufgabe: Vermischte Fragen (16 P)

1.1 Marking und Tagging (2 P)

Kann Taint-Checking Buffer-Overflows verhindern? Warum, bzw. warum nicht?

1.2 Buffer Overflows (2 P)

Nehmen Sie an, eine Sprache bei der Buffer gegen Overflow geschützt sind kommt zum Einsatz. Welche beiden typischen Implementierungstechniken werden hierfür bei den Buffern eingesetzt und welches Problem verursacht die eine, welches die andere Stragie?

1.3 Programmiersprachen (2 P)

In vielen Programmiersprachen sind Buffer-Overflows nicht möglich. Warum sind Buffer-Overflows trotzdem auch heute ein grosses Problem?

1.4 Input Validierung und Normalisierung (2 P)

Unter welchen Umständen können Sie Input Normalisierung ohne vorgeschaltete Input Validierung einsetzen?

1.5 Fehlerbehandlung (2 P)

Um Data-Leakage durch Verhalten zu Vermeiden, bricht eine Webapplikation bei jedem Fehler die Verbindung ab und akzeptiert auch keine Cookies aus älteren Verbindungen mehr. Ist dies eine effektive Strategie? Warum, bzw. warum nicht?

1.6 Zufallserzeugung (2 P)

Nehmen Sie an, Sie haben einen Hardware-Zufallsgenerator mit einem kompromittierten Design (z.B. Intel RDRAND) zur Verfügung. Können Sie diesen trotzdem sinnvoll und sicher einsetzen? Wie geht das, bzw. warum geht das nicht?

1.7 Scripting Attacken (2 P)

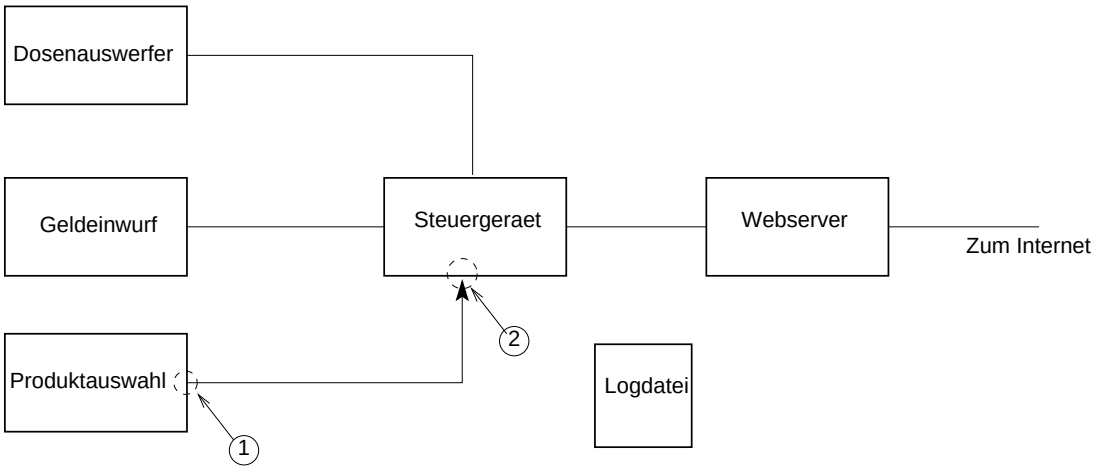
Können Sie durch das entfernen aller `<script>` Tags Stored-XSS Angriffe zuverlässig verhindern?

1.8 Application States (2 P)

Können Sie zuverlässig erkennen wenn ein in eine Web-Applikation eingeloggter User diese durch Schliessen des Tabs oder des Browsers verlässt? Wie, bzw. warum nicht?

2 Aufgabe: Privilege-Separation (8 P)

Gegeben sei folgendes System, das einen sehr einfachen Getränkedosenautomaten abstrakt modelliert:



Am Bedienmodul wählen Sie mit einer Taste das gewünschte Getränk aus, dann bezahlen Sie, und dann wird das Getränk ausgegeben. Da der Hersteller sparen muss, gibt es kein Rückgeld. Für einen Webserver, mit dem der Dosenbestand über das Internet abgefragt werden kann, hat es aber gerade so noch gereicht.

2.1 Rechte (3 P)

Markieren Sie jeden Punkt im Schema, an dem Rechte zu bestimmen sind und numerieren sie diese durch. Bestimmen sie die zugehoreigen Rechte und geben Sie die Richtung des Datenflusses durch Pfeile an. Als Beispiel ist der Austausch zwischen Produkauswahl und Steuergerät bereits vollständig markiert.

2.2 Validierung (2 P)

Wo wird hier Input Validierung benötigt? Wird Input Normalisierung benötigt?

2.3 Logging (3 P)

Welche Logs schreiben sie bei diesem Gerät sinnvollerweise und welche Komponenten machen das? Zeichnen sie entsprechende Pfeile ein und beschreiben sie kurz was jeweils geloggt wird.

3 Aufgabe: Passworte (10 P)

3.1 Passwort-Sicherheit (4 P)

Gegeben sei die folgene ”Eingabehilfe” für Passworte: Wenn der Benutzer versehentlich eine benachbarte Taste statt der richtigen Taste erwischt, dann wird die Eingabe korrigiert. Das wird für alle Stellen des Passwortes automatisch durchgeführt.
Es werden nur Passworte aus Kleinbuchstaben und den Ziffern 0-5 benutzt. Passworte sind 8 Zeichen lang. Zur Vereinfachung wird davon ausgegangen, dass jede Taste genau 7 relevante benachbarte Tasten hat.
Wieviel Entropie hätte ein solches zufaelliges Password mit diesen Daten ohne Eingabehilfe? Wieviel Entropie bleibt bei Einsatz der Eingabehilfe noch übrig?

3.2 Passwort-Sicherheit (2 P)

Wie beurteilen Sie die Sicherheit des Passwortschemas aus der ersten Teilaufgabe ohne, bzw. mit Eingabehilfe?

3.3 Passwort-Sicherheit (2 P)

Welches Problem ergibt sich für die Speicherung der Passworte wenn ein Eingabehilfe wie in der ersten Teilaufgabe vorgesehen wird?

3.4 Passwort Sicherheit (2 P)

Geben Sie eine Moeglichkeit an, wie ein Passwort mit nur 1'000'000 verschiedenen moeglichen Werten trotzdem sicher gegen Ausprobieren (d.h. Brute-Force Attacken) gemacht werden kann.

4 Aufgabe: Fehlerbehandlung (8 P)

4.1 Passwort-Eingabe (2 P)

Ist ein falsches oder falsch eingegebenes Passwort ein Fehler im Sinne der Fehlerbehandlung? Warum bzw. warum nicht?

4.2 Passwort-Eingabe (2 P)

In welches Log schreiben Sie wegen falschem Passwort gescheiterte log-in Versuche? Sollten Sie diese überhaupt loggen, oder ist das unproblematisch?

4.3 Fuzzing (4 P)

Ist Fuzzing geeignet um Verwundbarkeiten gegenüber Angriffen die algorithmische Komplexitaet ausnutzen zu ermitteln? Warum bzw. warum nicht?

5 Aufgabe: Web-Application Security (13 P)

5.1 Cookies (2 P)

Müssen zum sicheren Betrieb einer Web-Applikation alle Cookies geschützt werden oder nur bestimmte? Begründen Sie Ihre Antwort.

5.2 Cookies (2 P)

Können Sie sicherstellen, daß ein gesetzter Cookie auch wieder an den Server geschickt wird? Wie, bzw. warum geht das nicht?

5.3 Cookies (2 P)

Wenn ein Cookie mit ”HttpOnly” gesetzt wird, wo findet Authorization statt und wo Access Control?

5.4 Ausführungsumgebung (3 P)

Bei einer Web-Applikation mit Client-Seitigem JavaScript Code haben Sie keine Kontrolle über die Ausfuehrungsumgebung (Browser). Was sind die Implikationen fuer die Applikationssicherheit?

5.5 Ausführungsumgebung (4 P)

Können Sie Daten, die der Benutzer nicht sehen oder veraendern darf sicher und zuverlaessig im Webbrowser-State (als auf Client Seite) abspeichern? Wie geht das und wo sind die Grenzen?

6 Aufgabe: OAuth and Friends (6 P)

6.1 Implementierungsfehler (2 P)

Wellche der Vorgänge Identifizierung, Authentifizierung und Autorisierung werden von OAuth durchgeführt (Standardszenario, keine Zusatzfunktionen)? Jeweils mit Begründung.

6.2 Implementierungsfehler (4 P)

Beschreiben Sie zwei typische Fehler bei der Implementierung von OAuth und begründen Sie warum diese die Sicherheit von OAuth zerstören.

7 Aufgabe: Coding (6 P)

7.1 Style Guides (3 P)

Wenn Sie die native Implementierung von Java Funkionalitäten in C mittels JNI mit einem Style-Guide unterstützen wollen, brauchen Sie dann einen Style-Guide für Java oder einen für C oder einen anderen und warum?

7.2 Sichere Sprachen (3 P)

Wenn Sie eine Programmiersprache einsetzen, die keine Buffer-Overflows erlaubt, können Sie dann auf eine entsprechende Ausbildung der Programmierer verzichten, oder sollten diese den Mechanismus des Buffer-Overflows eben doch verstehen? Warum, bzw. warum nicht?