

**NAME DES DOZENTEN: BJÖRN KIMMINICH**

**KLAUSUR: IT-SICHERHEIT / A106**

**QUARTAL: (3/2018)**

Name des Prüflings: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Zenturie: \_\_\_\_\_

Dauer : 90min

Datum: 04.10.2018

Seiten der Klausur **mit** Deckblatt: 11

Hilfsmittel: Taschenrechner

Bemerkungen:

- **Bitte prüfen Sie zunächst die Klausur (alle Teile) auf Vollständigkeit**
- **Bitte lösen Sie nicht die Heftung**

Es sind 120 Punkte erreichbar!

Zum Bestehen der Klausur sind 60 Punkte ausreichend!

Aufgabe	Erreichbare Punkte	Erreichte Punkte
1	10	
2	13	
3	19	
4	12	
5	23	
6	15	
7	10	
8	8	
9	10	
<b>Summe</b>	<b>120</b>	

Note: \_\_\_\_\_

Prozentsatz: \_\_\_\_\_

Ergänzungsprüfung: \_\_\_\_\_

Datum: \_\_\_\_\_

Unterschrift: \_\_\_\_\_

Datum: \_\_\_\_\_

Unterschrift: \_\_\_\_\_

## Aufgabe 1 (10 Punkte)

	Beantworten Sie die folgenden Fragen:	stimme zu	stimme nicht zu
a)	Eine Whitelist erlaubt alles, was nicht explizit verboten ist.		<input checked="" type="checkbox"/>
b)	Unsichtbare Tinte ist eine frühe Form von Steganographie.	<input checked="" type="checkbox"/>	
c)	Schwachstellen in IT-Systemen basieren immer auf Bugs in der Software.		<input checked="" type="checkbox"/>
d)	Die Verwendung eigener Verschlüsselungsalgorithmen erhöht die Sicherheit.		<input checked="" type="checkbox"/>
e)	Das Signieren von Nachrichten schützt deren Vertraulichkeit.		<input checked="" type="checkbox"/>
f)	PGP wird zur Verschlüsselung und Signierung von Emails und Daten verwendet.	<input checked="" type="checkbox"/>	
g)	Trojanische Pferde verbreiten sich eigenständig von System zu System.		<input checked="" type="checkbox"/>
h)	Next-Generation Firewalls schützen Systeme vor Zero-Day Exploits.	<input checked="" type="checkbox"/>	
i)	Watermarking ist eine Möglichkeit der Urheberkennzeichnung in Dokumenten.	<input checked="" type="checkbox"/>	
j)	Die Integrität kann nur durch aktive Angriffe beeinträchtigt werden.	<input checked="" type="checkbox"/>	

Jede richtige Antwort wird mit 1 Punkt, jede falsche oder nicht gegebene Antwort mit 0 Punkten bewertet.

## Aufgabe 2 (13 Punkte)

(2.1) (3 Punkte) Nennen Sie die drei klassischen Schutzziele der IT-Sicherheit. Nennen Sie zu jedem Schutzziel beispielhaft eine mögliche Maßnahme zu dessen Erreichung.

Schutzziel der IT-Sicherheit	Mögliche Maßnahme zur Erreichung
Confidentiality	End-to-End / Rechte beam
Integrity	Prüfsumme
Availability	Redundanz / Backups

(2.2) (3 Punkte) Nennen Sie drei mögliche Einsatzzwecke für ein Botnet.

E-Mail-Spam	DDOS	Bitcoin
-------------	------	---------

(2.3) (4 Punkte) Nennen Sie jeweils zwei konkrete Anwendungsfälle von Verbindungsverschlüsselung sowie Ende-zu-Ende-Verschlüsselung.

Verbindungsverschlüsselung	Ende-zu-Ende-Verschlüsselung
Https	Whats app
SSH	Backups

(2.4) (3 Punkte) Grenzen Sie Implementierungsfehler (*Bugs*) gegen Mängel im Entwurf (*Design Flaws*) ab und erläutern Sie welcher von beiden Mängeln schwerwiegender für die Sicherheit eines IT Systems ist.

### Aufgabe 3 (19 Punkte)

(3.1) (13 Punkte) Ergänzen Sie alle fehlenden Informationen in der Risikoklassifizierungstabelle der OWASP Top 10 Liste von 2017.

Risk	Exploitability	Prevalence	Detectability	Tech. Impact	Score
A1 – <i>Injection</i>	Easy (3)	Common (2)	Easy (2)	Severe (3)	<i>7.0</i>
A2 – Broken Authentication	Easy (3)	Common (2)	Average (2)		7.0
A3 – Sensitive Data Exposure	Average (2)	<i>2</i>	Average (2)	Severe (3)	7.0
A4 – XML External Entities (XXE)	Average (2)	Common (2)	Easy (3)	Severe (3)	<i>7</i>
A5 – Broken Access Control	Average (2)	Common (2)	Average (2)	Severe (3)	<i>6</i>
A6 – Security Misconfiguration	Easy (3)	Widespread (3)	Easy (3)	<i>2</i>	6.0
A7 – <i>XSS</i>	Easy (3)	<i>3</i>	Easy (3)	Moderate (2)	6.0
A8 – Insecure Deserialization	Difficult (1)	Common (2)	Average (2)	<i>2</i>	5.0
A9 – Vulnerable Components	Average (2)	<i>3</i>	Average (2)	<i>2</i>	4.7
A10 – Insufficient Logging&Monitoring	Average (2)	Widespread (3)	Difficult (1)	Moderate (2)	<i>4</i>

(3.2) (3 Punkte) OWASP empfiehlt Unternehmen eine über die generische Klassifizierung hinaus gehende Bewertung der Top 10 Risiken nach anwendungsspezifischen Bedrohungen sowie dem Business Impact. Erläutern Sie die Vorteile dieser Vorgehensweise.

(3.3) (3 Punkte) Warum sollten die OWASP Top 10 als reines Awareness-Dokument und nicht als Checkliste für Sicherheitsanforderungen betrachtet werden?

#### Aufgabe 4 (12 Punkte)

(4.1) (4 Punkte) Grenzen Sie Whitelist- und Blacklist-Validierung gegeneinander ab und nennen Sie Vor- und Nachteile beider Verfahren. Welches ist in den meisten Fällen zu bevorzugen?

(4.2) (4 Punkte) „Jeder Output, der ursprünglich aus vom User beeinflussbarem Input entstanden ist, sollte in einer Webanwendung ein Encoding durchlaufen.“ – Beziehen Sie Stellung zu dieser Aussage und untermauern oder widerlegen Sie sie mit Beispielen.

(4.3) (4 Punkte) Grenzen Sie Reflected XSS und DOM-based XSS gegeneinander ab. Geben Sie Beispiele für Technologien oder Frameworks an, in denen man die jeweiligen XSS-Varianten vorfinden könnte.

## Aufgabe 5 (23 Punkte)

Ein einfacher WYSIWYG-Texteditor in einer web-basierten Content Management System (CMS) unterstützt folgende Formatierungen und speichert diese im HTML-Format in der Datenbank ab:

Formatierung	Gespeichertes HTML
Fett	<strong>Text</strong>
Kursiv	<em>Text</em>
Hyperlinks	<a href="URL">Text</a>
Horizontale Trennlinie	<hr/>
Zeilenumbrüche	 

Auf Knopfdruck ist das Wechseln in eine HTML-Editor Ansicht möglich. Die gespeicherten Texte werden auf einer News-Webseite als Artikel-Teaser dargestellt und sehen häufig wie folgt aus:

**Reißerische Artikel-Überschrift**  
*von irgendeinem Autor*

---

Hier steht eine kurze Zusammenfassung des Artikels, die den User zum Weiterlesen animieren möchte.

[Zum Weiterlesen hier klicken](#)

Es gibt jedoch einige „Power-User“ unter den Autoren, die über die HTML-Editor Ansicht auch Bilder (``) einfügen und die alten HTML-Tags `<b>Text</b>` und `<i>Text</i>` für fett bzw. kursiv hervorgehobenen Text verwenden. Die „Power-User“ werden vom CMS-Betreiber toleriert, denn es sei leider technisch nicht möglich, im WYSIWYG-Modus das Einfügen von Bildern zu ermöglichen, da es sich um eine Legacy-Komponente des CMS handelt. Weitere HTML-Tags oder -Attribute sind nicht erwünscht. Ein Austausch des gesamten Editors ist aus finanziellen Gründen unrealistisch.

(5.1) (8 Punkte) Schlagen Sie dem Betreiber des CMS alle denkbaren Optionen zur technischen Absicherung des Editors gegen Missbrauch vor. Erläutern Sie, welche davon Sie bei den gegebenen Rahmenbedingungen für den besten Kompromiss aus Sicherheit und Benutzerfreundlichkeit halten.

- die alten HTML-Tags werden nicht mehr unterstützt
- Hyperlinks `<a>` statt Power User `<i>` Tag
- whitelist/input-sanitizer  $\rightarrow$  src von `<img>` tags auf übliche Bilddatei Endungen überprüfen
- `<php>` tag verbieten, sonst Cross-Site Scripting möglich

(5.2) (3 Punkte) Beschreiben Sie die Aufgabe eines Input-Sanitizers. Erläutern Sie anhand eines Beispiels, welche Kriterien dieser erfüllen muss, um sicher gegen Angriffe zu sein.

- aktuelle Liste
  - Vor der Weiterleiten an den Server → Vor der Auswertung
  - Muss Böse Sachen verbieten
- <script>

(5.3) (5 Punkte) Ein Entwickler des CMS Betreibers hat einen HTML-Sanitizer für den beschriebenen WYSIWYG-Editor implementiert. Die folgende Tabelle enthält die Test Cases des Entwicklers zu verschiedenen Inputs. Kreuzen Sie an, ob die jeweilige Sanitization als sicher oder unsicher zu betrachten ist.

#	Input	Output	sicher	unsicher
1	<pre>&lt;strong&gt;Titel&lt;/strong&gt;&lt;br&gt; &lt;em&gt;von Localhorst&lt;/em&gt;&lt;hr&gt;&lt;br&gt; TextTextText &lt;br&gt;&lt;br&gt; &lt;a href="http://cms.de"&gt;Link&lt;/a&gt;</pre>	<pre>&lt;strong&gt;Titel&lt;/strong&gt;&lt;br&gt; &lt;em&gt;von Localhorst&lt;/em&gt;&lt;hr&gt;&lt;br&gt; TextTextText &lt;br&gt;&lt;br&gt; &lt;a href="http://cms.de"&gt;Link&lt;/a&gt;</pre>	X	
2	<pre>&lt;b&gt;Titel&lt;/b&gt;&lt;br&gt; &lt;i&gt;von Localhorst&lt;/i&gt;&lt;hr&gt;&lt;br&gt; &lt;a href="http://cms.de"&gt;   &lt;img src="http://cms.de/p.png"&gt; &lt;/a&gt;</pre>	<pre>&lt;b&gt;Titel&lt;/b&gt;&lt;br&gt; &lt;i&gt;von Localhorst&lt;/i&gt;&lt;hr&gt;&lt;br&gt; &lt;a href="http://cms.de"&gt;   &lt;img src="http://cms.de/p.png"&gt; &lt;/a&gt;</pre>	X	
3	<pre>&lt;h1&gt;Titel&lt;/h1&gt; &lt;h3&gt;von Localhorst&lt;/h3&gt;&lt;hr&gt;&lt;br&gt; TextTextText &lt;script&gt;alert("Test")&lt;/script&gt;</pre>	<pre>&lt;hr&gt;&lt;br&gt; TextTextText</pre>	X	
4	<pre>&lt;b&gt;Titel&lt;/b&gt;&lt;br&gt; &lt;i&gt;von Localhorst&lt;/i&gt;&lt;hr&gt;&lt;br&gt; &lt;a href="http://cms.de"&gt;   &lt;img src="http://cms.de/p_xs.png"     onmouseover="showImage('p.png') "&gt; &lt;/a&gt;</pre>	<pre>&lt;b&gt;Titel&lt;/b&gt;&lt;br&gt; &lt;i&gt;von Localhorst&lt;/i&gt;&lt;hr&gt;&lt;br&gt; &lt;a href="http://cms.de"&gt;   &lt;img src="http://cms.de/p_xs.png"     onmouseover="showImage('p.png') "&gt; &lt;/a&gt;</pre>		X
5	<pre>&lt;h1&lt;h2&gt;&gt;Titel&lt;/h2&gt;&gt;1&gt; &lt;u&gt;von Localhorst&lt;/u&gt;&lt;hr&gt;&lt;br&gt; &lt;small&gt;TextTextText&lt;/small&gt;</pre>	<pre>&lt;h1&gt;Titel&lt;/h1&gt; &lt;hr&gt;&lt;br&gt;</pre>		X

(5.4) (4 Punkte) Für alle unsicheren Bereinigungen aus 5.3 geben Sie einen beispielhaften Exploit an. Geben Sie außerdem an, was der Sanitizer gemäß Anforderungen und Sicherheitsaspekten hätte ausgeben müssen. Verweisen Sie bitte auf die jeweilige Nummer aus Spalte „#“ der obigen Tabelle.

Cookie  
X 4 <b>Titel</b><br>

X S  
Script  
in dem  
inneren  
der äusseren  
Tags  
Reverse  
Shell

(5.5) (3 Punkte) Empfehlen Sie dem Betreiber drei konkrete Maßnahmen zur Verbesserung der Sicherheit in seinem CMS.

Ein satz von Dast tools

Input Sanitizer

Next Generation Firewall

## Aufgabe 6 (15 Punkte)

```
import java.io.*;

public class PrintDirToConsole {
    public static void main(String[] args)
        throws IOException {
        if(args.length != 1) {
            System.out.println("Pass in a directory name!");
            System.exit(1);
        }
        Runtime runtime = Runtime.getRuntime();
        String[] cmd = new String[3];
        cmd[0] = "cmd.exe" ;
        cmd[1] = "/C";
        cmd[2] = "dir " + args[0];
        Process proc = runtime.exec(cmd);

        InputStream is = proc.getInputStream();
        InputStreamReader isr = new InputStreamReader(is);
        BufferedReader br = new BufferedReader(isr);

        String line;
        while ((line = br.readLine()) != null) {
            System.out.println(line);
        }
    }
}
```

→ Sicherheitslücke

(6.1) (3 Punkte) Skizzieren Sie kurz, was das obige Java-Programm bei Ausführung tut und wie man es im Sinne des Entwicklers korrekt aufruft.

- Darf nur mit einem Argument aufgerufen werden
- Das soll der directory name sein.



(6.2) (5 Punkte) Identifizieren Sie die Sicherheitslücke im obigen Java-Programm und erläutern Sie, wie diese ausgenutzt werden könnte. Geben Sie auch ein praktisches Beispiel für einen möglichen bösartigen Exploit an. (Kleinere Syntax-Fehler führen bei dieser Aufgabe nicht zu Punktabzug!)

Input wird nicht encoded/validiert

↓  
kein Plaintext

- Man könnte einen bösen Befehl anfügen der dann z.B. reverse shell öffnet.

(6.3) (7 Punkte) Korrigieren Sie die betroffenen Codestellen im obigen Programm so, dass ein Exploit nicht mehr möglich ist. (Kleinere Syntax-Fehler führen bei dieser Aufgabe nicht zu Punktabzug!)

2

### Aufgabe 7 (10 Punkte)

	Beantworten Sie die folgenden Fragen:	stimme zu	stimme nicht zu
a)	<del>Shodan ist eine KI, die beim Durchsuchen der CVE-Datenbank behilflich ist.</del>		
b)	Mobile-Apps aus inoffiziellen Quellen zu installieren ist meistens unbedenklich.		X
c)	Default-Accounts sollte man abschalten oder ihr Passwort zumindest ändern.	X	
d)	Stacktraces in client-seitigen Fehlermeldungen sind ungefährlich und nützlich.		X
e)	Skript Kiddies nennt man minderjährige Autoren von Profi-Hackingtools.		X
f)	Verärgerte Mitarbeiter gehören zu den potentiell gefährlichsten Angreifern.	X	
g)	Bei Reflected XSS führt das Opfer den Angriff gegen die Webanwendung aus.		X
h)	Bei Stored XSS wird der Schadcode vom Opfer versehentlich in der DB abgelegt.		X
i)	Eine XXE-Attacke funktioniert nur in Kombination mit einer XSS-Verwundbarkeit.		X
j)	Deserialisierungs-Mängel sind zwar schwierig auszunutzen aber sehr gefährlich.	X	

Jede richtige Antwort wird mit 1 Punkt, jede falsche oder nicht gegebene Antwort mit 0 Punkten bewertet.

→ XML-External Entities

### Aufgabe 8 (8 Punkte)

Erläutern Sie den Zweck und typischen Aufbau einer AppSec Pipeline. Erstellen Sie auch eine einfache schematische Darstellung solch einer Pipeline.

### Aufgabe 9 (10 Punkte)

Erklären Sie stichpunktartig die folgenden Prinzipien sicheren Designs:	
Fail securely	<ul style="list-style-type: none"><li>• throw errors not to user</li><li>• im zweifelsfall <sup>nicht</sup> ausführung</li><li>• default deny access</li></ul>
Fix Security Issues correctly	.
Keep Security simple	
Minimize Attack Surface Area	
Principle of Defense in Depth	verschiedene Ansätze

Viel Erfolg!