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:

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	
Summe	**120**	

Note:_____ Prozentsatz: _____ Ergänzungsprüfung:_____

Datum:_____ Unterschrift:_____

Datum:_____ Unterschrift:_____

Aufgabe 1 (10 Punkte)

	Beantworten Sie die folgenden Fragen:	**stimme zu**	**stimme nicht zu**
a)	Eine Blacklist erlaubt alles, was nicht explizit verboten ist.	X	
b)	YPS-Heft Geheimtinte ist eine einfache Form von Steganographie.	X	
c)	Schwachstellen in IT-Systemen basieren immer auf Mängeln im Softwaredesign.		X
d)	Selbst erfundene Verschlüsselungsalgorithmen gefährden die Sicherheit.	X	
e)	Das Verschlüsseln von Nachrichten schützt deren Vertraulichkeit.	X	X
f)	PGP wird ausschließlich zur Verschlüsselung von Emails und Daten verwendet.		X
g)	Würmer verbreiten sich eigenständig von System zu System.	X	
h)	Klassische Netzwerk-Firewalls schützen Systeme kaum vor Zero-Day Exploits.	X	
i)	Watermarking ist eine Möglichkeit der Verschlüsselung von Dokumenten.		X
j)	Die Integrität kann auch durch passive Angriffe beeinträchtigt werden.		X

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

9

Aufgabe 2 (13 Punkte)

(2.1) (4 Punkte) Nennen Sie die zwei Arten von Malware, sowie stichpunkthaft deren Wirkungsweise und Verbreitungsmethode.

Malware	**Wirkung / Verbreitung**
Virus	infiziert das jeweilige System; benötigt für die Verbreitung ein aktives Host-Programm, wodurch es auf den Computer gelangt
Wurm	infiziert das jeweilige System; verbreitet sich eigenständig über offene Netzwerkverbindungen

/2

(2.2) (2 Punkte) Nennen Sie zwei mögliche Methoden die Vertraulichkeit einer Nachricht sicherzustellen.

Rechtebeschränkung	Kryptographie

/2

(2.3) (4 Punkte) Nennen Sie jeweils zwei konkrete Angriffsformen passiver und aktiver Natur gegen IT Systeme.

Passiv	**Aktiv**
KeyLogger ✓	DDoS
Network Sniffing / VulnerabilityScan ↳ PortScan	XSS Reflective / Privilige Escalation/ Stride-Methoden → Spoofing

/3

(2.4) (3 Punkte) Grenzen Sie Implementierungsfehler (*Bugs*) gegen Mängel im Entwurf (*Design Flaws*) ab und erläutern Sie welcher von beiden Mängeln nach Go-Live der Anwendung leichter zu beheben ist.

Bei Bugs handelt es sich bspw. um Fehler im Softwarecode, wohin Mängel im Design schwergewichtiger sind, da somit der Entwurf betroffen ist und mögliche Fehlerquellen enthält. Bugs sind leichter zu beheben, da sie nur Code-Stellen betreffen.

Entwurfsfehler → Logikfehler

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 – Injection	Easy (3)	Common (2)	Easy (2)	3	7
A2 – Broken Authentication	Easy (3)	Common (2)	Average (2)	Severe (3)	7.0
A3 – Sensitive Data Exposure	Average (2)	Widespread (3)	Average (2)	3	7.0
A4 – XML External Entities (XXE)	Average (2)	Common (2)	Easy (3)	3	7.0
A5 – Broken Access Control	Average (2)	Common (2)	Average (2)	3	6.0
A6 – Security Misconfiguration	Easy (3)	3	Easy (3)	Moderate (2)	6.0
A7 – Cross-Site Scripting (XSS)	Easy (3)	3	Easy (3)	Moderate (2)	6.0
A8 –	1	Common (2)	Average (2)	Severe (3)	5.0
A9 – Vulnerable Components	Average (2)	3	Average (2)	2	4.7
A10 – Insufficient Logging&Monitoring	Average (2)	Widespread (3)	1	Moderate (2)	4.0

13/13

(3.2) (3 Punkte) OWASP bedient drei primäre Zielgruppen mit seinen Projekten. Nennen und beschreiben Sie diese kurz.

technische Entscheider Entscheide, welche Software und APIs verwendet werden sollen	Betriebs- und Sicherheitsverantwortliche - sicherzustellen, dass Appl. sicher sind - sicherstellen, dass es läuft	Appl. - Betroffene es wo schwachen sind und beheben werden kann

(3.3) (3 Punkte) Für welche der drei Zielgruppen ist der OWASP Juice Shop ein nützliches Projekt und warum?

- Appl. weil er weiß, dass er sein Projekt nicht so enden soll
- Für alle, weil dadurch den OWASP Top Ten Leben eingehaucht wird und so deutlich wird für jeden wie die Anwendung nicht aussehen sollte

Aufgabe 4 (12 Punkte)

(4.1) (4 Punkte) Erläutern Sie kurz den Zweck von Output Encoding bei Webanwendungen. Welche Arten von Output sollten ein Encoding durchlaufen? Worauf muss man als Entwickler besonders achten?

- Verhindern, dass durch User Input der Output manipuliert oder von
- Alle Arten
  Es jede Daten, die vorher vom Nutzer eingegeben wurden
- Input Encoding kann durch Sicherheitslücken umgangen werden -> daher Output Encoding notwendig

Reflective XSS

(4.2) (4 Punkte) *„Validierung basierend auf einer Blacklist ist gegenüber einer einfachen Whitelist weniger sicher."* – Beziehen Sie Stellung zu dieser Aussage und untermauern oder widerlegen Sie sie mit Beispielen.

- Die Aussage stimmt meiner Meinung, da
-> eine Whitelist nur bestimmt erlaubt, die auch explizit so definiert sind und alles fragwürdige automatisch ablehnt.
-> Blacklist hat das Problem, dass alles erlaubt bis auf das in der Liste und somit Schadsoftware nicht in der Liste vorhanden sein könnten.

SSH mit gültigen IPs / HTML-Tags

/4

(4.3) (4 Punkte) Grenzen Sie Reflected XSS und Stored XSS gegeneinander ab. Geben Sie Beispiele für Angriffsszenarien mit beiden Varianten.

- Kommentarbereich -> Stored XSS
- jmd. schickt ne bösartige URL } Cookie-Session stehlen

/4

Aufgabe 5 (23 Punkte)

Der WYSIWYG-Texteditor in einer web-basierten Forensoftware unterstützt folgende Formatierungen für Postings und Kommentare und speichert diese im HTML-Format in der Datenbank ab:

Formatierung	**Gespeichertes HTML**
Fett	`<b>Text</b>`
Kursiv	`<i>Text</i>`
Hyperlinks	`<a src="URL">Text</a>`
Zitate	`<blockquote>Text</blockquote>`
Zeilenumbrüche	` `

Auf Knopfdruck ist das Wechseln in eine HTML-Editor Ansicht möglich. Die gespeicherten Posting und Kommentare sind nicht öffentlich einsehbar, sondern erfordern zunächst eine Registrierung und Anmeldung am Forum.

Es gibt einige User, die über die HTML-Editor Ansicht auch Bilder (`<img src="Bild-URL">`) oder Inline-Zitate (`<cite>Text</cite>`) einfügen, was vom Foren-Betreiber toleriert wird. Es sei zudem technisch nicht möglich, im WYSIWYG-Modus das Einfügen von Bildern zu ermöglichen, da es sich um eine veraltete Forensoftware handelt. Weitere HTML-Tags oder -Attribute sind jedoch nicht erwünscht. Ein Austausch des gesamten Editors oder der Forensoftware ist aus finanziellen Gründen unrealistisch.

(5.1) (8 Punkte) Schlagen Sie dem Betreiber des Forums 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.

- Input Validation (Whitelist für Input tags)
- Input Sanitizer
- Output Encoding
- Event-Handler verbieten -> Sicherheit

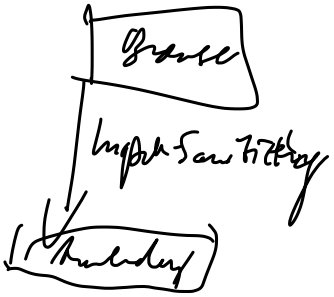
} Sicherheit u. Benutzerfreundlichkeit weil die alten Tags bleiben und genutzt werden können

- Power-User rauswerfen -> schlechter Kompromiss
  └ die Sachen, die sie verwenden wollt mehr erlauben

15

(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.

- er prüft den Input und räumt dann ggf. auf bzw. lässt unerwünschte Tags nicht zu. Meistens arbeitet dieses mit Whitelisten, wenn am besten gegen Angriffe geschützt werden. Wenn es durch eine Whiteliste beschränkt ist und somit Eventuelles nicht erlaubt, kann es eine Art sehr sicher sein gegen XSS geschützt zu sein



- Aufpassen, dass er nicht umgangen wird
  Serverseitig

/3

(5.3) (5 Punkte) Der Entwickler der Forensoftware 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	`<b>Kommentar</b> <i>von Localhorst</i> <blockquote>TextTextText</blockquote>  <a href=”http://for.um”>Link</a>`	`<b>Kommentar</b> <i>von Localhorst</i> <blockquote>TextTextText</blockquote>  <a href=”http://for.um”>Link</a>`	X	
2	`<b>Titel</b> <i>von Localhorst</i> <a href=”http://for.um”> <img src=”http://for.um/p.png”> </a>`	`<b>Titel</b> <i>von Localhorst</i> <a href=”http://for.um”> <img src=”http://for.um/p.png”> </a>`	X	
3	`Titel <h3>von Localhorst</h3><hr> <cite>TextTextText</cite> <script>alert(“Test”)</script>`	`Titel  <cite>TextTextText</cite>`	X	
4	`<b>Titel</b> <i>von Localhorst</i> <a href=”http://for.um”> <img src=”http://for.um/p_xs.png” onload=“alert(‘welcome!’)“> </a>`	`<b>Titel</b> <i>von Localhorst</i> <a href=”http://for.um”> <img src=”http://for.um/p_xs.png” ~~onload=“alert(‘welcome!’)“~~> </a>`		X
5	`<pre<code>>Titel</p</code>re> <u>von Localhorst</u> <small>TextTextText</small>`	~~<pre>Titel</pre~~ 		X

VG

(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.

#4 Durch den Eventhandler könnte es möglich sein, dass JS Code angegeben wird.

#5 <script<code>>
So würde laut dem Sanitiser nur der innere Tag entfernt werden wodurch dann script Tag als Output übernommen wird

4

(5.5) (3 Punkte) Empfehlen Sie dem Betreiber <u>drei</u> konkrete Maßnahmen zur Verbesserung der Sicherheit in seines Forums.

- Input Sanitizer
- Unerlaubte User nicht mehr zu tolerieren
- Output Encoding

- SAST/DAST

3

Aufgabe 6 (15 Punkte)

```
import java.io.*;

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

          InputStream in = proc.getInputStream();
          InputStreamReader stream = new InputStreamReader(in);
          BufferedReader out = new BufferedReader(stream);

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

(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.

– Konsolen Aufruf mit einem Argument, was Dateipfad ist

- Übergabe eines Pfades und Ausgabe des Inhaltes auf der Konsole zeilenweise

3

(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!)*

- Sicherheitslücke ist, dass das Argument ohne Prüfung akzeptiert wird und dann ausgeführt

args[0] = * && * .exe

5

(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!)*

Aufgabe 7 (10 Punkte)

	Beantworten Sie die folgenden Fragen:	**stimme zu**	**stimme nicht zu**
a)	Shodan ist eine anonyme Suchmaschine als Konkurrenz zu Google oder Bing.		X
b)	Mobile-Apps aus inoffiziellen Quellen zu installieren ist potentiell gefährlich.	X	
c)	Das Default-Passwort des eigenen WLAN-Routers braucht man nicht zu ändern.		X
d)	Versionsinformationen in client-seitigen Fehlermeldungen sind ungefährlich.		X
e)	Hacktivists nennt man die kommerziellen Autoren von Profi-Hackingtools.		X
f)	Korrupte Mitarbeiter gehören zu den potentiell ungefährlicheren Angreifern.		X
g)	Bei DOM-based XSS wird der Schadcode vom Server zum Client zurück geschickt.		X
h)	Bei Stored XSS wird der Schadcode vom Angreifer z.B. in der DB abgelegt.	X	
i)	Eine XSS-Attacke funktioniert nur in Kombination mit einer XXE-Verwundbarkeit.		X
j)	Deserialisierungs-Mängel sind leicht auszunutzen aber eher wenig gefährlich.		X

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

10

Aufgabe 8 (8 Punkte)

Erläutern Sie den Zweck und die Aufgaben einer Web Application Firewall. Motivieren Sie deren Einsatz zur Umsetzung des Security-Prinzips „Defense in Depth“ an einem praktischen Beispiel.

WAF = Firewall für eine Web Application

Aufgabe 9 (10 Punkte)

Erklären Sie stichpunktartig die folgenden Prinzipien sicheren Designs:	
Avoid Security by Obscurity	
Don't trust Services	
Establish Secure Defaults	
Principle of Least Privilege	
Separation of Duties	

Viel Erfolg!