Klausur Websecurity WS23/24

2. Klausurtermin am 09.04.2024

Name:

Matrikelnummer:

Hinweise

- Bitte schreiben Sie Ihren **Namen und Matrikelnummer** auf das Deckblatt.
- **KEINESFALLS** die Klammerung auflösen.
- Nur Dokumentenechte Stifte benutzen.
  Keinen roten oder grünen Stift nutzen.
  Keinen Bleistift, Tintenkiller oder TippEx benutzen.
- Es sind keinerlei Hilfsmittel erlaubt.
- Schreiben Sie leserlich. Unlesbare Antworten werden nicht gewertet.
- Falls Sie mehr Platz benötigen, fragen Sie nach Papier.
- Die Bearbeitungszeit beträgt **90 Minuten**.

Dieser Teil ist zur Auswertung bestimmt und
darf von den Teilnehmenden der Klausur **nicht** ausgefüllt werden!

Aufgabe	**A1** 5P	**A2** 6P	**A3** 5P	**A4** 6P	**A5** 8P	**A6** 9P	**A7** 6P	**A8** 5P	**Gesamt** 50P	**Note**
erreichte Punkte										

Aufgabe 1: Multiple Choice **(5 Punkte)**

Beurteilen Sie die folgenden Aussagen, indem Sie die richtigen ankreuzen und die falschen leer lassen! Ein Punkt für jede Teilaufgabe wird nur dann vergeben, wenn Sie in dieser alle richtigen und keine falschen Kreuze gesetzt haben.
Hinweis: Es können alle, keine oder mehrere Aussagen richtig sein.

(a) (1 Punkt) HTTP Status codes

Der HTTP Status-code 200 zeigt an, dass der Webserver die HTTP-Anfrage korrekt verarbeitet hat.	◯
Es existiert ein negativer HTTP Status-code der anzeigt, dass ein Webserver die HTTP Anfrage abgelehnt hat.	◯

(b) (1 Punkt) Cookies

Cookies können die Flags `insecure, samesite` und `httpsOnly` gesetzt haben.	◯
Der HTTP Standard sieht vor, dass immer nur ein Cookie an eine URL geschickt werden kann.	◯

(c) (1 Punkt) SOP

Die Same-Origin-Policy prüft den Zugriff einer Seite auf die Cookies dieser Seite.	◯
Die Same-Origin-Policy kann mittels CORS “aufgelockert” werden.	◯

(d) (1 Punkt) URLs & DNS

`https://tu.berlin:80/sect/` ist nach URL-Schema gültig.	◯
Das Domain Name System (DNS) kann die URL `http://tu.berlin/sect` zu `http://141.23.35.56/sect` ändern.	◯

(e) (1 Punkt) Passwörter

Schlecht gewählte Password-Policies können zu schwachen Passwörtern führen.	◯
Ein 6-stelliges Passwort aus Ziffern ist schneller brute-forcebar als ein 5-stelliges aus Gross- und Kleinbuchstaben.	◯

Punkte	

Aufgabe 2: SOP, Cookies, CORS **(6 Punkte)**

(a) (1 Punkt) **Kreuzen** Sie das korrekte Triple an, anhand dessen die Same-Origin-Policy den Zugriff von Seite A auf Seite B überprüft.

- [ ] (Protokoll, Domain, Pfad)
- [ ] (Domain, Port, Pfad)
- [ ] (Protokoll, Domain, Port)
- [ ] ([HTTPS], Domain, Pfad)
- [ ] (Domain, Pfad, Parameter)

(b) (1 Punkt) **Erklären** Sie in wenigen Sätzen, weshalb ein CORS-Header mit dem Wert $\star$ (d.h. `access-control-allow-origin: *`) gefährlich werden kann.

(c) (2 Punkte) Ein Browser hat die folgenden 4 Cookies gespeichert:

(a) `session=0xc0ffee; domain=tu-berlin.de; HttpOnly; Secure`

(b) `session=0x1337; domain=isis.tu-berlin.de; Secure`

(c) `session=0xdead; domain=moses.tu-berlin.de; HttpOnly`

(d) `session=0xbeef; domain=tu.berlin; path=/sect`

Geben Sie zu jeder URL an, welche Cookies ein Browser mitschicken würde. Es genügt, wenn Sie den Buchstaben des jeweiligen Cookies neben die URL schreiben (bspw. a, b, c).

- `https://tu-berlin.de/typo3/`
- `http://isis.tu-berlin.de/`
- `https://moses.tu-berlin.de/`
- `https://tu.berlin/`

Punkte	

(d) (2 Punkte) Alice implementiert in ihrer Freizeit einen eigenen Browser. Die Same-Origin-Policy (SOP) wurde allerdings bisher noch nicht eingebaut. Dennoch nutzt Sie den Browser bereits alltäglich, um im Internet zu surfen.
Beschreiben Sie zunächst in 1-2 Sätzen die Funktionweise der SOP und **begründen** Sie anhand eines Beispiels, ob Alice sich einem Sicherheitsrisiko aussetzt.

Punkte	

Aufgabe 3: Authentication **(5 Punkte)**

(a) (1 Punkt) **Begründen** Sie, weshalb die Validierung von Zugangsdaten auf Webseiten nicht client-seitig, sondern server-seitig durchgeführt werden sollte.

(b) (1 Punkt) Alice hat die folgenden Hash-Algorithmen zur Auswahl um Passwörter in einer Webapplikation zu hashen: Argon2, Bcrypt, MD5, SHA-1 und SHA-256.
Begründen Sie für welchen Hash-Algorithmus sich Alice bei der Implementierung entscheiden sollte.

(c) (2 Punkte) Beim Passwort-Hashing tauchen häufig die Begriffe “salt” und “pepper” auf.
Beschreiben Sie zunächst die zwei Begriffe in eigenen Worten und **erklären** Sie in 1-2 Sätzen, welchen Vorteil die Nutzung dieser hat.

(d) (1 Punkt) **Nennen** Sie zwei weitere Klassen an Schutzmechanismen die Webapplikationen implementieren sollten, um Accounts vor unberechtigtem Zugriff zu schützen.

-
-

Punkte	

Aufgabe 4: Misconfiguration **(6 Punkte)**

(a) (1 Punkt) Sie navigieren auf einer Webseite in den Unterordner `/folder/`. Der Webserver zeigt plötzlich eine Seite mit dem Titel `Index of /folder/` und allen Inhalten des Ordners an. **Beschreiben** Sie das Risiko von Directory Listing.

(b) (4 Punkte) In dem intercepting Proxy sehen Sie beim Abruf einer Webseite die folgenden HTTP-Header in der HTTP-Antwort.

```
1 HTTP/2 200
2 content-type: text/html; charset=utf-8
3 date: Fri, 05 Apr 2024 15:33:53 GMT
4 referrer-policy: strict-origin
5 server: nginx/1.12.2
6 x-powered-by: php/5.1.2
7 strict-transport-security: max-age=31536000
8 x-content-type-options: nosniff
9 x-xss-protection: 1; mode=block
```

Geben Sie an, welche **zwei** Header Sie aus Sicherheitsgründen entfernen lassen würden und **begründen** Sie diese Entscheidung in 1-2 Sätzen.

-
-

Geben Sie nun **zwei** HTTP-Header mit sinnvollem Wert an, welche in der HTTP-Antwort fehlen und die Sicherheit der Webseite erhöhen würden.

-
-

(c) (1 Punkt) Heutzutage wird fast der gesamte HTTP-Verkehr durch SSL/TLS verschlüsselt übertragen. **Erläutern** Sie in wenigen Sätzen, weshalb damit die Sicherheit der Nutzer im Internet erhöht wird.

Punkte	

Aufgabe 5: HA-Share-Webseite **(8 Punkte)**

Alice hat eine Webseite ('https://ha-share.de/') zum Teilen von Informationen zu Hausaufgaben entwickelt. Bevor diese genutzt kann, muss ein Account registriert werden.

(a) (2 Punkte) Nach einem erfolgreichen Login wird der Nutzer auf die zuvor besuchte Unterseite von HA-Share weitergeleitet, wie im folgenden Quelltext implementiert:

```
1 <?php
2  // [...]
3  if(isset($_GET['r'])) {
4   header("Location: $r");
5  }
6  // [...]
```

Nennen Sie die Art der Sicherheitslücke, die hier erkennbar ist und **geben** Sie einen Exploit an, mit dem ein Angreifer diese Lücke ausnutzen könnte.

(b) (1 Punkt) Bob empfiehlt Alice folgende Änderung am Quelltext[1], um die Sicherheitslücke zu beheben:

```
1 if(isset($_GET['r']) &&
2  preg_match('/https?:\/\/ha-share.de\//', $_GET['r'])) {
3   header("Location: $r");
4 }
```

Beurteilen Sie, ob diese Änderung ausreichend ist. Falls nein, **geben** Sie einen Exploit an, der diese Validierung umgeht.

[1] `preg_match($regex, $subject)` prüft ob das 2. Argument zu dem regulären Ausdruck im 1. Argument passt.

Punkte	

(c) (1 Punkt) **Beschreiben** Sie in wenigen Sätzen eine alternative Behebungmöglichkeit, die das Sicherheitsproblem beseitigt.

(d) (1 Punkt) Eingeloggte Nutzer können Hausaufgabenhinweise auch wieder löschen. Auf der Übersichtsseite werden die Löschlinks wie folgt erzeugt:

```
1 //[...] get_user_ha_ids liefert die numerischen IDs der HAs.
2 for($id in get_user_ha_ids()) {
3  echo "<a href='/ha.php?action=delete&id=$id'>Loeschen</a>";
4 }
5 //[...]
```

Die Löschung wird in `ha.php`[2] wie folgt implementiert:

```
1 if($_GET['action'] == "delete") {
2  $id = intval($_GET['id']);
3  delete_ha_from_user($id, $_SESSION['uid']);
4 }
```

Nennen Sie die Art der Sicherheitslücke die Sie hier erkennen können und **begründen** Sie Ihre Wahl.

(e) (1 Punkt) **Beschreiben** Sie, wie ein Angreifer diese Sicherheitslücke ausnutzen kann und welche Auswirkung(en) das für die HA-Share Seite hat.

[2] `intval($str)` wandelt den String in eine Ganzzahl um.

Punkte	

(f) (2 Punkte) Um die Sicherheitslücke zu beheben, wird anstatt des Löschlinks ein Löschbutton implementiert und in der `ha.php` wird `$_GET` durch `$_POST` ersetzt.

```
1 for($id in get_user_ha_ids()) {
2  echo "<form action='/ha.php?action=delete&id=$id' method='post'>
3   <button type='submit'>Loeschen</button></form>";
4 }
```

Beurteilen Sie, ob damit die Sicherheitslücke behoben ist. Falls nein, **beschreiben** Sie eine Ihnen bekannte Methode zur Behebung!

Punkte	

Aufgabe 6: HA-Forum **(9 Punkte)**

Die HA-Share-Seite wurde zu einem Forum umgebaut, in dem Nutzer Beiträge verfassen können.

(a) (2 Punkte) Beim Verfassen der Beiträge können sogenannte "tags" genutzt werden, welche beim Anzeigen des Beitrags in HTML Elemente umgewandelt werden. Bspw. wird aus dem Tag `[a]http://website.de/[/a]` der HTML-Link `<a href="http://website.de/">http://webseite.de</a>`.
Die folgenden Zeilen implementieren diese Ersetzung[3]:

```
1 $content = $_POST['content'];
2 $search = '#\[a\](.+)\[\/a\]#iUs';
3 $replace = "<a href=\"$1\">$1</a>";
4 $result = preg_replace($search, $replace, $content, 1);
5 echo $result;
```

Geben Sie an, um welche Art von Cross-Site Scripting es sich handelt.

Geben Sie einen Exploit an, der `alert(1)` ausführt.

(b) (2 Punkte) Die Entwickler behaupten, die Sicherheitslücke mit folgendem Code[4] behoben zu haben:

```
1 $search = '#\[a\](.+)\[\/a\]#iUs';
2 $result = preg_replace_callback($search, function($matches) {
3  $safe = htmlentities($matches[1]); // $matches[1] entspricht $1;
4  return "<a href=\"$safe\">$safe</a>";
5 }, $content, 1);
6 echo $result;
```

Begründen Sie, ob Sie der Aussage zustimmen. Falls nicht, **geben** Sie einen Exploit an, der `alert(1)` ausführt.

[3] Die genaue Funktionsweise von `preg_replace` ist nicht relevant. Die Funktion sucht und ersetzt mit einem regulären Ausdruck innerhalb eines Strings.

[4] `preg_replace_callback` ruft eine Funktion mit den gefundenen Zeichenketten auf, anstatt diese direkt zu ersetzen.

Punkte	

Alice meldet sich im HA-Forum an. Ihr ist `nutzer123` und ihr Passwort ist komplex und enthält Sonderzeichen. Der Login in das HA-Forum scheitert allerdings mit der Fehlermeldung `You have an error in your SQL syntax ...` Bei Bobs Account mit einem schwachen Passwort ohne Sonderzeichen funktioniert der Login fehlerfrei. Sie vermutet daher, dass der Code wie folgt aussieht:

```
1 $res = $db->query("SELECT uid FROM users WHERE uname = '$username'
2                          AND pw = '$pw' AND admin = 0;");
3 if($res) {
4  echo "Willkommen. Du wirst weitergeleitet...":
5  header("Location: /has");
6 } else {
7  echo "Login fehlgeschlagen";
8  echo $db->error();
9 }
```

(c) (1 Punkt) **Nennen** Sie zwei mögliche SQL-Injektion Angriffstechniken, die hier angewendet werden könnten:

-
-

(d) (1 Punkt) **Erklären** Sie, was Sie über die Speicherung der Passwörter ableiten können.

(e) (2 Punkte) **Beurteilen** Sie, ob die Schwachstelle genutzt werden kann, um sich als Administrator ("`haadmin`") ohne Kenntnis des Passworts einzuloggen. Falls ja, **geben** Sie einen entsprechenden Exploit an.

(f) (1 Punkt) **Beschreiben** Sie, was Sie den Entwicklern des HA-Forums empfehlen würden, um diese Art der Schwachstellen zu vermeiden?

Punkte	

Aufgabe 7: VL-Folien Downloader **(6 Punkte)**

Bob hat eine Webapplikation gebaut, um Folien oder andere Dokumente aus dem Internet herunterzuladen und auf seinem Server abzuspeichern. Ein Download ist auf 1MB begrenzt, weil der Server nur 100GB Speicherplatz hat.

(a) (2 Punkte) Der Quelltext des Downloaders enthält folgende Zeilen. Bob behauptet, dass keine RCE-Sicherheitslücke existiert, weil a) die URL per https:// gesichert ist und b) per IP-Auflösung geprüft wird, ob die URL valide ist.

```
1 $url = $_GET['url'];
2 $fname = "/home/files/" . uniqid(); // Zufaelliger Dateiname.
3 if(str_starts_with($url, "https://")
4  && gethostbyname(parse_url($url, PHP_URL_HOST))) {
5   system("curl -v --max-filesize 1000000 '$url' -o $fname");
6 }
```

Begründen Sie, ob Sie der Aussage zustimmen oder widersprechen. Falls Sie widersprechen, **geben** Sie einen Exploit an, welcher die RCE-Sicherheitslücke demonstriert.

(b) (2 Punkte) Alice schlägt Bob vor, Zeile 5 durch die folgende Zeile auszutauschen:

```
1 file_put_contents(file_get_contents($url), $fname);
```

Beurteilen Sie, ob es nach dieser Änderung keine Schwachstelle mehr im o.g. Quelltext gibt. Falls die Änderung zu einer anderen Schwachstelle führt, **benennen** Sie diese und **beschreiben** Sie den Nutzen für einen Angreifer.

(c) (2 Punkte) Unabhängig von den möglichen vorherigen Schwachstellen, kann es zu einem weiteren Problem kommen, wenn die Anzahl der Downloads nicht kontrolliert wird. **Benennen** Sie die Schwachstellenart und **beschreiben** Sie mögliche Auswirkungen.

Punkte	

Aufgabe 8: Race Conditions **(5 Punkte)**

Alice und Bob betreiben einen Onlineshop und haben festgestellt, dass einige Kunden einen Gutschein mehrfach nutzen konnten. Der Pseudo-Code der Gutschein-Einlösens auf der Warenkorb-Seite des Onlineshop sieht wie folgt aus:

```
1  gc = eingegebener_gutscheincode;
2  g = hole_gutschein_aus_db(gc);
3  if(g.existiert && g.gueltig) {
4   wc = lade_warenkorb_des_kunden();
5   if(gc.wert >= wc.summe) {
6    gc.wert -= wc.summe;
7    wc.summe = 0;
8   } else {
9    wc.summe -= gc.wert;
10   gc.wert = 0;
11  }
12  aktualisiere_warenkorb_des_kunden(wc);
13  aktualisiere_gutschein_in_db(g);
14 }
```

(a) (2 Punkte) **Geben** Sie an, welche Zeilen das Race-Window umfasst. Beispielsweise bedeutet 7-9, dass das Race-Window vor Zeile 7 anfängt und nach Zeile 9 aufhört.

(b) (2 Punkte) **Erklären** Sie in eigenen Worten, weshalb das Ausnutzen von bzw. Testen auf Race-Conditions in der Praxis oft erschwert ist. **Beschreiben** Sie zudem eine Angriffsmethode, die die Erfolgschancen bei Webapplikationen erhöht.

(c) (1 Punkt) **Beschreiben** Sie, wie Race-Conditions verhindert werden können.

Punkte	