UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR IT-SICHERHEIT

Übungsblatt 13

Security Evaluation und Penetration Testing

Abgabe bis 25. Juli 2023 um 12:00 Uhr im Moodle

Cybersecurity im Sommersemester 2023

Thomas Eisenbarth, Jan Wichelmann, Anja Köhl

Einführung

Ihre Beratertätigkeit bei ITS neigt sich dem Ende zu: Die Firmenwebsite scheint endlich einigermaßen sicher zu sein, und Malware hat sich seit einiger Zeit auch nicht mehr blicken lassen. Viele der durch das neue Abrechnungssystem verschreckten Kunden sind inzwischen zurückgekehrt, das Unternehmen schreibt wieder gute schwarze Zahlen. Das Management ist vollends mit sich zufrieden – fast. Es steht nämlich noch eine Frage im Raum: Was passiert in Zukunft?

Die meisten Ihrer Aufgaben können nun von gut geschulten Mitarbeitern übernommen werden, aber es ist durchaus möglich, dass die Firma einmal mit bisher unbekannten Sicherheitslücken konfrontiert sein wird. Das Management bittet Sie um einige letzte Tipps für allgemeine Ansätze, um auch in Zukunft die Stellung von *Immature Technical Solutions* als „Marktführer Nr. 1 im Bereich sicheres Hosting“ (Zitat aus der neuesten Broschüre) zu verteidigen.

Aufgabe 1 Bug Bounties vs. Common Criteria Audits (7 Punkte)

Ein erster Vorschlag von Ihnen ist es, ein Bug Bounty-Programm aufzubauen, damit zukünftige Sicherheitslücken im Idealfall gemeldet und nicht ausgenutzt werden. Das Management ist da eher skeptisch – warum sollte man jemanden dafür bezahlen, das Firmensystem anzugreifen? Will man nicht gerade das verhindern? Davon abgesehen sei auch überhaupt nicht klar, wie sowas ablaufen soll.

1. Zur Erläuterung zeigen Sie, wie andere große Softwarefirmen dies handhaben: Beschreiben Sie kurz die Vorgehensweise, um
   i. einen Fehler in einem Intel-Produkt zu melden;
   ii. einen Fehler bei GitHub zu melden.
2. Ein Manager hat auf einer Schulung mal von offiziellen *Common Criteria Audits* gehört. Er möchte gern wissen, ob Sie eher zu einem offiziellen Common Criteria Audit oder zu einem Bug Bounty-Programm raten würden. Geben Sie zu beiden Möglichkeiten Ihre Einschätzung bezüglich Kosten, Produktivität, erreichter Sicherheit und Außenwirkung an.

Aufgabe 2 Sicherheitsbewertung von Leitsätzen (5 Punkte)

Technikerin Tanja hat über die letzten Monate einige Aussagen von Kunden und Mitarbeitern gesammelt, die unter anderem bei dem Kontakt im Rahmen einer Datenbank-Sicherheitsüberprüfung getätigt wurden:

- CEO des Reiseunternehmens *MyHolidaysDB*: „Wir werden unseren Code sicher nicht als Open-Source veröffentlichen, das macht es Angreifern doch viel leichter, Sicherheitslücken zu finden!“
- Pressesprecher des Betreibers der Bußgeld-Datenbank *FlensBase*: „Trotz unseres OpSec-Prozesses führen wir regelmäßig Penetration Testings nach Level 2 durch. Wir glauben, dass es uns hilft, wichtige Schwachstellen schon vor den offiziellen Überprüfungen durch die Behörde zu finden.“

- Hauptmann der Feuerwehr eines unbekannten Dorfes zur Feuerwehr-Mitgliederdatenbank *FeuermelderDB*: „Vor kurzem hatten wir ein Problem mit unserem Rechner, und einige Mitgliederdaten konnten nicht wieder hergestellt werden. Das hat sehr viel Stress gegeben. Wir denken jetzt darüber nach, uns einer Common Criteria Evaluation zu unterziehen.“
- Leiterin der Entwicklungsabteilung von ITS: „Wir haben ja schon immer etwas Testing in unseren Entwicklungsprozessen gehabt, aber unser Prinzip der Continuous Integration hat die Sicherheit unserer Software enorm verbessert.“
- Entwickler bei *SmartHeat*, einem Hersteller von ferngesteuerten Thermostaten: „Ja, unsere Produkte sind nach Common Criteria verifiziert. Ich fühle mich nicht immer wohl dabei, sicherheitsrelevante Updates zurück zu halten, aber man muss eben abwägen.“

Nehmen Sie zu den jeweiligen Zitaten Stellung.

Aufgabe 3 Wiederholung: Datenbanksicherheit (7 Bonuspunkte)

Diese Aufgabe muss nicht bearbeitet werden um den Zettel zu bestehen, liefert jedoch Bonuspunkte, die auf einen anderen nicht bestandenen Zettel angerechnet werden können.

Bei der Analyse des Logins einer Online-Forum-Software fällt Ihnen folgendes Codefragment ins Auge:

```
$user = mysql_query(
                "SELECT * FROM users WHERE username='$name' AND password='$pw'"
);
if($user == null)
{
    error("Invalid username or password");
}
else
{
    // Login...
}
```

Hier wird anhand des eingegebenen Nutzernamens `$name` und des Passworts `$pw` ein Nutzerdatensatz gesucht. Falls dieser existiert, wird dieser von der Funktion `mysql_query` zurückgegeben und der Login durchgeführt. Falls dieser nicht existiert, wird `null` zurückgegeben und der Login verweigert.

Allerdings fällt Ihnen auf, dass keinerlei Überprüfung der eingegebenen Zeichenfolgen stattfindet, d.h. die Benutzereingaben werden direkt in das Query eingesetzt.

1. Skizzieren Sie eine Eingabe für Benutzername und Passwort, mit der Sie sich als beliebiger Benutzer (z.B. `admin`) einloggen können, ohne dessen Passwort zu kennen, und geben Sie das resultierende SQL-Query an. Erklären Sie Ihren Ansatz.
2. Erklären Sie, wie sich Angriffe wie der von Ihnen oben skizzierte zuverlässig verhindern lassen.
3. Sie treffen auf eine modifizierte Version des obigen Codes, der den Passwortcheck aus der Datenbankabfrage in ein separates if-Statement verlagert:

   ```
   $user = mysql_query("SELECT * FROM users WHERE username='$name'");
   if($user == null)
   {
       error("Unknown user");
   }
   else
   {
       // Check whether password from user does match submitted password
       if($user["password"] != $pw)
       {
           error("Invalid password");
       }
       else
       {
   ```

```
            // Login...
        }
}
```

Ist hier immer noch ein Angriff möglich? Begründen Sie Ihre Antwort.