Ludwig-Maximilians-Universität München
Institut für Informatik
Lehr- und Forschungseinheit für Kommunikationssysteme und Systemprogrammierung
Prof. Dr. Helmut Reiser, PD Dr. Wolfgang Hommel

MNM TEAM
Munich Network Management Team

IT-Sicherheit im Wintersemester 2015/2016
Übungsblatt 5

Abgabetermin: 24.11.2015 bis 12:00 Uhr

Achtung: Zur Bearbeitung einiger Übungsaufgaben ist es notwendig sich über den Vorlesungsinhalt hinaus, durch Internet- und Literaturrecherche mit dem Thema zu beschäftigen.

Die schriftlichen Lösungen aller mit **H** gekennzeichneten Aufgaben sind **vor Beginn** der jeweils nächsten Übungsveranstaltung abzugeben (über Uniworx als **Einzelabgabe**). Während des Semesters werden vier Übungsblätter ausgewählt, korrigiert und bewertet. Bei vier als korrekt bewerteten Lösungen (mind. 75% der erreichbaren Punkte) erfolgt ein Bonus von zwei Drittel Notenstufen auf die Klausurnote, bei nur drei oder zwei richtigen Lösungen erhalten Sie einen Notenbonus von einer Drittel Notenstufe.

Bei Fragen zu Übungsaufgaben sowie generellen Anregungen, Verbesserungsvorschlägen, ... zum Übungsbetrieb wenden Sie sich bitte per Email an **`uebung-itsec@lrz.de`**.

Aufgabe 13: **(H)** Rootkits (6 Punkte)

Nachdem ein Angreifer erfolgreich Zugang zu einem IT-System, etwa durch das Ausnutzen einer dort vorhandenen Schwachstelle, erlangen konnte, wird dort meist ein Rootkit installiert.

a. Man unterscheidet grundsätzlich zwei Varianten von Rootkits: User-Mode- und Kernel-Mode-Rootkits. Erläutern Sie diese kurz.

b. Wie unterscheidet sich ein Rootkit von anderer Malware, z.B. Viren, Würmer und Trojanischen Pferden?

c. Rootkits verfügen im Allgemeinen über eine sogenannte *Dropper*-Komponente. Welchem Zweck dient diese Komponente. Was versteht man unter einem *Multistage Dropper*?

d. Charakteristisch für Rootkits sind sogenannte Anti-Forensik-Maßnahmen. Erläutern Sie folgende Maßnahmen

   - Data Destruction
   - Data Concealment
   - Data Fabrication

e. Wie beurteilen Sie die Erkennung von Rootkit-Software durch eine signatur-basierte Anti-Virus-Software. Würde der Einsatz einer heuristischen (Anomalie-basierten) Erkennung helfen?

Aufgabe 14: (H) XSS & SQL-Injection (8 Punkte)

a. In der Vorlesung wurden drei verschiedene Arten von Cross-Site-Scripting (XSS) vorgestellt. Beschreiben Sie die drei Arten und ordnen Sie sie entsprechend ihres Bedrohungspotentials.

b. Um XSS zu vermeiden, wird empfohlen jegliche Eingaben von Benutzern zu filtern. Eine weitere Maßnahme zur Vermeidung von XSS ist die Content Security Policy (CSP). Beschreiben Sie die Funktion sowie Vor- und Nachteile von CSP. Wo liegen die Schwierigkeiten CSP für eine bestehende Webseite (wie z.B. http://www.heise.de) zu aktivieren.

c. Beschreiben Sie wie SQL-Injection funktioniert und wie Sie eine Anfrage formulieren würden, um die Loginmaske einer Webseite zu umgehen und auf den Account **administrator** zuzugreifen, von der Sie wissen, dass die Passwörter mit folgendem SQL-Query überprüft werden:

```
'SELECT uid FROM users WHERE username = "' + $username + '" AND
password = "' + $password + '"'
```

Sie können dabei die Parameter `$username` und `$password` über das Formular frei eingeben.

d. Beschreiben Sie sowohl abstrakt als auch konkret, wie Sie den Query aus der vorhergehenden Teilaufgabe verändern müssen, um SQL-Injection zu verhindern. Sie können sich bei der konkreten Beschreibung eine gängige Programmiersprache für das Web (z.B. PHP oder Perl) aussuchen.

Aufgabe 15: (H) Common Vulnerability Scoring System 3 (CVSSv3) (6 Punkte)

Für diese Aufgabe soll die folgende Schwachstellenbeschreibung verwendet werden, die über die vier Teilaufgaben hinweg modifiziert wird. Änderungen in einer der Teilaufgaben gelten auch in den darauf folgenden Teilaufgaben. (d.h. Änderungen in Teilaufgabe b) gelten auch für Teilaufgaben c) und d)).

In einer weit verbreiteten Webanwendung, die in ihrem Unternehmen als Kundenportal zur Verwaltung von Softwarelizenzen verwendet wird und daher öffentlich im Internet zugänglich sein muss, existiert eine cross-site request forgery (CSRF) Schwachstelle. Durch diese Schwachstelle können Angreifer aus der Ferne Aktionen mit den Rechten des angegriffenen Benutzers ausführen, wenn der Benutzer eine aktive Session hat und dazu gebracht werden kann, einen schädlichen Link zu öffnen.

Hinweis: Geben Sie bei den Aufgaben, bei denen explizit CVSS-Berechnungen gefordert sind, nicht nur deren Ergebnisse an, sondern begründen Sie auch die von Ihnen gewählten Optionen.

a. Beschreiben Sie kurz wie ein Angriff per CSRF üblicherweise funktioniert.

b. Berechnen Sie mithilfe des unter `https://www.first.org/cvss/calculator/3.0` verfügbaren CVSSv3-Calculators für die beschriebene Schwachstelle den CVSSv3 Base-Score. Vergleichen Sie diesen mit dem über `https://nvd.nist.gov/cvss.cfm?calculator&version=2` berechneten CVSSv2 Base-Score.

c. Die beschriebene Schwachstelle wurde am selben Tag auch auf der Security-Mailingliste *Full-Disclosure* publiziert und deren Ausnutzbarkeit anhand eines Proof-of-Concept (POC) bewiesen. Der Hersteller der Webanwendung hat die Schwachstelle nun auch offiziell bestätigt, aber bislang nur einen Workaround veröffentlicht. Wie verändert sich dadurch der CVSSv3 Base- bzw. Temporal-Score?

d. Bereits am nächsten Tag tauchte in einschlägigen Foren ein Exploit für diese Schwachstelle auf. Dieser besitzt keine besonderen Voraussetzungen und ist somit in jeder Situation funktional. Wie verändert sich dadurch der CVSSv3 Base-/Temporal-Score aus Aufgabe c)?