Ludwig-Maximilians-Universität München
Institut für Informatik
Lehr- und Forschungseinheit für Kommunikationssysteme und Systemprogrammierung
Dr. Helmut Reiser

MNM TEAM
Munich Network Management Team

IT-Sicherheit im Wintersemester 2009/2010
Übungsblatt 3

Abgabetermin: 18.11.2009 bis 14:00 Uhr

Achtung: Zur Bearbeitung einiger Übungsaufgaben benötigen Sie eine Rechnerkennung für unsere Praktikums Infrastruktur. Um diese zu erhalten, melden Sie sich bitte über die Vorlesungs Webseite zum Übungsberieb an.

Die schriftlichen Lösungen aller mit **H** gekennzeichneten Aufgaben sind **vor Beginn** der jeweils nächsten Übungsveranstaltung abzugeben (per Email, in der Vorlesung oder vor der Übung). Während des Semesters werden vier Übungsblätter korrigiert. Bei vier richtigen Lösungen 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.

Aufgabe 7: (H) XSS

a. Wie werden XSS-Attacken allgemein klassifiziert?

b. Erstellen Sie eine Webseite, die einen in der URL der Seite übergebenen Parameter namens *?search=...* in den Seiteninhalt übernimmt und als `<p>`*parameter*`</p>` ausgibt. Wie verhält sich das Programm, wenn der Parameter *search* den Wert `<script>alert("XSS Attack!")</script>` enthält?

c. Löst das folgende Konstrukt das Problem? `<p><![CDATA[` *parameter* `]]></p>`
Warum (nicht)?

Aufgabe 8: (H) Steganographie

a. Betten Sie die Nachricht "*Dies ist eine versteckte Botschaft*" in die Bilder `rot.jpg`, `bunt.jpg` und `kariert.jpg`, welche Sie auf der Webseite herunterladen können ein. Verwenden Sie hierzu das Werkzeug *steghide*.

b. Extrahieren Sie die versteckten Nachrichten wieder aus den Bildern.

c. Vergleichen Sie die Histogramme der Bilder mit und ohne versteckter Nachricht. Was fällt auf?

d. Welche Techniken existieren, um Nachrichten in Bildern zu verstecken?

e. Wie robust sind die eingebetteten Nachrichten gegenüber nachträglichen Veränderungen am Bild?

Aufgabe 9: (T) Buffer Overflow - Spawning a shell

In Aufgabe 5 auf Übungsblatt 2 wurde das Thema "Buffer Overflow"behandelt. Eine mögliche Zielsetzung eines Hackers, der einen Buffer Overflow ausnutzt, ist das Starten einer Shell. Wie kann man gezielt den Instruction-Pointer manipulieren, um Shellcode-Anweisungen auszuführen?