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 2

Abgabetermin: 11.11.2009 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 4: (H) DoS & DDoS

In der Vorlesung wurden verschiedende Angriffstechniken vorgestellt, u.a. auch DoS und DDoS-Attacken.

a. Erläutern Sie in Stichpunkten den Ablauf von DoS- und DDoS-Angriffen und zeigen Sie wirksame Gegenmaßnahmen auf.

b. Erläutern Sie konkret die Funktionsweise von Syn-Cookies und zeigen Sie, wie dadurch Syn-Flooding Attacken vermieden werden können!

c. Welche Nachteile haben Syn-Cookies?

Aufgabe 5: (H) Buffer-Overflow

Sehr viele aktuelle Angriffe basieren auf Pufferüberläufen, engl. Buffer Overflows.

a. Was passiert bei einem Buffer-Overflow genau? Wie kann ein Hacker diesen ausnutzen? Erläutern Sie stichpunktartig.

b. Folgendes Programm ist gegeben:

```
#include <stdio.h>
int test() {

// initialisiere ein Array mit nur einem Eintrag
int i[1] = {42};
        ...

// Ausgabe des Array und Ende
printf("test: i[0] = %i\n\n", i[0]);
return i[0];
// -> Rücksprung ins aufrufende Programm
}

int main() {

// initialisiere x, x sollte im Nachhinein den Wert 42 haben
int x = test();

printf("Dies ist ein Testprogramm.\n");
printf("Es demonstriert einen Puffer-Überlauf.\n\n");

// inkrementiere x, neuer Wert von x sollte 43 sein
x++;

// Ausgabe von x und Ende
printf("main: x = %i\n", x);
return 0;
}
```

(i) Ergänzen Sie das Programm an der gekennzeichneten Stelle so, dass die Inkrementierung von x in der `main()`-Routine nicht ausgeführt wird! Erklären Sie das Phänomen.

Aufgabe 6: (K) Windows Password Hashverfahren

a. Windows verwendet unter anderem das LM Hashverfahren um Passwörter zu speichern, welches wie folgt arbeitet:

(i) Das Passwort des Benutzers in Form eines OEM-String wird zu Großbuchstaben umgeformt.

(ii) Dieses Passwort wird entweder auf 14 Bytes mit Nullen gefüllt oder gekürzt.

(iii) Das Passwort mit der festen Länge wird in zwei 7 Byte-Hälften aufgeteilt.

(iv) Aus jeder Hälfte wird durch hinzufügen eines NON-Parity-BITs ein 64-BIT langer DES-Schlüssel erzeugt.

(v) Jeder dieser Schlüssel wird dazu genutzt, den Konstanten ASCII-String ”KGS!@#$%” zu DES-verschlüsseln, woraus zwei 8 Byte Chiffretext-Werte resultieren.

(vi) Diese beiden Chiffretext-Werte werden verbunden, um einen 16 Byte-Wert zu bilden, der den LM hash darstellt.

Bewerten Sie das eingesetzte Hashverfahren bezüglich seiner Sicherheit! Wie können die genannten Defizite entschärft werden?