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 2014/2015
Übungsblatt 4

Abgabetermin: 18.11.2014 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.

Aufgabe 8: **(H)** DoS & DDoS (4 Punkte)

Auf dem vorherigen Übungsblatt 3 haben sie sich mit wirksamen Maßnahmen als Reaktion auf DoS und DDoS-Attacken beschäftigt. Erläutern Sie folgende drei Verfahren im Detail

- Blackholing bzw. Blackhole Route
- Upstream filtering
- Cloud-based Mitigation (z.B. Cloudflare)

Gehen Sie außerdem auf Stärken und insbesondere Schwächen oder Probleme des jeweiligen Verfahrens ein.

Aufgabe 9: (H) Buffer-Overflow (6 Punkte)

Angreifer nutzen oftmals Schwachstellen in lokal installierten Applikationen.

a. Erläutern Sie knapp, was bei einem Buffer-Overflow genau passiert? Wie kann ein Angreifer dies ausnutzen?

b. Folgendes Programm ist gegeben:

```
1  #include <stdio.h>
2
3  char shellcode[] = "\xbb\x14\x00\x00\x00"
4                     "\xb8\x01\x00\x00\x00"
5                     "\xcd\x80";
6
7
8  int main() {
9
10     int *ret;
11
12     ret = (int *)&ret + <integer>;
13
14     (*ret) = (int)shellcode;
15 }
```

(i) Beschreiben Sie den grundsätzlichen Ablauf dieses Programms.

(ii) Der Stack habe bei Ausführung dieses Programms folgenden Aufbau (mithilfe von *gdb* und Breakpoint in Zeile 12 ermittelt):

0xbfa62f84:	0x08048350	0xbfa62fe8	0xb7df0390	0x00000001
0xbfa62f94:	0xbfa63014	0xbfa6301c	0xb7f262d0	0x00000000

Was verbirgt sich hinter den Werten *0x08048350* und *0xb7df0390*?

(iii) Ergänzen Sie den Programmtext an der Stelle *integer* (Zeile 12) mit dem korrekten Wert, damit der Shellcode in Zeile 14 (Adresse: 0x08049504) korrekt aufgerufen wird. Am Ende der Programmausführung sehe der Stack wie folgt aus:

0xbfa62f84:	0xbfa62f8c	0xbfa62fe8		0x00000001
0xbfa62f94:	0xbfa63014	0xbfa6301c	0xb7f262d0	0x00000000

Welcher Wert sollte in der Lücke nun stehen?

c. Betrachten Sie den gegebenen Shellcode. Welche Probleme könnten hierbei auftreten?

d. Als wirksame Gegenmaßnahme gegen Buffer Overflows wurde die Address Space Layout Randomization (ASLR) eingeführt. Welches Problem besteht insbesondere dann, wenn das Programm nicht als *Position Independent Executable (PIE)* kompiliert wurde?

Aufgabe 10: (H) Passwort-basierte Authentifizierung (4 Punkte)

Bei älteren Unix-Systemen werden Nutzerpasswörter per *crypt* verschlüsselt gespeichert.

a. Oftmals findet sich in der Datei /etc/passwd statt eines verschlüsselten Passwort-Strings der Wert *x*. Was bedeutet dieser Wert und welchen Vorteil hat dieser gegenüber der herkömmlichen Methode?

b. Passwörter werden heutzutage meist gehashed gespeichert. Trotzalledem welche Defizite weist der in Microsoft Windows eingesetzte LM-Hash auf.

c. Auf der Webseite zur Vorlesung finden Sie im Abschnitt Übung zu diesem Übungsblatt eine Datei passcodes.nfo. Downloaden und installieren Sie sich das bekannte Passwort-Cracking-Programm *John the Ripper* und versuchen Sie die in der Text-Datei enthaltenen Passcodes zu knacken. Geben Sie die ungefähre Dauer an, die für das Knacken des jeweiligen Passworts erforderlich war. (Hinweis: Der Programmlauf könnte je nach verwendetem IT-System einige Zeit (ca. 1 Stunde) in Anspruch nehmen!)