Grundlagen der Systemsoftware

SoSe 2010 · Übungsblatt 6 · Abgabe: Mi. 30.06.2010 (bis 12:00 Uhr)

Aufgabe 1: Einsatz von Kryptographie in der Praxis (30 Punkte)

Die Universität Hamburg ist Teil einer Public-Key-Infrastruktur, in deren Rahmen die Identität einer Person anhand eines Ausweises zunächst überprüft wird, bevor diese Person dann ein Zertifikat erhält, mit dem beliebige andere Kommunikationspartner überprüfen können, ob und von wem der Ausweis überprüft wurde. Jede Person, die an der Uni arbeitet oder studiert, kann so ein Zertifikat erhalten.

Finden Sie die Informationen über die Zertifizierung an der Uni und richten Sie Ihren Email-Client entsprechend ein. Mit dem ausgedruckten Antrag wenden Sie sich bitte an den dafür vorgesehenen Mitarbeiter des RZ, der die Überprüfung des Ausweises vornimmt. Den Nachweis der erfolgreichen Aufgabenerfüllung erbringen Sie durch das Absenden einer verschlüsselten und signierten Email an ihren Betreuer.

Diese Aufgabe ist von jedem Mitglied einer Kleingruppe zu erfüllen. Die Vergabe der Punkte erfolgt gemäß der Anzahl erhaltener eMails in Relation zur Gruppengröße.

Aufgabe 2: Details über TCP/IP (40 Punkte)

Die TCP/IP-Protokollfamilie bildet heute das elementare Bindungsglied für alle Arten von IT-Systemen. Damit kommt den Details der Protokolle eine große Bedeutung zu. Die folgenden Fragen dienen der Einarbeitung in solche Details, ohne deren Kenntnis bestimmte Sicherheitskonzepte – gerade im Firewall-Umfeld – nur schwer zu vermitteln sind.

a) Bestimmen Sie alle Felder eines IP-Headers (Version 4) und beschreiben Sie kurz in eigenen Worten die Funktion der Felder. Achten Sie besonders auf die Werte „TTL“ (Time to Live) und „Fragment Offset“. (10 Punkte)

b) Bestimmen Sie alle Felder eines TCP-Headers und beschreiben Sie kurz in eigenen Worten die Funktion der Felder. Achten Sie besonders auf die Flags und Ports. (10 Punkte)

c) Beschreiben und erklären Sie den sogenannten TCP-Handshake, bei dem ein Client eine TCP-Verbindung zu einem Server aufbaut. Ignorieren Sie dabei mögliche Fehlerfälle. Fertigen Sie eine Grafik an, die zeigt, welche TCP-Pakete mit welchen Flags von wem zu wem geschickt werden. (10 Punkte)

d) Recherchieren Sie, durch wen und wie TCP-Verbindungen abgebaut werden können. Verdeutlichen Sie den entsprechenden Fluss der TCP-Pakete wieder in Grafiken. (10 Punkte)

Aufgabe 3: Software Security (30 Punkte)

Buffer Overflows zählen zu den häufigsten Schwachstellen, die immer wieder nicht nur die Hersteller zum Nachbessern veranlassen, sondern vor allem für viele kompromittierte Rechner betroffener Benutzer im Internet verantwortlich sind.

Gerade C als anerkannte Programmiersprache im kommerziellen Bereich weist Konstrukte auf, die sehr einfach zu Buffer Overflows führen:

- `strcpy()`
- `strcat()`
- `sprintf()`

a) Schreiben Sie ein kurzes C Programm, mit dem Sie einen Buffer Overflow demonstrieren können. Das Programm soll eine beliebige Eingabe von der Tastatur akzeptieren und in einen Buffer kopieren, bevor dieser dann auf dem Bildschirm ausgegeben wird. Bei einer kurzen Eingabe soll das Programm richtig funktionieren, bei längeren Eingaben, als der Buffer groß ist, wird das Programm in der Regel abbrechen oder abstürzen.

Schicken Sie den Quelltext (idealerweise auch eine kompilierte Version, die auf rzdspc<X> lauffähig ist), an ihren Betreuer. Falls Sie die Programmiersprache C nicht beherrschen, reicht auch ein syntaktisch ähnlicher Dialekt, der die grundlegende Idee des Programms erkennen läßt. Achten Sie bitte darauf ihren Quelltext sinnvoll zu kommentieren.

(20 Punkte)

b) Recherchieren Sie im Internet, welche Programmiertechniken oder Bibliotheken (als oft kommerzielle Lösungen) eingesetzt werden können und Buffer Overflows verhindern bzw. verhindern, dass diese schwerwiegende Auswirkungen haben. Stichworte: Canaries, Stackguard, LibSafe, SafeC (u. a.). Suchen Sie sich eine Lösung heraus und skizzieren Sie kurz (10–15 Zeilen), was die Lösung bietet und nach welchem Funktionsprinzip diese gegen Buffer Overflows wirkt. (10 Punkte)