NetSec 1. Übungsblatt

Carl-Friedrich Lübeck, 2101543
Florian Forster, 2099894
Sebastian Harl, 21095050

7. November 2006

Aufgabe 1

a Lehrstuhl

- **Confidentiality**: Da bei der Kommunikation mit Kunden, externen Mitarbeitern o. Ä. vertrauliche Daten ausgetauscht werden, muss die Kommunikation geschützt werden.
- **Data Integrity**: Es sollte nicht möglich sein Forschungsergebnisse (unbemerkt) zu verfälschen, die Buchhaltung zu manipulieren, etc.
- **Controlled Access**: Lehrstuhlfremde Personen sollen/dürfen keinen Zugriff auf Forschungsergebnisse, Geschäftsbeziehungen und Finanzinformationan haben.

b Arztpraxis

- **Confidentiality**: Patientendaten sollten, etwa beim Austausch mit dem Haus- bzw. Allgemeinarzt, auch nur wirklich den beabsichtigten Personen zugänglich sein.
- **Accountability**: Im Zweifelsfall sollte nachvollziehbar sein, **wer** bestimmte Daten erhoben hat.
- **Controlled Access**: Selbstverständlich sollten Patientendaten auch nicht wildfremden Personen zur Verfügung stehen.

c Zu Hause

- **Confidentiality**: Mein Nachbar soll WLAN-Traffic nicht einfach mitlesen, der Netzbetreiber E-Mails nicht mitsniffen können.
- **Controlled Access**: Ohne gültiges Login soll kein Zugriff auf Daten und Funktionen der Systeme möglich sein.

Zu **c**: So lange kein physikalischer Zugang zu den Systemen existiert, ist die Einhaltung dieser Ziele nicht unwahrscheinlich. Da ich physikalischen Zugang als ausreichend unwahrscheinlich einschätze, sind keine entsprechenden Anstrengungen unternommen worden.

Aufgabe 2

	Ereignis	Beteiligte
a	Erstellung der Software	Hersteller
b	mögliche Attacken auf das System	Cracker? *Fehlt hier nicht ein Verb?*
c	Erkennung/Entdeckung der Attacken	Hersteller, ggf. Anwender
d	Analyse der Attacken	Hersteller, ggf. Anwender
e	Erstellung einer Aktualisierung der Software	Hersteller, ggf. Hacker
f	Verbreitung der Aktualisierung	Hersteller, ggf. Anwender

Teilaufgabe a

Entwickler müssen **mögliche** Attacken voraussehen und möglichst vorbeugen. Wenn trotzdem ein Angriffspunkt gefunden wird, liegt es an ihnen den Angriff zu analysieren und den Fehler zu beheben. Der Entwickler muss sich also deutlich länger um das System kümmern, als die Entwicklungsphase dauert.

Teilaufgabe b

Zwar ist der Entwickler nicht ganz machtlos: Er kann beispielsweise eine Big-Endian-Platform wählen, weil die meisten auf Buffer-Overflows basierenden Exploids fuer Little-Endian geschrieben werden. Das kann ihm die Schar der an sich ungefährlichen (aber nervigen) Script-Kiddies vom Hals halten. Gegen einen „richtigen“ Cracker hilft das natürlich nicht.

Einen klaren Vorteil hat der Angreifer: Er kann sich den Ort des Geschehens und den Zeitpunkt wählen und braucht an sich nur einen Fehler des Entwicklers finden. So bald das Produkt also veröffentlicht wurde, befindet sich der Entwickler ununterbrochen (mehr oder minder) in der Defensive.

Aufgabe 3