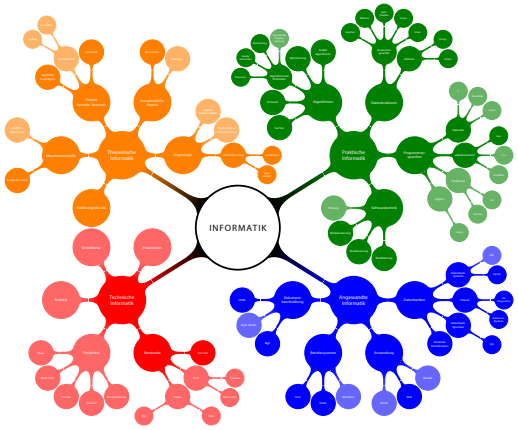
UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR
THEORETISCHE INFORMATIK

Kapitel 44

System-Sicherheit

Von Viren, Würmern und SQL-Spritzen

Vorlesung Einführung in die Informatik 2 vom 15. Juli 2014 von Till Tantau



Lernziele von Kapitel 44

1. Bedrohungsszenarien der Sicherheit von IT-Systemen kennen und einschätzen können
2. Schutzmaßnahmen für die Sicherheit von IT-Systemen kennen und ergreifen können
3. Beispiel eines Sicherheitslochs verstehen

Gliederung von Kapitel 44

Wiederholung: Worum geht es bei IT-Sicherheit?

Bei *IT-Sicherheit* geht es um folgende Anliegen:

1. Schutz vor und die Aufrechterhaltung des Betriebs bei
   - Ausfall von Teilen des Systems (Stromausfall, Absturz)
   - Angriffen auf das System (durch Hacker, korrumpierte Mitarbeiter)

   Diese *Systemsicherheit* wird uns in diesem Kapitel interessieren.
2. Schutz von Daten und Kommunikation vor
   - Spionage
   - Fälschung

   Diese *Daten- und Kommunikationssicherheit* war Thema des letzten Kapitels.

Wie erreicht man IT-Sicherheit?

Es gibt verschiedene Maßnahmen, um die IT-Sicherheit zu erhöhen.
Perfekte Sicherheit kann es nicht geben.
Mögliche Maßnahmen sind:

Redundanz Daten liegen mehrfach vor.
Stichwörter: Backups.

Abschottung Es wird schwierig gemacht, in das System hineinzukommen.
Stichwörter: Passwörter und Firewalls.

Aktive Kontrolle Es wird aktiv im laufenden Betrieb überprüft, ob das Systemverhalten normal ist.
Stichwort: Virenchecker, Vier-Augen-Prinzip, Intrusion-Detection

Verschlüsselung Alle Daten werden verschlüsselt. Ohne die Schlüssel sind die Daten nichts wert.
Stichwörter: ssh (secure shell), pgp (pretty good privacy), gpg (gnu privacy guard), https (http secure)

Rechner müssen geschützt werden.

1. Hardware kann ausfallen, was *Datenverlust* und/oder *Produktivitätsverlust* zur Folge hat.
2. Hardware kann sich *bösartig verhalten*:
   - Moderne Rechner sind Mehr-Prozessor- und Mehr-Benutzer-Systeme.
   - Sie können *selbstständig* mit anderen Rechnern kommunizieren.
   - Dadurch kann ein Rechner *von außen übernommen werden*.
   - Dies bedeutet, dass sich jemand als ein Benutzer ausgibt und dann dem Computer (bösartige) Befehle erteilt.
   - Ist der Angreifer geschickt, so merkt man davon nichts. Solche Rechner heißen dann *Zombies*.

Daten sind wichtiger als Hardware.

Neulich auf einem Schild in einem kleinen Laden in Berlin:

> *Gestern wurde in diesen Laden eingebrochen und mehrere Computer gestohlen. Die Computer sind uns egal, aber wir benötigen die Daten auf den Rechnern! Bitte, ihr Diebe, gebt uns die Daten zurück, Diskretion und eine Belohnung garantiert!*

- Praktisch alle Daten, die in den Systemen einer Firma lagern, sind schützenswert.
- Ein paar wichtige sind:
  - Kundenkontaktdaten,
  - Forschungsergebnisse,
  - Lagerbestandsdatenbanken oder
  - Buchhaltung.
- Diese Daten müssen nicht nur gegen Diebstahl, sondern auch gegen Verlust durch Feuer, etc. geschützt werden.

Private Daten müssen geschützt werden.

- Zu Ihrer Privatsphäre gehört nicht nur Ihre Wohnung, sondern auch Ihre Festplatte.
- Das Bundesverfassungsgericht hat dies kürzlich sogar in einem Grundsatzurteil zu einem Grundrecht erhoben.
- Sie müssen aber Ihre Grundrechte auch selbst aktiv verteidigen.
- Viele Leute legen ihre Daten *völlig ungeschützt* und *für alle lesbar* ab.

Motto

Zeige mir deinen Web-Browser-Cache und ich sage dir, was für ein Mensch du bist.

Von jedem lesbare Daten von Informatikstudierenden an der TU Berlin.


```
murmel:~ tantau$ ssh conde.cs.tu-berlin.de
tantau@conde.cs.tu-berlin.de's password:
Last login: Tue Apr 29 13:31:48 2008 from murmel.tcs.uni-
Sun Microsystems Inc.   SunOS 5.10      Generic January 2005
conde tantau 1 (~): cslocate sex
...
/home/all/b/believer/.kde/share/apps/krn/alt.sex.homosexual
/home/all/b/belo/man/manl/sex.l
/home/all/m/mawia/bin/BORUSSIA/XP/profile/Cookies/mawia@counter6.sextracker[1].txt
/home/all/m/mawia/bin/BORUSSIA/XP/profile/Cookies/mawia@rb4.worldsex[2].txt
/home/all/m/mawia/bin/BORUSSIA/XP/profile/Cookies/mawia@sextracker[1].txt
/home/all/m/mayer/.kde/share/cache/favicons/sextensiv.com.png
/home/all/s/salou/.kde/share/cache/favicons/www.sexmosaic.com.png
/home/all/s/schaefni/ml_projekt/doku/emerkmalsextraktion.tex
/home/all/s/seemanns/.kde/share/cache/favicons/www.perfectsexteens.com.png
/home/all/s/seemanns/.kde/share/cache/favicons/www.sexycrazybabes.com.png
/home/all/t/tabet/BORUSSIA/DOTNET/profile/Cookies/tabet@counter.sexsuche[1].txt
/home/all/t/thommy/.sigfiles/sig.Linuxsex
/home/all/t/thommy/.sigfiles/sig.alt.sex
/home/all/t/thommy/.sigfiles/sig.computer.sex
/home/cis/cissoft/.netscape/xover-cache/host-/alt.homosexual.snm
/home/cis/cissoft/.netscape/xover-cache/host-/alt.politics.homosexuality.snm
/home/cis/cissoft/.netscape/xover-cache/host-/alt.politics.sex.snm
/home/cis/cissoft/.netscape/xover-cache/host-/alt.sex.bestiality.snm
/home/cis/cissoft/.netscape/xover-cache/host-/alt.sex.bondage.snm
/home/cis/cissoft/.netscape/xover-cache/host-/alt.sex.graphics.snm
/home/cis/cissoft/.netscape/xover-cache/host-/alt.sex.homosexual.snm
/home/cis/cissoft/.netscape/xover-cache/host-/alt.sex.masturbation.snm
...
```

Die größte Sicherheitslücke: Der Mensch.

- Die größte Gefahr für Systeme geht häufig von den *Benutzern* aus.
- In Unternehmen können die *eigenen Mitarbeiter* ihren Zugriff nutzen, um Daten oder gleich die ganze Hardware zu stehlen.
- Menschen können *auf Zettel aufgeschriebene Passwörter* ausspionieren.
- Menschen benutzen oft ganz *leicht zu erratende* Passwörter wie `gott` oder auch `26121975`.

Moral

Der »Faktor Mensch« muss in jedes Sicherheitskonzept einbezogen werden.

Die zweite Sicherheitslücke: Bösartige Programme.

- Damit bösartige Software überhaupt zum Zuge kommt, müssen sie erstmal *ausgeführt* werden.
- Normalerweise geschieht dies *nicht freiwillig* durch den Benutzer oder das Betriebssystem.
- Vielmehr nutzt bösartige Software so genannten *Sicherheitslöcher* aus (dazu gleich mehr).

Glossar der Schädlinge.

Wurm Programm, das sich selbstständig über ein Netzwerk ausbreitet, indem es Kopien von sich selbst an andere Rechner schickt.

Trojaner Programm, das etwas sinnvolles oder hübsches macht, aber eine Schadensroutine enthält (normalerweise den Rechner zum Zombie macht).

Virus Programmteil, der eine Kopie von sich selbst an andere Programme anhängt und immer dann gestartet wird, wenn ein infiziertes Programm gestartet wird.

Zur Diskussion

Wie bekommt man diese Schädlinge?

Leider kein Science-Fiction: Der Wurm der Apokalypse.

Eine Studie der Wurmforscher des ICSI (International Computer Science Institute, Berkeley) ergab 2005 folgendes:

- Würmer sind in der Regel *extrem schlecht* und schlampig programmiert.
- Ein *sehr gut programmierter Wurm*, der alle bekannten Tricks nutzt und in ein einziges IP-Paket passt, könnte sich in ca. *30 Sekunden weltweit* ausbreiten.
- *In wenigen Minuten* könnte er das Internet komplett lahmlegen.
- Lädt er noch einen Firmware-Flasher nach (sehr schwierig), dann könnte er die weltweite IT-Infrastruktur *für Wochen lahmlegen*.
- Die Folgen für die Weltwirtschaft könnte man wohl als apokalyptisch bezeichnen.

Gegen Datenverlust helfen nur Sicherungskopien.

Daten müssen regelmäßig gesichert werden. Punkt.

Glossar der Sicherungsmaßnahmen.

Kennwörter Systeme werden zum Schutz in verschiedene *Bereiche* aufgeteilt, zu denen man nur mittels des richtigen Kennworts Zugriff bekommt.

Firewall Programm oder Rechner, der die Verbindung eines Rechners oder eines Teilnetzes zum Internet überwacht. Er lässt nur als *sicher eingestufte* und *vertrauenswürdige* Kommunikation zu.

Virenscanner Programm, das Speicher und Festplatte nach den ihm bekannten Viren, Würmern oder Trojanern durchsucht.

IDS Intrusion-Detection-Systeme beobachten das Verhalten von Rechner(netzen). Im Falle von auffälligem Verhalten (beispielsweise massenhafte E-Mails) wird der Rechner gestoppt oder verlangsamt.

Das Sicherheitsloch »SQL-Injection«.

- SQL-Injection ist ein Methode, Schwachstellen von *Web-Servern* auszunutzen, die auf eine *SQL-Datenbank* zugreifen.
- Die *Angreifer* sind Menschen oder Computer.
- Die *Angegriffenen* sind Web-Server.
- Ziel des Angreifers ist es, den Web-Server dazu zu bringen, dass er *SQL-Code des Angreifers ausführt*.
- Man sagt, der Angreifer *»injiziert SQL-Code in den Web-Server«*, daher der Name.

Ablauf einer SQL-Injection.

Normale Kommunikation

1. Nutzer trägt Daten in ein Web-Formular ein
2. Web-Client schickt Formular-Daten an den Web-Server
3. Web-Server führt daraufhin einen SQL-Befehl aus, um Daten aus der Datenbank zu holen
4. Web-Server schickt Antwort an den Web-Client

Kommunikation mit SQL-Injection

1. Nutzer trägt *ungewöhnliche Daten* in ein Web-Formular ein
2. Web-Client schickt Formular-Daten an den Web-Server
3. Web-Server führt SQL-Befehl aus, der aber aufgrund der ungewöhnlichen Daten *ungewöhnliche Effekte* hat.
4. Web-Server schickt *nicht gewollte* Antwort an den Web-Client

Das Intranet von Molecular Sheep.

- Die Firma Molecular Sheep verfügt über ein *Intranet*.
- Dieses bietet Zugang auf firmeninterne Daten (wie die Fellfarben von Dolly und Flauschi).
- Man muss sich *authentifizieren*, um Einlass zu erhalten.
- Die Liste der berechtigten Personen und deren Passwörter ist in einer Datenbanktabelle gespeichert.
- Leider wurde an der Sicherheit gespart, weshalb die Eingangskontrolle schlampig programmiert wurde.

Der Login-Vorgang von Molecular Sheep.


```
<!-- HTML-Login-Seite --!>
<form action="http://molecular-sheep.com/login.java"
      method="post">
  <p>User:     <input name="user" type="text"/> </p>
  <p>Password: <input name="pass" type="text"/> </p>
  <p><input name="submitButton" value="Login" type="submit"/></p>
</form>
```

Wenn sich User `ich` mit dem Passwort `gott` einloggt, wird folgende Anfrage an den Web-Server von Molecular-Sheep geschickt:

```
http://molecular-sheep.com/login.java?user=ich&pass=gott
```

Daraufhin ruft der Web-Server das Programm `login.java` auf mit den Parametern `ich` und `gott` auf. Darin:

```
boolean checkPassword (String user, String password) {
  ...
  String sqlQuery =
    "select␣*␣from␣password_table␣where␣user_name=\"" +
    user + "\"␣and␣password=\"" + password + "\"";

  Statement statement = connection.createStatement();
  statement.executeQuery (sqlQuery);
  if (statement.getMoreResults () == false)
    return false;
  else
    return true;
}
```

Das Sicherheitsloch von Molecular-Sheep.

Login für User `ich` mit Passwort `gott`

Der `sqlString` lautet

```
select * from password_table where
  user_name="ich" and password="gott";
```

Dies liefert genau fann mehr als null Treffen, wenn es den passenden Eintrag in der Tabelle `password_table` gibt.

Login mit SQL-Injection als User mit leerem Passwort

Ein Angreifer tippt nun als »Benutzernamen« folgendes ein:

```
egal" or true; --
```

Dann wird folgender SQL-Befehl ausgeführt:

```
select * from password_table where
  user_name="egal" or true; --" and password="";
```

Da `--` einen Kommentar in SQL beginnt, liefert dies immer Treffer.

Moral

1. Vertraue *niemals* Benutzereingaben!
2. Benutzereingaben dürfen *niemals* ohne besondere Vorkehrungen mit Befehlen vermischt werden.

Little Shop of Horror für Datenschützer: Webseite des Department of Corrections, Oklahoma.

- Im US-Staat Oklahoma sind (im Jahr 2008) die Insassen *aller Gefängnisse* über das Internet zugreifbar.
- Für *jeden Gefangenen* sind über ein *komfortable Suchfunktion* folgende Informationen bequem abrufbar:
  - Name, Geburtsdatum, Rasse (!),
  - Foto(s) des Gefangenen (!!),
  - Komplettes Vorstrafenregister.
- Für ehemalige Sexualstraftäter sind auch *nach der Entlassung (für mindestens 15 Jahre bis lebenslang)* verfügbar:
  - aktuelle Anschrift,
  - Telefonnummer.
- Sexualstraftaten sind dort neben Vergewaltigung auch »distribution of obscene videos« (= Verkauf von Pornos).
- Sehr liebevoll gemacht ist auch die Seite mit dem *Hinrichtungs-Countdown* für die Menschen im Todestrakt.

(Mit dem Betreiben einer solchen Seite würden Sie sich in Deutschland strafbar machen.)

Das Sicherheitsloch des DOC Oklahoma.

- Die Informationen über die (ehemaligen) Gefangenen stehen in einer relationalen Datenbank.
- Die Suche in diesen Daten wird durch eine SQL-Anfrage bewerkstelligt.
- Das (absolut vollkommen unvorstellbar gigantische) Sicherheitsproblem besteht darin, dass die SQL-Anfrage in einen Link eingebettet ist.
- Dieses Sicherheitsproblem bestand zwischen den Jahren 2005 und 2008.
- Das DOC wurde auf das Problem aufmerksam gemacht, reagierte durch (völlig nutzlose) Änderungen der SQL-Anfrage.
- Das DOC löste das Problem erst, als man ihm die (ebenfalls in der Datenbank gespeicherte) Liste der medizinischen Behandlungen des Personals des DOC schickte.

Der Link im Original, umformatiert.


```
<a href="http://docapp8.doc.state.ok.us/pls/portal30/url/page/sor_roster?sqlString=
     select distinct
       o.offender_id,doc_number,o.social_security_number, o.date_of_birth,
       o.first_name,o.middle_name,o.last_name,o.sir_name,sor_data.getCD(race) race,
       sor_data.getCD(sex) sex,l.address1 address,l.city,l.state stateid,l.zip,
       l.county,sor_data.getCD(l.state) state,l.country countryid,
       sor_data.getCD(l.country) country,decode(habitual,'Y','habitual','')
       habitual,decode(aggravated,'Y','aggravated','') aggravated,
       l.status,x.status,x.registration_date,x.end_registration_date,l.jurisdiction
     from registration_offender_xref x, sor_last_locn_v lastLocn, sor_offender o,
       sor_location l, (select distinct offender_id
                        from sor_location
                        where status = 'Verified' and upper(zip) = '73064' ) h
     where lastLocn.offender_id(%2B) = o.offender_id and
           l.location_id(%2B) = lastLocn.location_id and
           x.offender_id = o.offender_id and
           x.status not in ('Merged') and x.REG_TYPE_ID = 1 and
           nvl(x.admin_validated,to_date(1,'J')) >= nvl(x.entry_date,to_date(1,'J'))
           and x.status = 'Active' and x.status <> 'Deleted' and
           h.offender_id = o.offender_id
     order by o.last_name,o.first_name,o.middle_name&sr=yes">
Print Friendly </a>
```

5-Minuten-Aufgabe

Zur Übung

Das Sicherheitsloch (eher Sicherheitsabgrund) dieser Webseite lässt sich ausnutzen, indem Sie einfach eine eigene Web-Seite erstellen, auf der Sie den Link kopieren, dann aber den Link-Text an entscheidenden Stellen ändern.
Was müssen Sie ändern, um

1. die Liste der Gefangenen zu bekommen, die eigentlich von der Liste gestrichen sind?
2. herauszubekommen, welche anderen interessanten Tabellen lesbar sind?
3. Gefangene aus der Datenbank zu löschen?
4. fiktive Gefangen in die Datenbank einzutragen?
5. dem Spuk ein Ende zu bereiten und die ganze Datenbank zu löschen?

Zusammenfassung

1. *Daten, Kommunikation und Rechner* sind vielfältigen Gefahren ausgesetzt, die hauptsächlich von (unvorsichtigen, bösartigen oder dummen) *Menschen* ausgehen sowie von *Würmern*.
2. Gute Passwörter, regelmäßiges Schließen von Sicherheitslöchern und die Benutzung von Verschlüsselung bieten *guten, aber längst nicht perfekten Schutz*.
3. Seine Daten zu schützen sollte genauso selbstverständlich sein, wie das Abschließen der Wohnungstür oder des Fahrrades.
4. Der Staat Oklahoma ist allerdings der Meinung, dass all dies für ihn nicht gilt.