Malware, E-mail, Database Security

Malware

- A general term for all kinds of software with a malign purpose
  - Viruses, Trojan horses, worms etc.
  - Created on purpose
- Can
  - Prevent correct use of resources (DoS)
  - Cause general malfunctions
  - Destroy information
  - Modify information
  - Transmit information to unauthorized parties, also randomly
  - Enable others to have a complete control over a computer
- Secondary effects
  - Can disturb critical systems
  - Our society is controlled by computers

Types of Malware

- Trojan horses
  - Programs that promise something good, but instead or in addition do something nasty when you run them
  - Examples: root kits, games with backdoors, etc.
  - Do not spread automatically
  - Can open existing security vulnerabilities or create new ones
  - Some software allows complete control of the target host
- Spyware / Adware
  - Software, which the user installs and which sends information about the user's actions to a server
  - Often financed by advertising
  - Possibly legal, if the user is told of the privacy issues

…Types of Malware

- Viruses
  - Self-replicating software
  - Attach themselves to other executable content
    - Program files
    - Boot sectors on disks
    - Documents with macro instructions
  - Currently the biggest problem are macro viruses in various Windows software
  - MS Office macros, Visual Basic
- Worms
  - Network aware viruses that propagate as independent programs and usually use security vulnerabilities to enter different host computers

…Types of Malware

- Hostile Java applets and ActiveX components
  - User may not even notice he is running an active component on his computer
  - Component certification doesn't mean the component cannot be a Trojan, it is only intended to identify the source of the software component
  - Java's sandbox is in theory fairly secure, but vulnerabilities have been found
  - ActiveX does not limit the component in any way
- "Remote administration" programs
  - Examples: NetBus, Back Orifice
  - Can be packaged inside Trojan horses

How Does a Virus Spread?

- For example a simple MS-DOS virus:
  - Moves part of a program's binary code from the beginning of the file to the end and places itself in the beginning
  - When the program is run the virus activates and places its code into the computer's memory and hooks to the operating system so that the code is periodically activated
  - Then replaces the original program's code to the correct place in memeory and allows the program to be executed
  - When the OS activates the virus in the memory, it infects other files and possibly performs some additional tasks

How Do Macro Viruses Spread?

- Macros are small application specific programs stored in the application's data files
  - The structured environment and high level services make life easier for the viruses
- The viral macro is stored among other macros and is configured to automatically activate when the document is opened by the application
  - After activation the macro can easily copy itself to other documents
  - The macro can also use other available services and for example send random documents using e-mail

Viruses for Purposes

- Viruses can be used to take over hosts in the network environment
  - E.g. the virus spreads as e-mail and inserts a small network server to the target system
- The captured servers can be used for various purposes
  - DoS attacks
  - E-mail spamming
- Spammers are actually doing this
  - An easy way to take over a large amount of hosts and to use them for sending e-mail
  - This is a business, with some money moving, also illegal in most countries
  - The spam mail is used to gather contact information about potential customers and sold to legitimate companies

How to Defeat Viruses?

- Avoid environments that actively support viruses
  - E.g. Microsoft Office tools
- Promote diversity by using less popular operating systems
  - Popular operating systems are more likely to attract viruses
- Use a virus scanner that knows the signatures of different viruses
  - The virus signature database needs to be updated frequently
  - Virus scanning program manufacturers currently share new viruses efficiently and focus on keeping the scanning programs up to date
  - Heuristic scanning that would recognize “bad intentions” of a program has some applicability
  - The virus scanner can remove the virus from the host file or destroy the file
  - The scanning can be done for every file when it is opened
  - The scanning can also be done to file servers or at firewalls

Securing E-Mail Services

- The e-mail system is a good example of a service, which consists of several applications, protocols and hosts
- There are attacks towards the messages and the system itself
  - Eavesdropping, spoofing
  - Spam
  - Break-ins
- A comprehensive attitude is needed to protect the whole

The SMTP Protocol

- The basic SMTP (Simple Mail Transfer Protocol) used for transmitting mail messages over the Internet contains no security features:
  - It is possible to read your e-mail, as it travels in public networks and is stored in servers, that might be outside your control
  - It is generally easy to send e-mail messages with a forged sender address
- SSH and SSL can be used to protect the protocol sessions protecting messages in transit (but not in servers)
- PGP and S/MIME can be used to encrypt and sign messages, providing end-to-end security

Server Software

- Sendmail is usually delivered as a default SMTP server on UNIX boxes
  - Very powerful and difficult to understand configuration
- Hardening Sendmail
  - If no local delivery (e.g. relay only), can be run as non-root
  - Can be run chrooted
- Replacing Sendmail
  - One alternative for UNIX platforms would be qmail
- Most software (SMTP, POP and IMAP servers) have had major security problems
  - Buffer overflows etc.

Handling Spam

- The SMTP server must be configured to avoid spam
  - No relaying by default
    - Nobody should be able to use our server to spam others
  - Sender domain must resolve from DNS
    - Sender addresses must be replyable
  - A whitelist or a blacklist of known good/bad addresses can be used
    - E.g. ADSL customers should not send e-mail directly, but use the operator's server
- Heuristic (Bayesian) analysis
  - A system which learns to recognize spam
  - Must be trained
  - Useful to analyze incoming mail

Database security

- Databases contain organized sets of data inside an application
  - The application implements its own access control to the data
  - Here we are not interested in the known problem of somebody accessing the file where the database data is stored
- All aspects of the CIA model are relevant here
- Databases often have legal protection requirements related to privacy rights

TEKNILLINEN KORKEAKOULU

Relational Databases and SQL

- Data is organized to tables that have records
  - Records of one table have identical structure, but different content
- Actual data is stored in fields that make up the record
- Fields in records can contain values that point to different records in different tables
- Structured Query Language is the most common user interface to databases
  - Access requests are expressed in this language

Consistency

- Integrity of the information in a database is usually considered the most important criteria
  - Remember the CIA model
- Internal consistency means having a set of rules for the database data
  - E.g. the price of an item can not be a negative value
- External consistency means that the data is consistent with an outside world view
  - Can not be enforced by the database itself

SQL Access Control

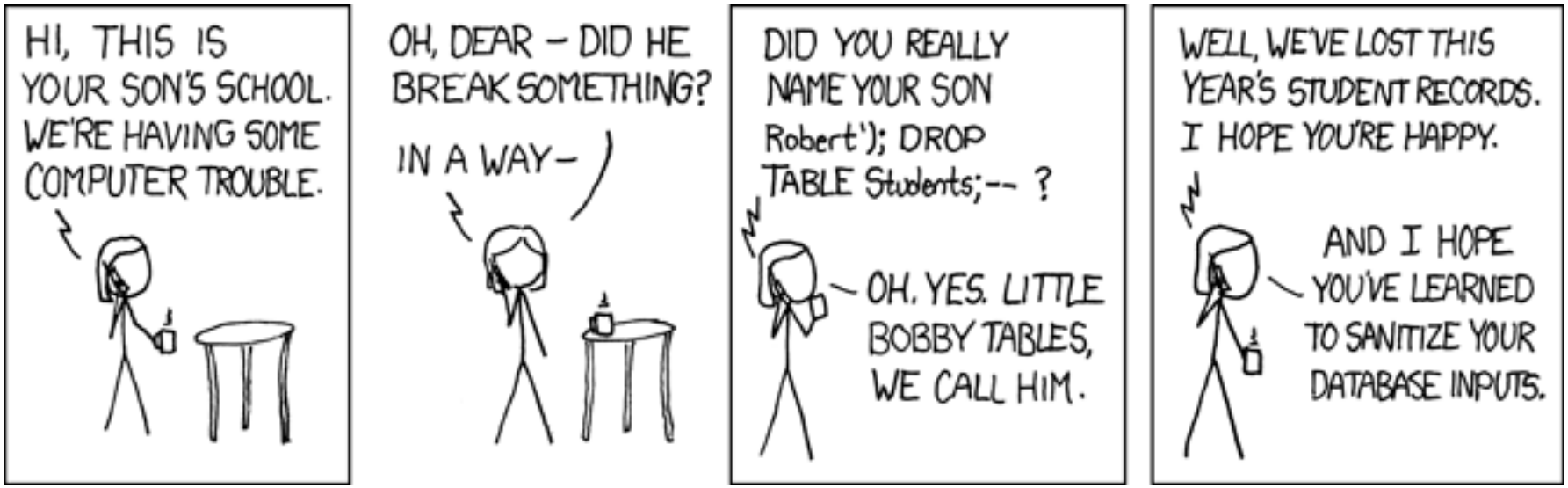
- Users of the database are authenticated
- Any use of the database must be allowed by a privilege:

  (grantor, grantee, object, action, grantable)
  - Privileges start from the administrator and if (grantable=true) can be extended to other users
  - Objects are tables or views
  - *Select* is the action to retrieve (read) data,
  - *Insert*, *update* and *delete* actions change the data
- SQL *view* is a pre-defined selection from a table
  - Can be easily used to limit access to only part of the data

Services Using Databases

- The database is usually the back-end of a system
- The front-end is an application using the database
  - E.g. a WWW based service
- Communicate using SQL
- Security is controlled by the front-end application
  - Receives user input and translates it to SQL commands
- The database can (and should) have its own rules
  - E.g. the front-end can *insert* a credit card number, but only retrieve the last five digits of the number

SQL Injection Attack



- http://www.xkcd.com/327/
- Enter SQL commands as data to an application, hoping they can be passed to the database
  - Any user input must be sanitized before passed on to the database

SQL Injections

- Destroying data as previously seen
  - Effects can be limited by using database access control
- Reading a whole database table
  - Entering name: "Alice' OR 1 = 1 ; --" generates a query:

```
SELECT * FROM customers WHERE ID='Alice' OR 1 = 1;--
```

- Gaining access

  http://www.example.com/index.php?username=1'%20or%20'1'%20=%20'1&password=1'%20or%20'1'%20=%20'1

- Entering data etc.
- SQL injection is a powerful attack caused by the feature of SQL being an interpreted language instead of a protocol
  - Generally the more expressive a language is, the more room it has for misuse

Preventing Injection

- Scrub your input carefully
  - Using a whitelist instead of a blacklist
- Analyze the path of the user input through the code
  - How is user input handled and evaluated
  - Are there processes like converting %0A notation, that can be exploited after input has been scrubbed
- Encapsulate user input, when appropriate and escape harmful characters
  - This approach is not sufficient alone, as it implies a blacklist and it can be exploited
- Avoiding dynamic processing of data
- Using parameterized APIs designed to prevent injection
  - API verifies that each unit of input contains only one parameter for the SQL request, not a SQL statement

Statistical Database Attacks

- Information that is not directly available can be *inferred* from the database
- *Aggregates* are statistical *views* to the database
  - E.g. a user may not request the individual scores of students, but may request the average, highest score and lowest score
- Direct attack: generate a query, that asks for the average from only one record
  - This is commonly thwarted by requiring that aggregates can only be provided if the query matches a certain amount of records (often at least three)
- Indirect attack: deduct the data from multiple queries
  - No general solution exists against this, specific solutions can be designed

Summary

- Viruses exist
- Attacks exist
- Spam exists
- There is no brief solution to these problems, instead we should protect ourselves
- Databases contain data that is often the target for attackers
  - Protection should be implemented in both the front-end application and the database rules
  - Besides direct attacks confidentiality can be attacked indirectly

TEKNILLINEN KORKEAKOULU

Questions For the Exam

- How does malware spread, give at least three examples
- If you were programming a WWW based application based on an SQL database, how would you protect the data in the database, give at least two examples