Insecure PCs

- virus
- malware
- phishing
- spam
- spyware
- botnets
- Trojan horse
- worm
- buffer overflow
- DoS attack
- cross-site scripting
- identity theft
- DNS spoofing
- keyloggers
data theft (inc. ID theft)

phishing

exploit software vulnerabilities

DNS spoofing

hardware keylogger

XSS

install malware

DDoS

send spam

create botnet

destruction (inc. vandalism)

promotion

α → β means β is a possible way to achieve α
• Malware is
  – software intended to intercept or take partial control of a computer's operation without the user's informed consent.
  – It subverts the computer's operation for the benefit of a third party.

• Also called spyware.
  – The term “spyware” taken literally suggests software that surreptitiously monitors the user. But it has come to refer more broadly to any kind of malware,

• Malware covers all kinds of intruder software
  – including viruses, worms, backdoors, rootkits, Trojan horses, stealware etc. These terms have more specific meanings.
How malware spreads

• Trojan horse
  – a malicious program that is disguised as useful and legitimate software. Can be part of, or bundled with, the carrier software.

• Virus
  – Self-replicating program that spreads by inserting copies of itself into other executable code or documents.

• Worm
  – Self-replicating program, similar to virus, but is self-contained (does not need to be part of another program). Spreads by exploiting service vulnerabilities.

• Drive-by
  – installs as side-effect of visiting a website; exploits browser vulnerability.
Why does this problem exist?

Why can't engineers create systems that are not vulnerable to this plethora of attacks?

Compare:
- cars
- aircraft
- telephone system
- electricity production
We have the technology...

<table>
<thead>
<tr>
<th>Attack</th>
<th>Defence</th>
</tr>
</thead>
<tbody>
<tr>
<td>malware</td>
<td>• digital signatures for code</td>
</tr>
<tr>
<td></td>
<td>• anti-virus software</td>
</tr>
<tr>
<td>phishing</td>
<td>• encrypted traffic</td>
</tr>
<tr>
<td></td>
<td>• key certificates</td>
</tr>
<tr>
<td></td>
<td>• education</td>
</tr>
<tr>
<td>DNS spoofing</td>
<td>• key certificates</td>
</tr>
</tbody>
</table>
Why does this problem exist?

- complexity
- immaturity
  - of technology: “release and fix”
  - of designers/programmers: bad culture
  - of users: a new one born every day...
- open platform
- monoculture
How to create an undetectable backdoor:

- Change the compiler so that, when compiling the login program, it adds the hard-coded username/password check to the login program.
  - Thus, the login program source code looks completely normal.
- As an extra twist, change the compiler so that, when compiling the compiler, it adds the code to add the code to the login program.
  - Thus, even if the compiler is recompiled, the backdoor will still be inserted.
  - And none of the source code reveals the backdoor.

What's in this module?

- Cryptography
- Malware
- Authentication
- Access control
- Protocols
- Hardware-based security
  - The TPM
- Electronic voting