Spezielle Kapitel aus Betriebssysteme: Secure Code

LVA 353.013

Fundamentals

secure: [si-'kyur]
1: free from danger
2: free from risk of loss
3: affording safety

Introductions

Andreas Schabus

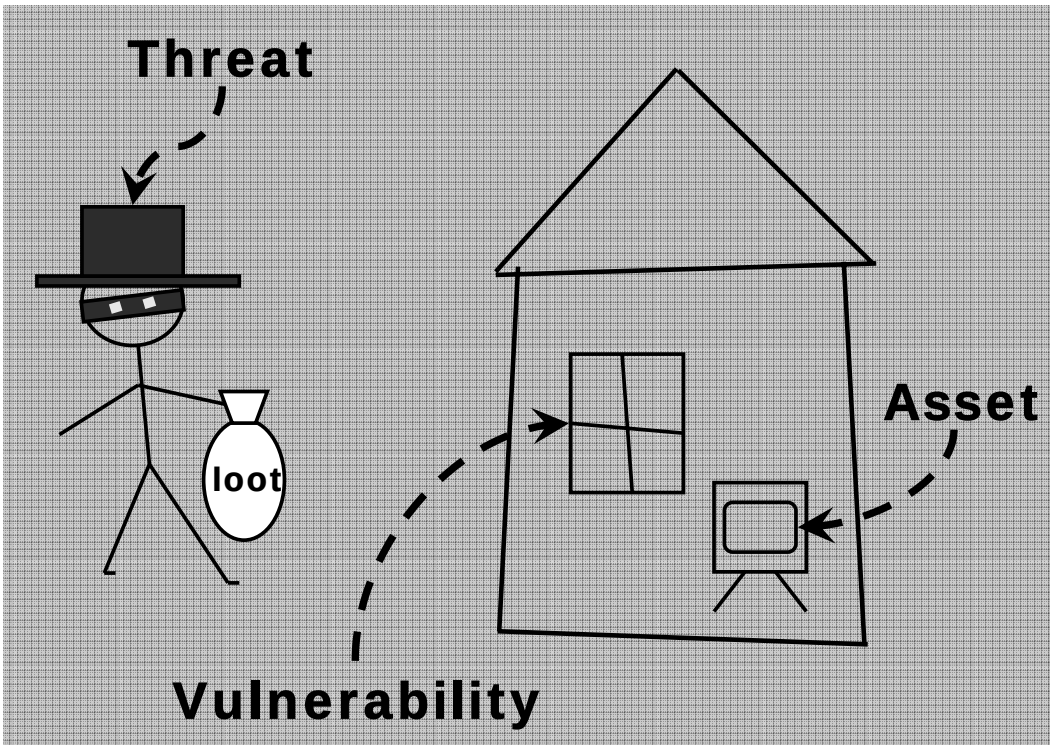
Resources

- Textbooks
  - Michael Howard et.al., The 19 Deadly Sins of Software Security, Osborne McGraw-Hill
  - Michael Howard and David LeBlanc, Writing Secure Code, 2nd Ed.,Microsoft Press.
  - Franz Swiderski and Window Snyder, *Threat Modeling*, Microsoft Press.

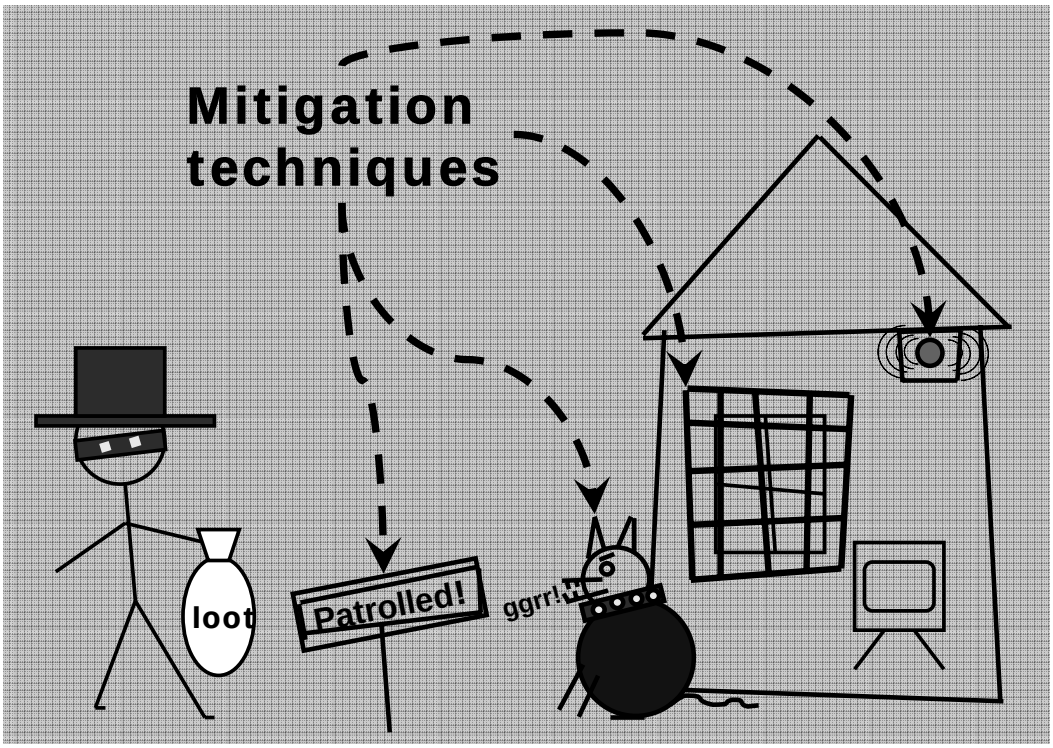
What is the course all about?

Security Terms

Security Terms

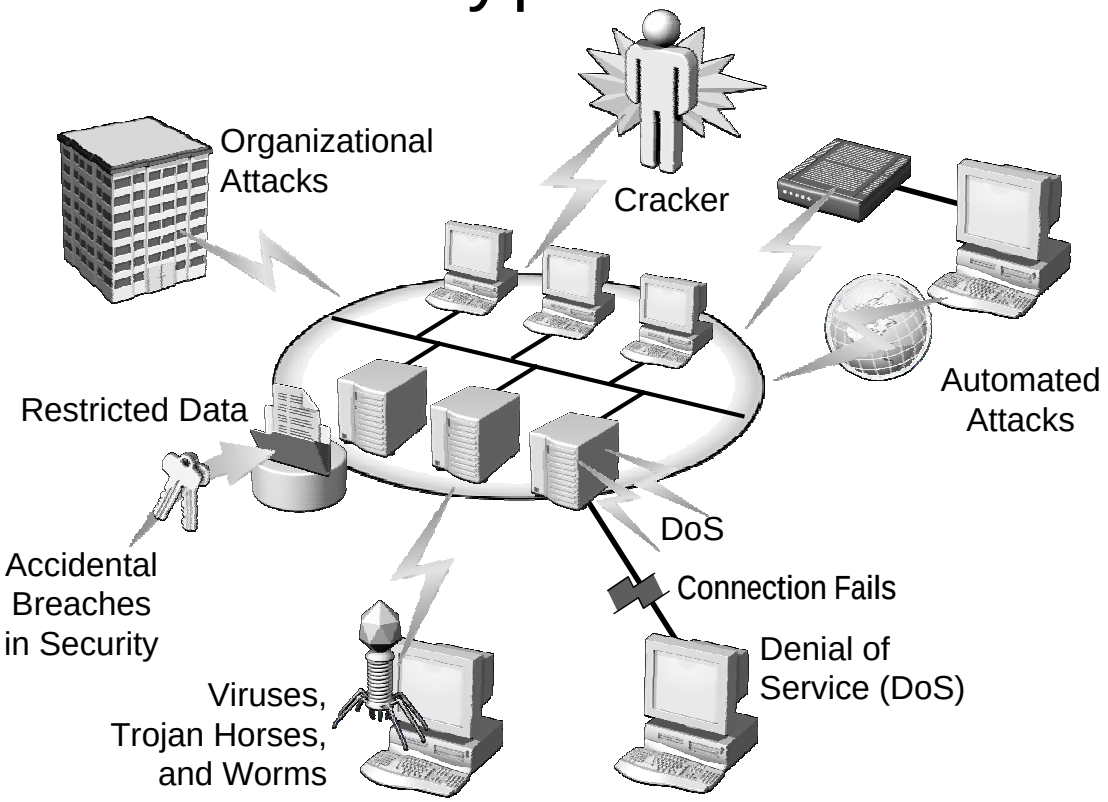


Security Terms

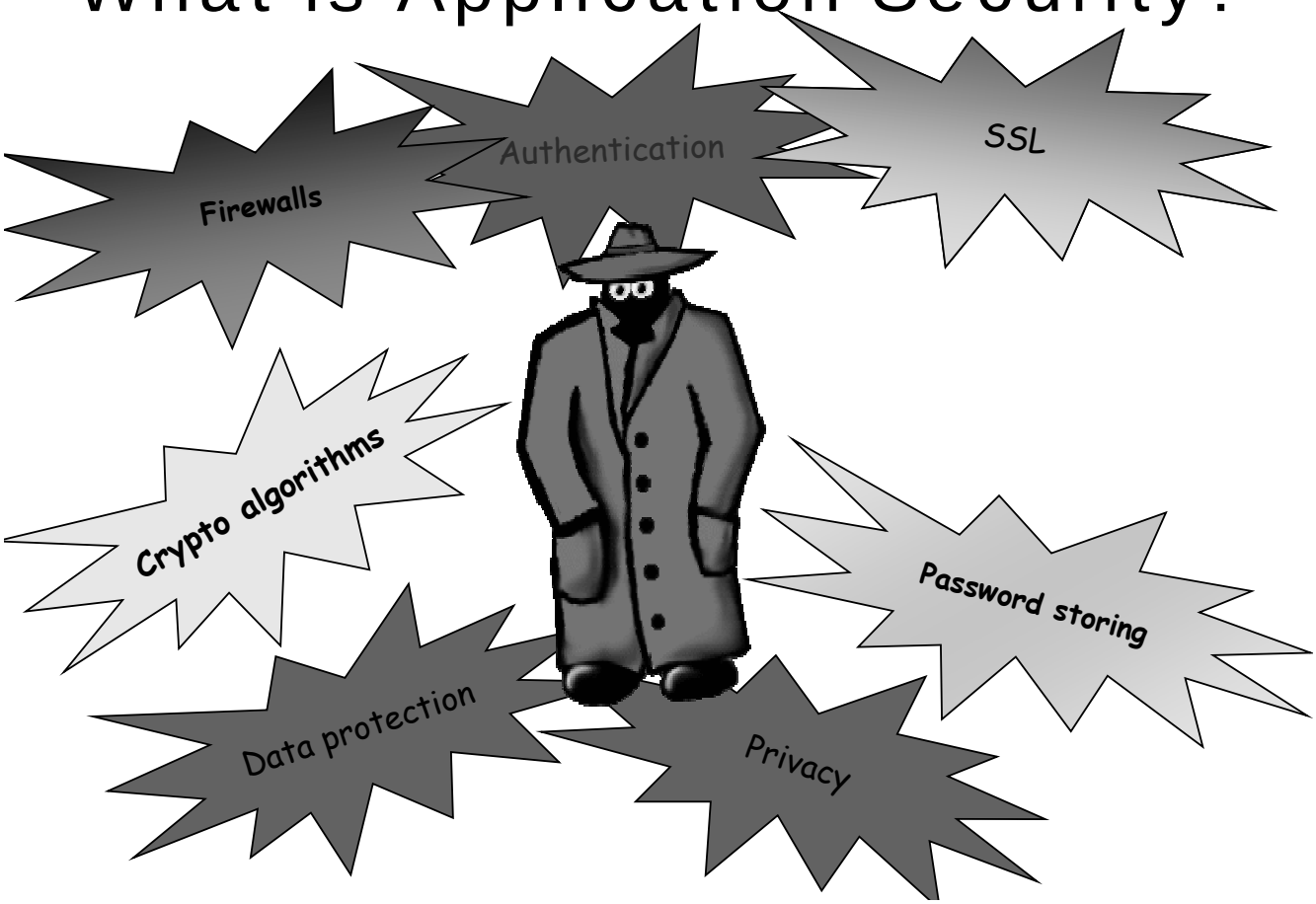


Hacker, Cracker, Shellcoder, …

Common Types of Attack



What is Application Security?



Where are we?

Security Breaches

Affecting Businesses and Consumers

CNNMoney

40M credit cards hacked

Breach at third party payment processor affects 22 million Visa cards and 14 million MasterCards.

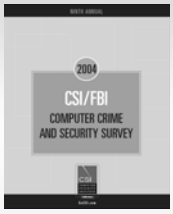
June 20, 2005: 3:18 PM EDT
By Jeanne Sahadi, CNN/Money senior writer

MSNBC HOME

Britain warns of major e-mail attack

Hackers seen aiming at government, corporate networks

The Associated Press
Updated: 1:42 p.m. ET June 16, 2005



In 2004, 78% of enterprises hit by viruses, 49% had laptops stolen, 37% reported unauthorized access to information

--2004 CSI and FBI Computer Crime and Security Survey

CERT/CC Statistics 1988-2005 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites

Address http://www.cert.org/stats/cert_stats.html

msn Search Web Form Fill Net Snippets Add Selection

Related

CERT Contact Information

CERT Statistics

Meet the CERT/CC

CERT/CC Overview and Intruder Trends

CERT Annual Reports

Publications by CERT/CC Staff

Presentations by CERT/CC Staff

Press Releases

Vulnerabilities reported

1995-1999

Year	1995	1996	1997	1998	1999
Vulnerabilities	171	345	311	262	417

2000-2005

Year	2000	2001	2002	2003	2004	1Q,2005
Vulnerabilities	1,090	2,437	4,129	3,784	3,780	1,220

Total vulnerabilities reported (1995-1Q,2005): **17,946**

Done

http://www.securityfocus.com/bid

Search Web Form Fill Viewer Allowed Net Snippets Add Selection

- 2005-05-02: Linux Kernel SYS_EPoll_Wait Local Integer Overflow Vulnerability
- 2005-05-02: Apple Mac OS X Default Pseudo-Terminal Permission Vulnerability
- 2005-05-02: Info-ZIP Zip Remote Recursive Directory Compression Buffer Overflow Vulnerability
- 2005-05-02: MaxWebPortal Multiple SQL Injection Vulnerabilities
- 2005-05-02: ARPUS Ce/Ceterm Insecure Temporary File Creation Vulnerability
- 2005-05-02: ARPUS Ce/Ceterm Command Line Buffer Overflow Vulnerability
- 2005-05-02: ARPUS Ce/Ceterm Environment Variables Buffer Overflow Vulnerability
- 2005-05-02: CodeToSell ViArt Shop Enterprise Multiple Cross-Site Scripting and HTML Injection Vulnerabilities
- 2005-05-02: Mtp-Target Server Memory Corruption Vulnerability
- 2005-05-02: Mtp-Target Client Remote Format String Vulnerability
- 2005-05-02: Kerio Administration Port Denial of Service Vulnerability
- 2005-05-02: Video Cam Server Administrative Interface Authentication Bypass Vulnerability
- 2005-05-02: Video Cam Server Path Disclosure Vulnerability
- 2005-05-02: Video Cam Server Directory Traversal Vulnerability
- 2005-05-02: Linux Kernel it87 and via686a Drivers Insecure File Creation Denial of Service Vulnerability
- 2005-05-02: GlobalSCAPE Secure FTP Server Remote Buffer Overflow Vulnerability
- 2005-05-02: Multiple Vendor Telnet Client Env_opt_add Heap-Based Buffer Overflow Vulnerability
- 2005-05-02: Multiple Vendor Telnet Client LINEMODE Sub-Options Remote Buffer Overflow Vulnerability
- 2005-05-02: Multiple Vendor TCP/IP Implementation ICMP Remote Denial Of Service Vulnerabilities
- 2005-05-02: Horde Forward Module Remote Cross-Site Scripting Vulnerability
- 2005-05-02: Horde Accounts Module Remote Cross-Site Scripting Vulnerability
- 2005-05-02: Horde Chora Remote Cross-Site Scripting Vulnerability
- 2005-05-02: Horde Nag Remote Cross-Site Scripting Vulnerability
- 2005-05-02: Horde MNemo Remote Cross-Site Scripting Vulnerability
- 2005-05-02: Horde Vacation Remote Cross-Site Scripting Vulnerability
- 2005-04-30: Microsoft Windows WINS Association Context Data Remote Memory Corruption Vulnerability
- 2005-04-30: ESRI ArcInfo Workstation Multiple Local Buffer Overflow And Format String Vulnerabilities
- 2005-04-30: PHPMyAdmin Insecure SQL Install Script Permissions Vulnerability
- 2005-04-30: Golden FTP Server Username Remote Buffer Overflow Vulnerability
- 2005-04-30: BulletProof FTP Server Local Privilege Escalation Vulnerability
- 2005-04-30: JGS-Portal ID Variable SQL Injection Vulnerability
- 2005-04-30: DotText HTTP Referer HTML Injection Vulnerability
- 2005-04-30: APSIS Pound Remote Buffer Overflow Vulnerability
- 2005-04-29: Cocktail Admin Password Disclosure Vulnerability
- 2005-04-29: SNMPPD SNMP Proxy Daemon Remote Format String Vulnerability
- 2005-04-29: IBM Lotus Notes Local NOTES.INI Buffer Overflow Vulnerability

How did we get to this state?

1989

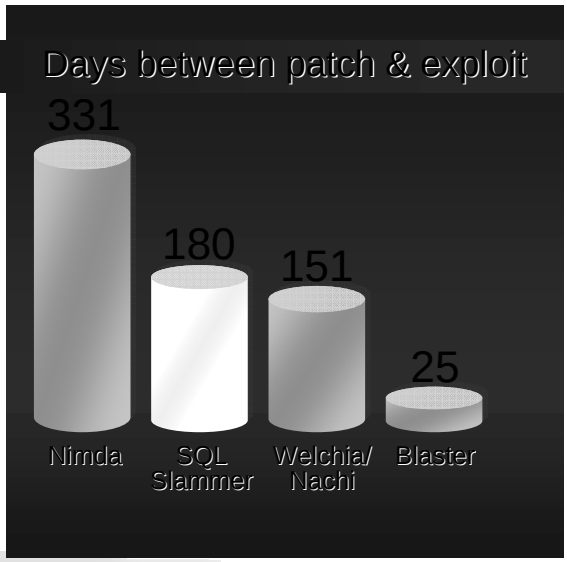
- NT project is 1 year old
- There was no World Wide Web
- TCP/IP was not the default communications protocol
- Virology 101 published, Morris Worm is one year old
- Authentication meant passwords
- DES too heavyweight for most users

2004

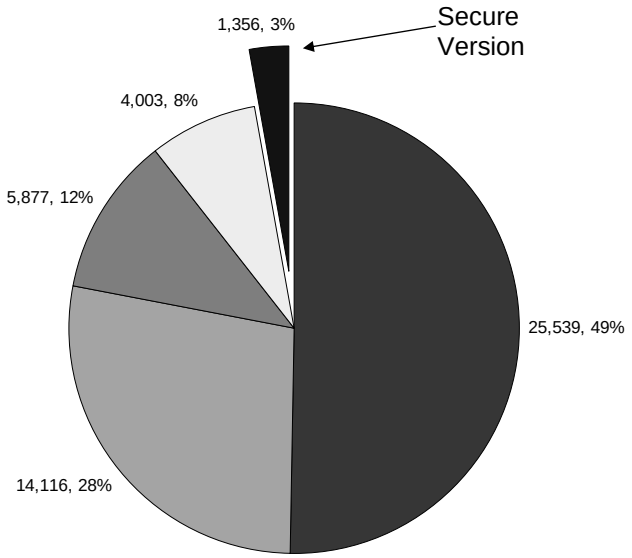
- NT project is 16 years old (We call it Windows now)
- Everyone is on the World Wide Web
- TCP/IP is the default communications protocol
- Threats: Viruses, Worms, Trojans, DOSs
- File Swapping popular
- DES too insecure
- Authentication means PK scheme

The Challenge

- **Patches proliferating**
- **Time to exploit decreasing**
- **Exploits more sophisticated**
- **Customer frustration**



Time To Apply Patch - OpenSSL



49% have not applied a patch in 1 ½ years

0.9.6d and earlier	25,539	30-Jul-02
0.9.6e-h and 0.9.7	14,116	19-Feb-03
0.9.6i and 0.9.7a	5,877	17-Mar-03
0.9.6j and 0.9.7b	4,003	30-Sep-03
0.9.6k and 0.9.7c	1,356	

Source: "Vulnerable versions of OpenSSL apparently still widely deployed on commerce sites" netcraft.com 11/03

The Attacker's Advantage and the Defender's Dilemma

Principle #1:

The defender must defend all points;
the attacker can choose the weakest point.

The Attacker‘s Advantage and the Defender‘s Dilemma

Principle #2:

The defender can defend only against known attacks; the attacker can probe for unknown vulnerabilities.

The Attacker‘s Advantage and the Defender‘s Dilemma

Principle #3:

The defender must be constantly vigilant; the attacker can strike at will.

The Attacker‘s Advantage and the Defender‘s Dilemma

Principle #4:

The defender must play by the rules;
the attacker can play dirty.

Was hat das nun mit Developern zu tun?

demo

DOS Sample

demo

Buffer-Overflow Sample

Developer Security Data Points

“75 percent of hacks happen at the application” - *Gartner “Security at the Application Level”*

“Over 70 percent of security vulnerabilities exist at the application layer, not the network layer” – *Gartner*

"The conclusion is unavoidable: any notion that security is a matter of simply protecting the network perimeter is hopelessly out of date” - *IDC and Symantec, 2004*

“11 of CERT’s 13 major security advisories for 2003 are bugs arising from programming errors in applications [not the OS]” - *Carnegie Mellon University*

“If only 50 percent of software vulnerabilities were removed prior to production … costs would be reduced by 75 percent” *- Gartner “Security at the Application Level”*

Developer Security Data Points

“The battle between hackers and security professionals has moved from the network layer to the Web applications themselves" *- Network World*

“64 percent of developers are not confident in their ability to write secure applications” - *Microsoft Developer Research*

“The Economic Impacts of Inadequate Infrastructure for Software Testing 2002” put the cost of fixing a bug in the field at $30,000 vs. $5,000 during coding - *NIST*

Developer Security Data Points

"By 2006, 80 percent of application development teams will have a person or team responsible for application security." - *Gartner*

"If only 50 percent of software vulnerabilities were removed prior to production … costs would be reduced by 75 percent" *- Gartner "Security at the Application Level"*

"Through 2009, enterprises that do not treat application security as a unified part of a comprehensive development and QA plan are 75 percent more likely to suffer a security-related catastrophic event." - *Gartner "Security at the Application Level"*

"The most damaging targeted attacks — those against specific businesses — have focused on vulnerabilities in Web applications and custom-developed software." - *Gartner "Security at the Application Level"*

Developer Security Data Points

"64 percent of developers are not confident in their ability to write secure applications." - *Microsoft Developer Research*

"The Economic Impacts of Insufficient Infrastructure for Software Testing, removing a software defect after a system is operational can cost two to five times more than if the defect was fixed during final QA testing." - *National Institute of Standards and Technology*

"The cost of fixing vulnerabilities and regression testing the repaired code can be reduced by a factor of at least three by detecting security errors during code and unit tests, compared to finding errors during integration tests. Detecting commonly made coding errors during this phase can also provide feedback to other modules still in design and early coding to avoid repeating the same mistakes." - *National Institute of Standards and Technology*

www.sans.org – 9.5.2005

Address http://www.sans.org/newsletters/newsbites/newsbites.php?vol=7&issue=19#311 Go Links

msn Search Web Net Snippets Add Selection Add Entire Page Add Lin

WORMS, ACTIVE EXPLOITS, VULNERABILITIES, AND PATCHES

Fixes Not Yet Available for Firefox Vulnerabilities (9 May 2005)
Two vulnerabilities in the Firefox web browser could allow attackers to gain control of users' computers just by getting them to visit a maliciously crafted web site. Mozilla is recommending that Firefox users disable Javascript or lock down the browser to prevent it from installing additional software. There is no a patch available, although information about the vulnerabilities and proof-of-concept exploit code have already been released. Mozilla plans to release an update, Firefox 1.0.4, as soon as possible.
-http://informationweek.com/story/showArticle.jhtml?articleID=163100338
-http://www.vnunet.com/news/1162904
[Editor's Note (Schultz): The number of vulnerabilities in Firefox recently has been alarming. At first Firefox appeared to be an attractive alternative to Internet Explorer (IE) for security reasons, but IE is now looking better and better in comparison.

(Shpantzer): There's so much hacking at the application layer, at some point we'll have to actually lock down configurations for all browsers, regardless of the security mythology that surrounds the project's code and architecture. If you have a supposedly 'secure' browser that's insecurely configured, well, it's not very secure.]

The 10 Most Critical Web Application Security Vulnerabilities

A1 Unvalidated Input	SQL Injection, Command Injection, Cross-Site Scripting
A2 Broken Access Control	Improper File Access
A3 Broken Authentication and Session Management	Use of Magic URLs and Hidden Form Fields
A4 Cross Site Scripting (XSS) Flaws	Cross-Site Scripting
A5 Buffer Overflows	Buffer Overruns, Format String Problems, Integer Overflows

http://www.owasp.org/documentation/topten.html

The 10 Most Critical Web Application Security Vulnerabilities

A6 Injection Flaws	SQL Injection, Command Injection
A7 Improper Error Handling	Failing to Handle Errors
A8 Insecure Storage	Failing to Store and Protect Data Securely
A9 Denial of Service	This is the outcome of an attack, not a coding defect.
A10 Insecure Configuration Management	This is an infrastructure issue

http://www.owasp.org/documentation/topten.html

Ridiculous Excuses We‘ve Heard

Excuse:
No one will do that!

Excuse:
Why would anyone do that?

Google Search: inurl:indexFrame.shtml Axis

File Edit View Favorites Tools Help

Back Search Favorites

Address Go

Google

Web Images Groups News Froogle Local New! more »

inurl:index Search Advanced Search Preferences

Web Results **11** - **20** of about **137** for **inurl:indexFrame.shtml Axis**. (**0.10** seconds)

AXIS Video Server
Frame.shtml?newstyle=Quad - 1k - Cached - Similar pages

AXIS Video Server
xFrame.shtml?newstyle=One&cam=1 - 1k - Cached - Similar pages

AXIS Video Server
webcam.gordon.edu/indexFrame. shtml?newstyle=One&cam=1 - 1k - Cached - Similar pages
[More results from webcam.gordon.edu]

AXIS 2130R PTZ Network Camera
Frame.shtml - 1k - Cached - Similar pages

AXIS Video Server
fice.net/ indexFrame.shtml?newstyle=One&cam=1 - 1k - Cached - Similar pages

AXIS Video Server
dexFrame.shtml - 1k - 30 Apr 2005 - Cached - Similar pages

AXIS Video Server
exFrame.shtml - 1k - Cached - Similar pages

AXIS Video Server
me.shtml?newstyle=Quad - 1k - Cached - Similar pages

AXIS Video Server
rame.shtml?newstyle=Quad - 1k - Cached - Similar pages

AXIS Video Server
Frame.shtml?newstyle=Quad - 1k - Cached - Similar pages

◀ Goooooooooooogle ▶

Result Page: **Previous** 1 2 3 4 5 6 7 8 9 10 11 **Next**

AXIS Video Server

File Edit View Favorites Tools Help

Back Search Favorites

Address http://xFrame.shtml Go

AXIS 2400+ Video Server

AXIS COMMUNICATIONS

Video All

Video 1

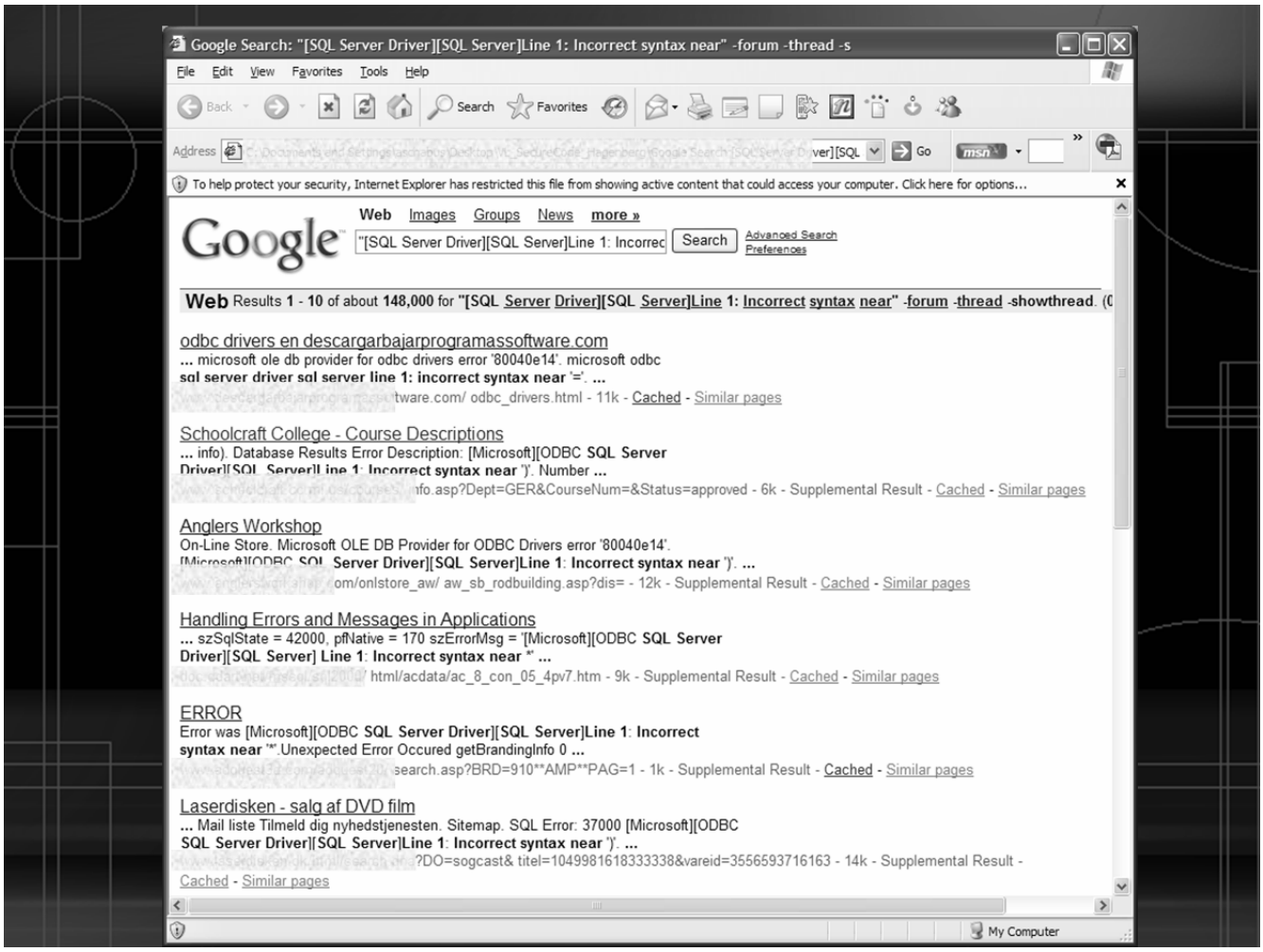
Video 2

Video 3

Video 4

Admin

Done Internet



Excuse:
We‘ve never been attacked

Excuse: We‘re secure –
we use cryptography

demo

Random Numbers

demo

Encryption

Hide & Seek Stored Keys

Figure 1 Key information (in the middle of the figure) looks more noisy than the rest of the data

Excuse: We‘re secure –
we use ACLs

Excuse: We‘re secure –
we use a firewall

demo

SQL Injection

Anatomy of SQL Injections

- Problem: string concatenation

```
strSql = "SELECT * FROM titles " & _
         "WHERE id LIKE '" & textName.Text & "'"
Dim cmd As New SqlCommand(strSql, "server=...")
myReader = cmd.ExecuteReader()
```

Good Guy

ID: 1001

Not so Good Guy

Really Bad Guy

Downright Evil Guy

ID: 1001'; exec xp_cmdshell('fdisk.exe') --

```
SELECT *
FROM titles
WHERE id='1001'; exec xp_cmdshell('fdisk.exe') --'
```

```
ders -- '
```

demo

Cross-Site Scripting

Anatomy of Cross-Site Scripting

- Web based applications
  - Redirect info via *<form>*
  - E-Mail platforms & discussion boards
- Allows hackers to:
  - Execute script in client's browser
  - *<script>, <object>, <applet>, <form>, <embed>*
- Arising threats
  - Steal session / AuthN cookies
  - Access to client computer

Excuse:
We‘ve reviewed the code, and there are no security bugs

“Many Eyeballs Makes all Bugs Shallow”

demo

EBay

Is there a Security Vulnerability?

```
char dest[50], src[100];
int x, y;

if (x=1)
{
   strcpy(dest,src);
   dest[50] = '\0';
}

return y;
```

should be using ==

src is larger than dest

buffer overrun

Returning uninitialized variable

Example: “Evils” of strn...

```
// code prior to this verifies pszSrc
// is <= 50 chars
#define MAX (50)
char *pszDest = malloc(sizeof(pszSrc));
strncpy(pszDest,pszSrc,MAX);
```

The code is allocating the size of a pointer, 4-bytes on a 32-bit CPU, and then trying to copy e.g. 40 bytes.

Example: “Evils” of strn...

```
#define MAX (50)
char szDest[MAX];
strncpy(szDest,pszSrc,MAX);
```

If the length of the string pointed to by pszSrc is exactly MAX, then strncpy does NOT null- terminate szDest.

demo

Culture-Safe Code

```
static bool IsFileURI(string path) {
        return (String.Compare(path, 0, "file:", 0, 5, true) == 0);
    }
```

Scrubbing Secrets in Memory

What's wrong with this code?

```
void Function() {
        char pwd[32];
        GetPwdFromUser(pwd,32);
        UsePwd(pwd,32);
        memset(pwd,0,32);
}
```

Victim of
"dead store removal"
by optimizing compilers

```
void Function() {
        char pwd[32];
        GetPwdFromUser(pwd,32);
        UsePwd(pwd,32);
        SecureZeroMemory(pwd,32);
}
```

Excuse:
We know it‘s the default, but
the administrator can turn it off

Excuse:
If we don‘t run as
administrator, stuff breaks

Excuse:
But we‘ll slip the schedule!

Excuse:
It‘s not exploitable!

Excuse:
But that‘s the way we‘ve always done it

Excuse:
If only we had better tools ...

Address http://www.cs.berkeley.edu/~flab/languages.html Go Links

msn Search Web Form Fill Net Snippets Add Selection Add Entire Page Add Link

Programming Language Usage Graph

by François Labelle

Below is the past and current usage of top computer languages, as defined by statistics on open-source projects at SourceForge. Granted, this does not necessarily correspond to the "real usage out there" (whatever that means) which I don't know how to get.