# Statička sigurnosna analiza izvornog koda

IBM Security Services

Vlatko Košturjak, IBM PSS EMEA
CISSP, CISA, C|EH, MBCI, LPIC-3, ...

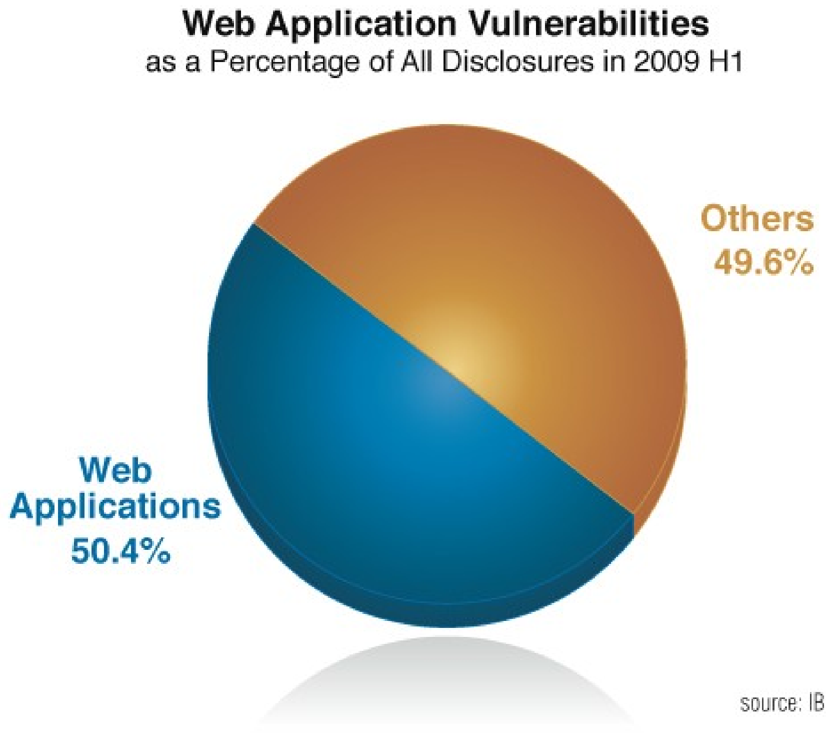
HrOUG, ~~Rovinj~~, Sv. Andrija (Crveni otok), 22.10.2010

# Sažetak

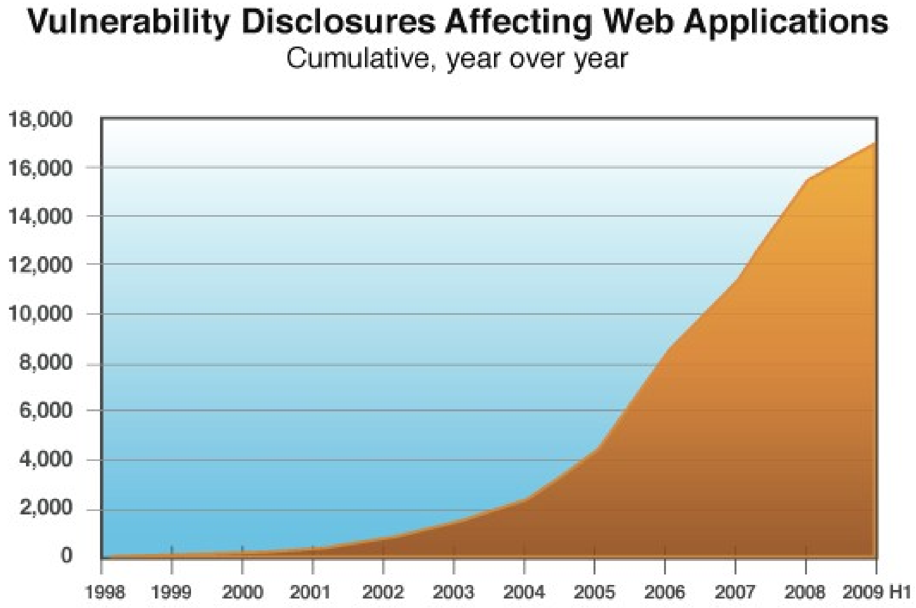
- Uvod
- Zablude
- Analiza izvornog koda aplikacije
- Dinamička vs. Statička analiza
- Tipovi statičke analize
- Alati
  - Otvorenog koda
  - Komercijalni
- Najčešće ranjivosti
- Tijek podatka
- Pitanja i odgovori

30 minuta

# Web ranjivosti rastu... (i bez zalijevanja! :-)

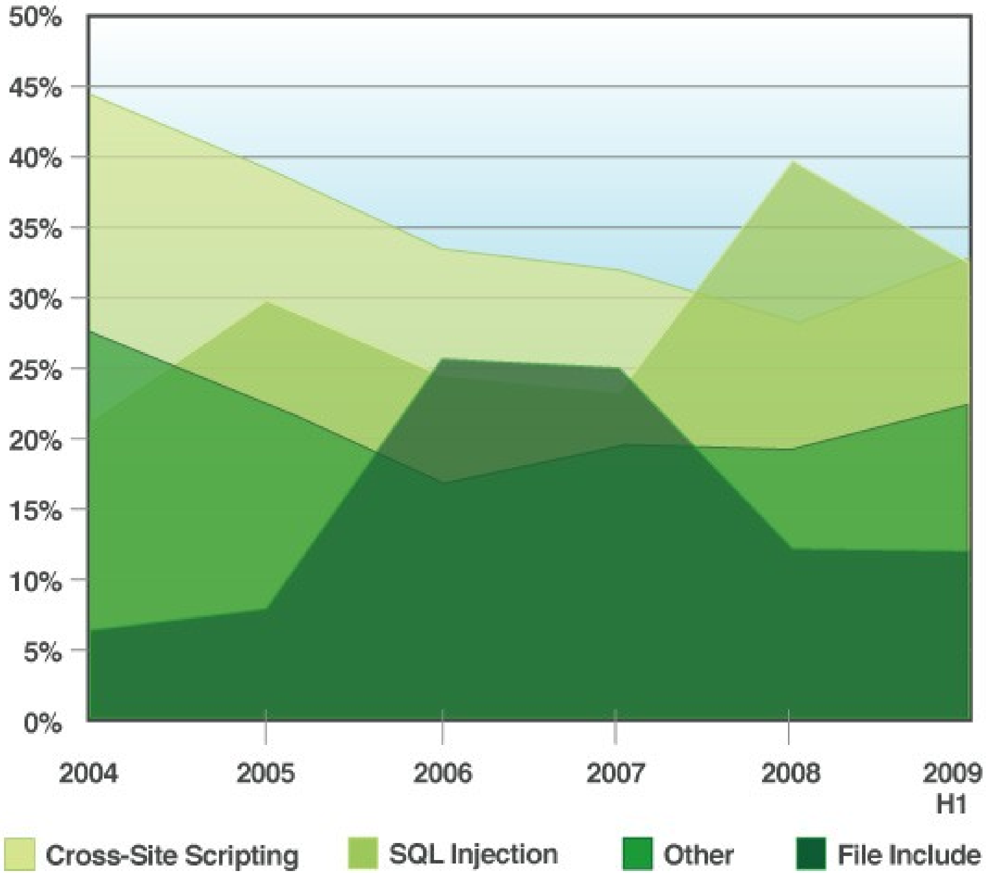


source: IBM X-Force®



source: IBM X-Force®

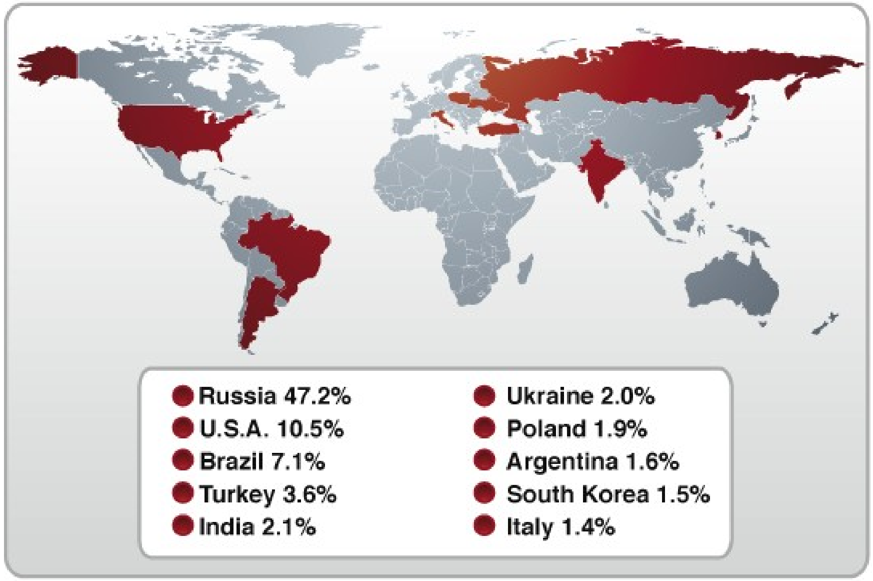
# Najčešće ranjivosti i izvori napada



**Web Application Vulnerability Disclosures**
2004-2009 H1

source: IBM X-Force®

**Geographical Distribution of Phishing Senders**



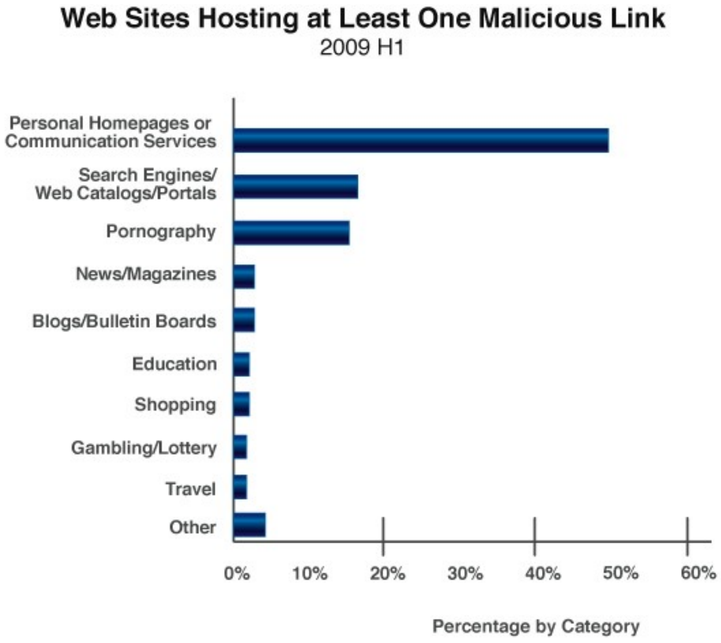
| | |
|---|---|
| ● Russia 47.2% | ● Ukraine 2.0% |
| ● U.S.A. 10.5% | ● Poland 1.9% |
| ● Brazil 7.1% | ● Argentina 1.6% |
| ● Turkey 3.6% | ● South Korea 1.5% |
| ● India 2.1% | ● Italy 1.4% |

source: IBM X-Force®

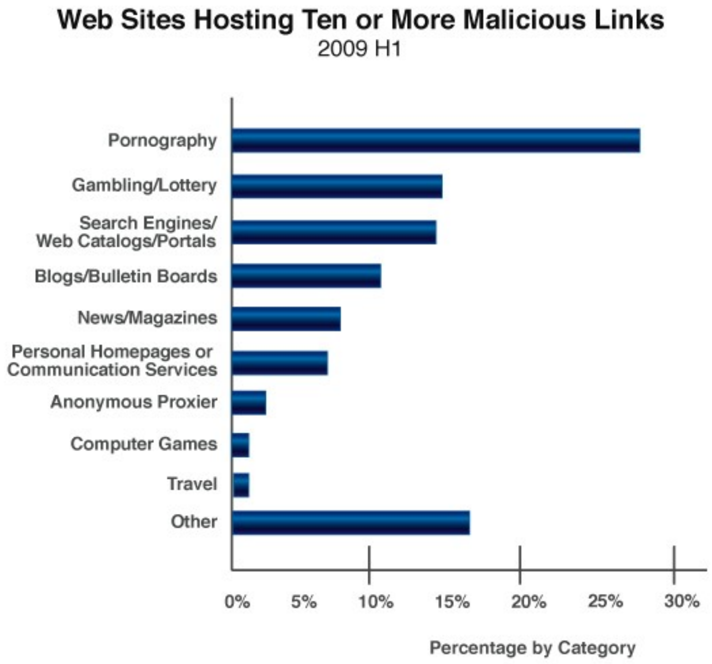
*People & Identity:*

# Good Websites Hosting Bad Links

- Personal homepages (typically hosted by communication service company domains) account for approximately half of all the "trusted" domains hosting at least one malicious link
- As for domains hosting 10 or more malicious links, pornography accounts for nearly **28%** and gambling accounts for more than **14%**



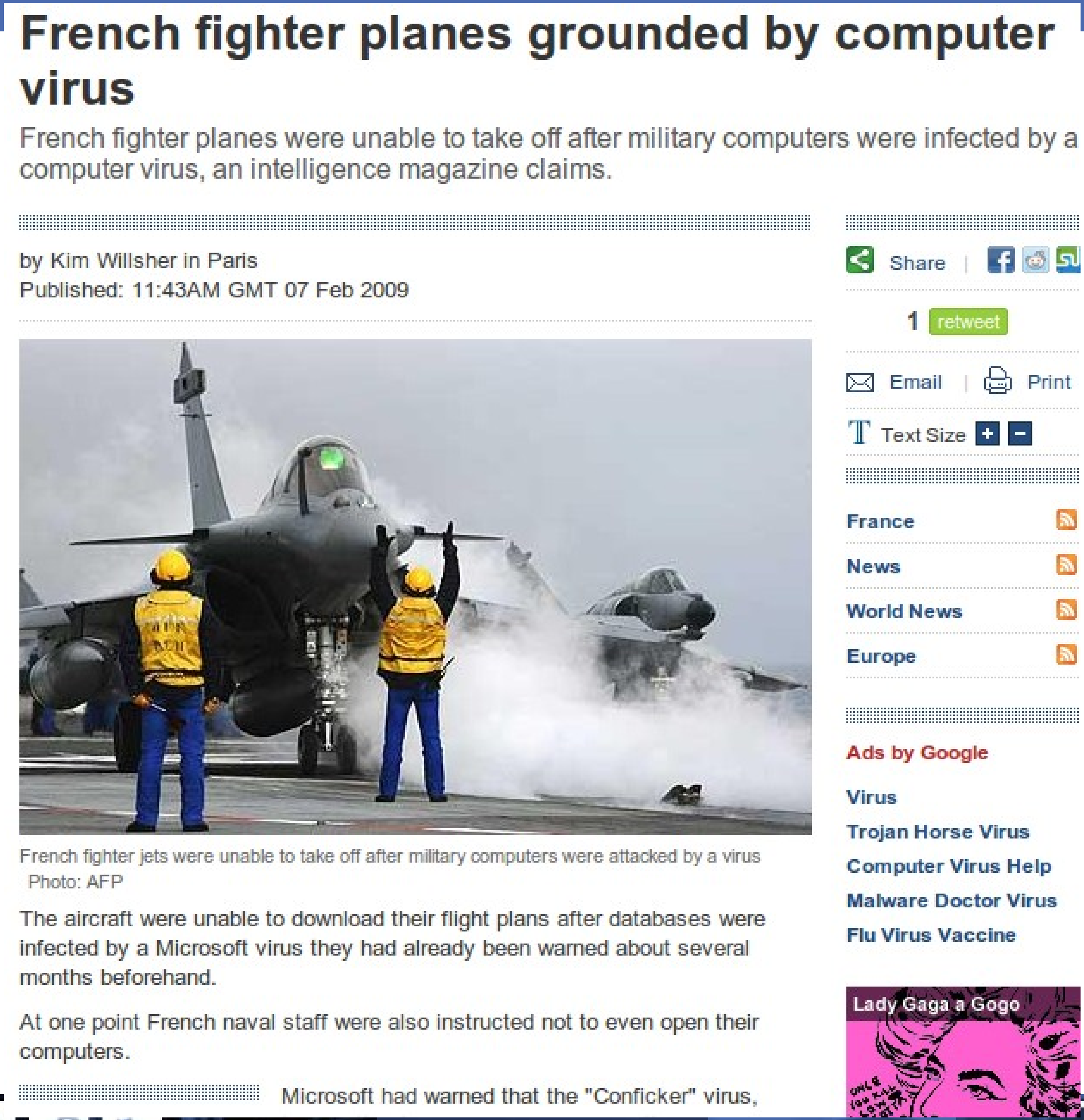
source: IBM X-Force®



source: IBM X-Force®

# Sigurnost i preventiva

- Javna računala
  - Svjetski kriminalci
  - Danonoćno (24/7)
- Napadi
  - Pravo ne radi preko granice
  - Napadači DA :)
- Udaljenost nema razlike
  - Dobro, par milisekundi...
- Dostupni alati za hacking
  - Youtube.com
  - SecurityTube.Net
- ...



# French fighter planes grounded by computer virus

French fighter planes were unable to take off after military computers were infected by a computer virus, an intelligence magazine claims.

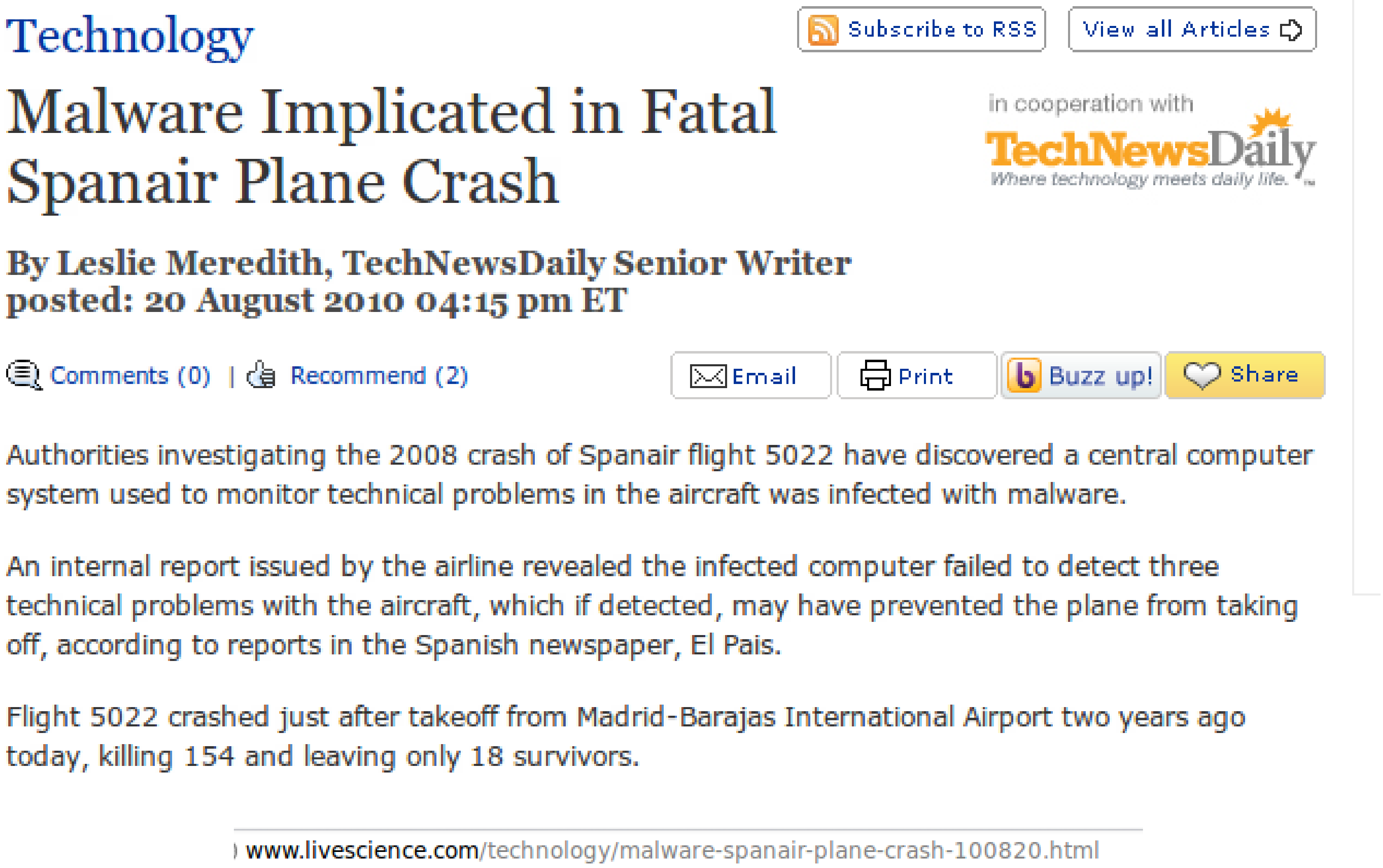
by Kim Willsher in Paris
Published: 11:43AM GMT 07 Feb 2009

French fighter jets were unable to take off after military computers were attacked by a virus
Photo: AFP

The aircraft were unable to download their flight plans after databases were infected by a Microsoft virus they had already been warned about several months beforehand.

At one point French naval staff were also instructed not to even open their computers.

Microsoft had warned that the "Conficker" virus,



Technology

# Malware Implicated in Fatal Spanair Plane Crash

**By Leslie Meredith, TechNewsDaily Senior Writer**
**posted: 20 August 2010 04:15 pm ET**

Authorities investigating the 2008 crash of Spanair flight 5022 have discovered a central computer system used to monitor technical problems in the aircraft was infected with malware.

An internal report issued by the airline revealed the infected computer failed to detect three technical problems with the aircraft, which if detected, may have prevented the plane from taking off, according to reports in the Spanish newspaper, El Pais.

Flight 5022 crashed just after takeoff from Madrid-Barajas International Airport two years ago today, killing 154 and leaving only 18 survivors.

www.livescience.com/technology/malware-spanair-plane-crash-100820.html

# Par zabluda

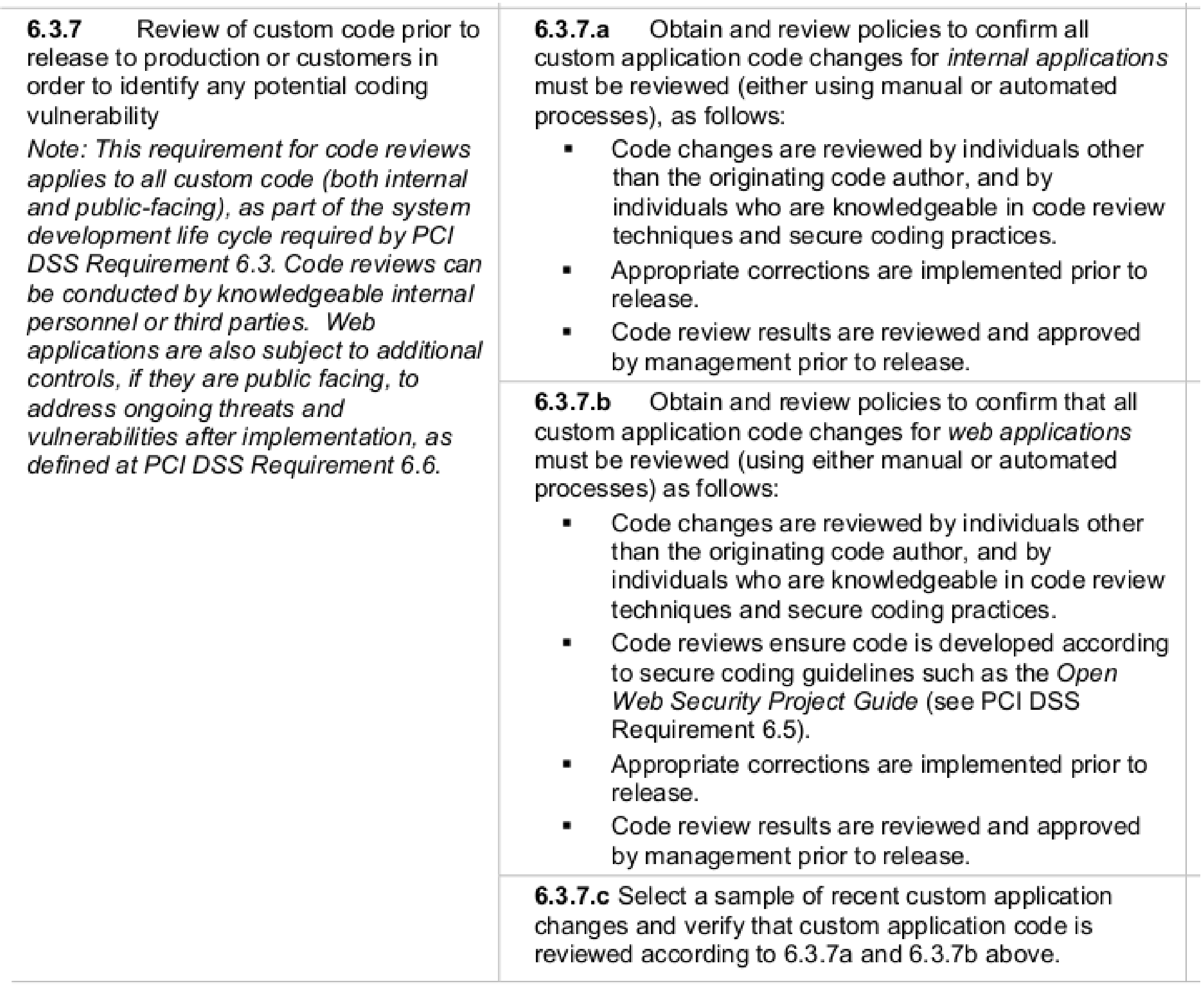
- Aplikacija je prošla penetracijski test
  - Automatskim alatom?
  - S kojom razinom privilegija?
  - Sa svim podacima?
- Bio nam je audit....
  - ...sve OK!
- Imamo vatrozid na razini aplikacije
  - Ispravno implementiran?
  - Ispravno konfiguriran?
- Pregledali smo izvorni kod aplikacije
  - Automatskim alatom?
  - Ručno?

# Penetracijski test?

```
# primjer pseudo koda

if (action==doreport) {
        If (userlevel==bla) {
                If (usergroup==trans) {
                        If (timeofweek=='wednesday') {
                                system(userinput)
                        }
                }
        }
}
```

Osjetljivi dijelovi aplikacije

# Payment Card Industry(PCI) – Data Security Standard(DSS)



| | |
|---|---|
| **6.3.7** Review of custom code prior to release to production or customers in order to identify any potential coding vulnerability<br>*Note: This requirement for code reviews applies to all custom code (both internal and public-facing), as part of the system development life cycle required by PCI DSS Requirement 6.3. Code reviews can be conducted by knowledgeable internal personnel or third parties. Web applications are also subject to additional controls, if they are public facing, to address ongoing threats and vulnerabilities after implementation, as defined at PCI DSS Requirement 6.6.* | **6.3.7.a** Obtain and review policies to confirm all custom application code changes for *internal applications* must be reviewed (either using manual or automated processes), as follows:<br>▪ Code changes are reviewed by individuals other than the originating code author, and by individuals who are knowledgeable in code review techniques and secure coding practices.<br>▪ Appropriate corrections are implemented prior to release.<br>▪ Code review results are reviewed and approved by management prior to release. |
| | **6.3.7.b** Obtain and review policies to confirm that all custom application code changes for *web applications* must be reviewed (using either manual or automated processes) as follows:<br>▪ Code changes are reviewed by individuals other than the originating code author, and by individuals who are knowledgeable in code review techniques and secure coding practices.<br>▪ Code reviews ensure code is developed according to secure coding guidelines such as the *Open Web Security Project Guide* (see PCI DSS Requirement 6.5).<br>▪ Appropriate corrections are implemented prior to release.<br>▪ Code review results are reviewed and approved by management prior to release. |
| | **6.3.7.c** Select a sample of recent custom application changes and verify that custom application code is reviewed according to 6.3.7a and 6.3.7b above. |

## Sigurnosna provjera koda

- Postupak pregledavanja koda s ciljem traženja ranjivosti
- Vrste
  - Statička analiza
    - Kod se pregledava, ali ne izvršava
  - Dinamička analiza
    - Kod se izvršava

## Grep-based

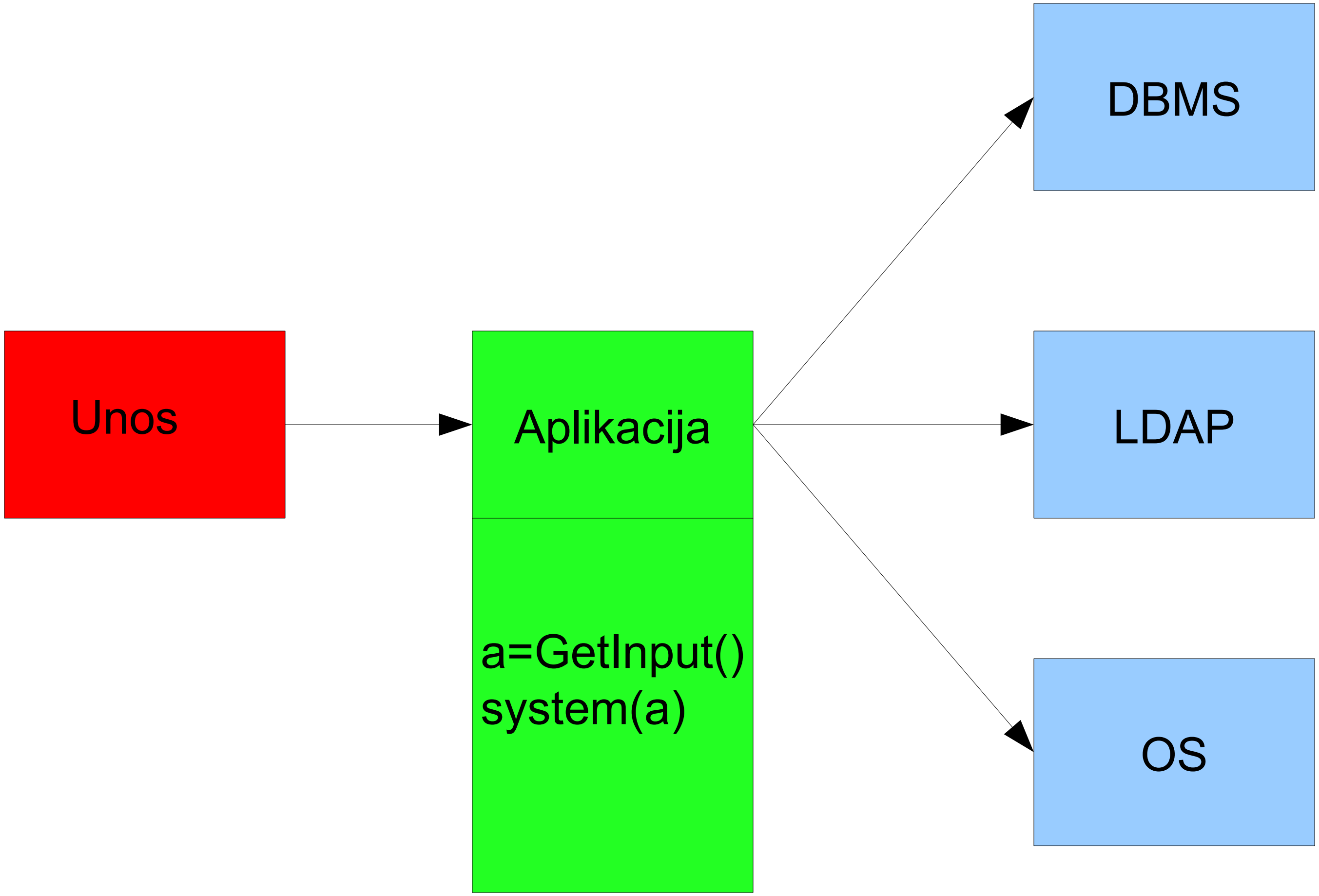
- Traženje "problematičnih" funkcija
  - Buffer overflow f()
  - System execute
- Traženje opasnih funkcija
  - System execute
  - Modify system configuration
  - File write
  - File read
  - ...

## OWASP Top 10  - 2010

- Injection
- Cross Site Scripting (XSS)
- Broken Authentication and Session Management
- Insecure Direct Object References
- Cross Site Request Forgery (CSRF)
- Security misconfiguration
- Failure to Restrict URL Access
- Unvalidated Redirects and Forwards
- Insecure Cryptographic Storage
- Insufficient Transport Layer Protection

http://www.owasp.org/index.php/Croatia

# Taint?



Unos

Aplikacija

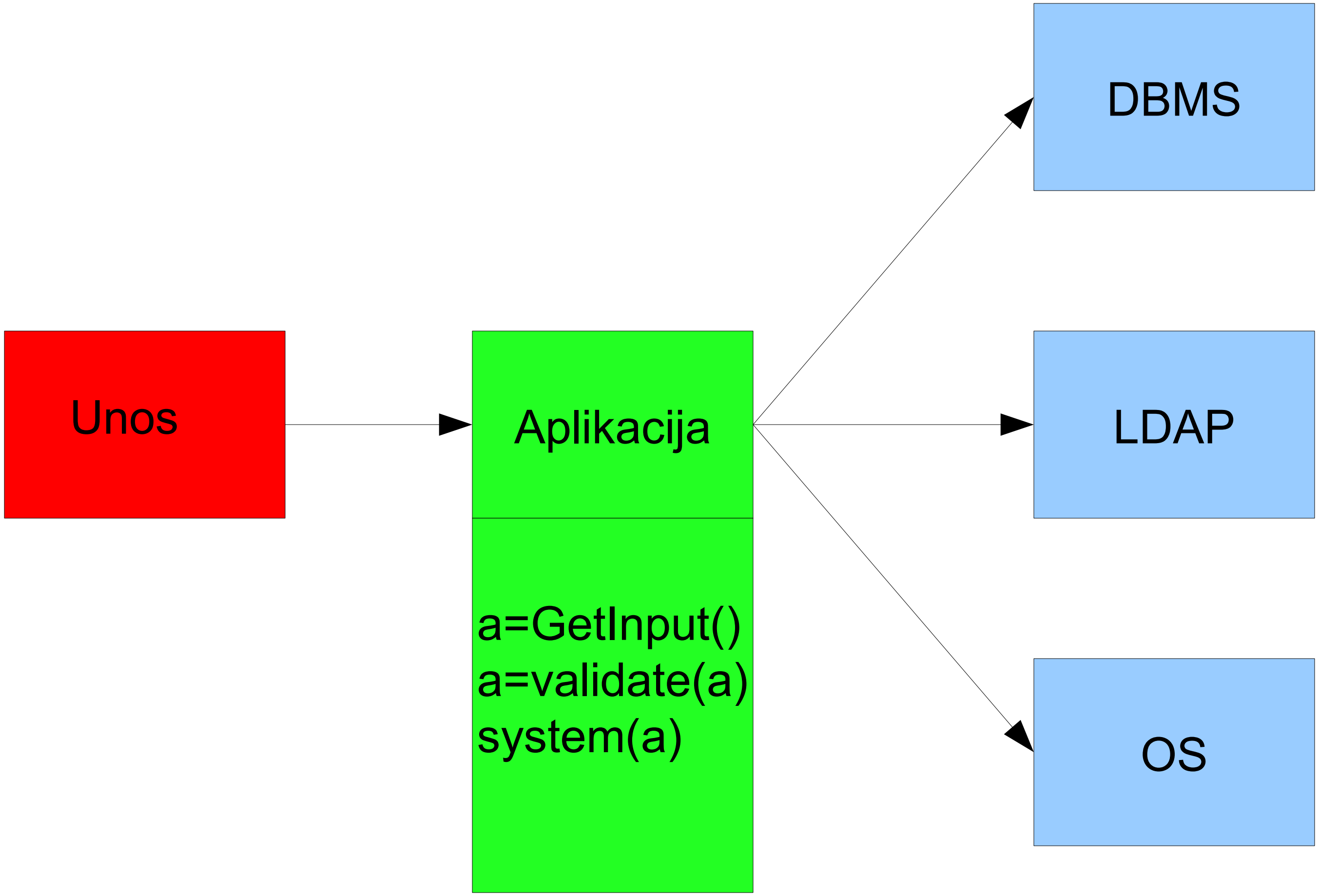
a=GetInput()
system(a)

DBMS

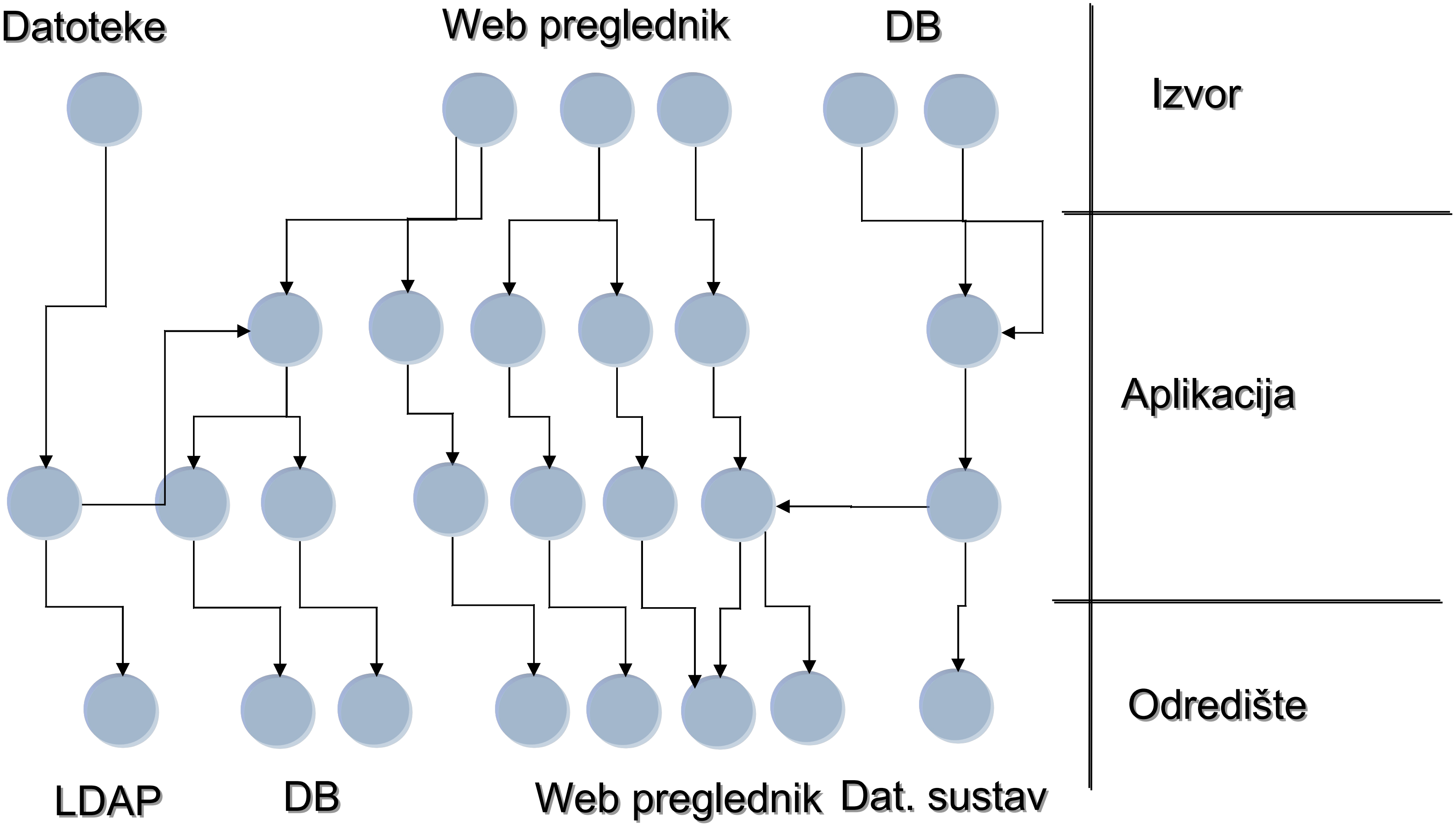
LDAP

OS

# Taint?


Unos
Aplikacija
a=GetInput()
a=validate(a)
system(a)
DBMS
LDAP
OS

# Tijek podatka



Datoteke

Web preglednik

DB

Izvor

Aplikacija

Odredište

LDAP

DB

Web preglednik

Dat. sustav

# Taint checks?

- Ruby
  - -T level
- Perl
  - -t (upozorenje)
  - -T (blokiranje)
- Python?
  - -t

# Alati

- Jednostavni - Unix/Linux/Windows(Cygwin)
  - cat
  - less
  - more
  - grep
  - awk
  - sed
  - vi(m)
  - ...

# Alati: open source

- Flawfinder
- Rats
- C code analysis (CCA)
- Graudit (grep)
- SWAAT (OWASP)
- Yasca
- FindBugs
- pylint
- ...

## Alati: komercijalni

http://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis

- **Rational Appscan**
  - Web based apps
  - JavaScript Static Analyzer (JSA) - 8.0
  - DOM based XSS
- **Rational AppScan Source**
  - ex. Ounce Labs
  - Web and GUI
  - Poluautomatski alat
  - Java, C/C++, PHP, ...

- Fortify (HP)
- Justcode
- Coverity
- Parasoft
- ...

# Application Source Code Security Assessment

## Solution Overview

Services to help clients reduce cost, reduce risk and improve regulatory compliance by assessing the source code of applications to identify security vulnerabilities and provide recommendations for prioritization and resolution.

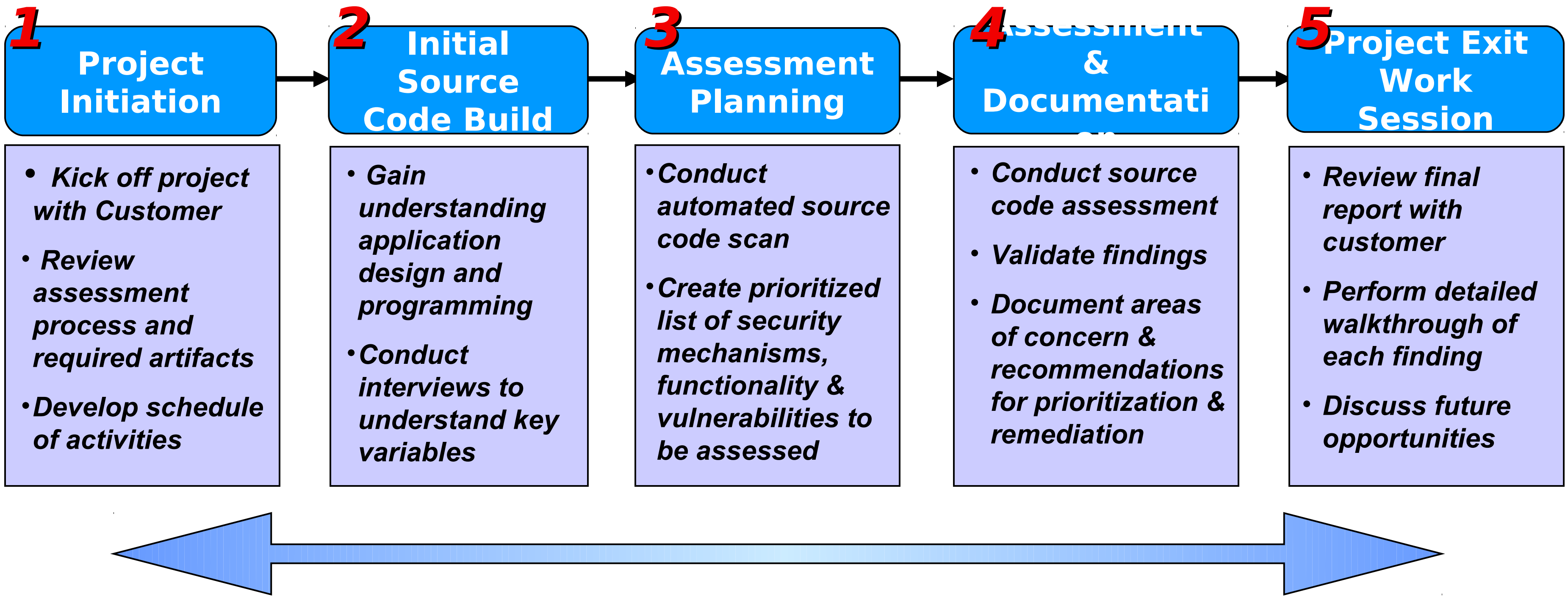
## Key Components

- Delivered by application security experts with development experience either remotely or onsite
- In-depth assessment of application vulnerabilities that may jeopardize critical sensitive data that can only be found through source code analysis
- Detailed reporting provides recommendations for prioritizing & mitigating discovered risks
- Map compliance with internal policies and industry regulations

## Benefits

- Leverages industry leading product **Rational AppScan Source Edition**
- Helps **reduce costs and risks** by proactively identifying vulnerabilities earlier in the software development lifecycle
- **Maximizes resources** by prioritizing the most critical vulnerabilities for remediation
- Supports efforts to **achieve and maintain compliance** with government and industry regulations

# Application Source Code Security Assessment- Outline of Activities



**1 Project Initiation**

- Kick off project with Customer
- Review assessment process and required artifacts
- Develop schedule of activities

**2 Initial Source Code Build**

- Gain understanding application design and programming
- Conduct interviews to understand key variables

**3 Assessment Planning**

- Conduct automated source code scan
- Create prioritized list of security mechanisms, functionality & vulnerabilities to be assessed

**4 Assessment & Documentation**

- Conduct source code assessment
- Validate findings
- Document areas of concern & recommendations for prioritization & remediation

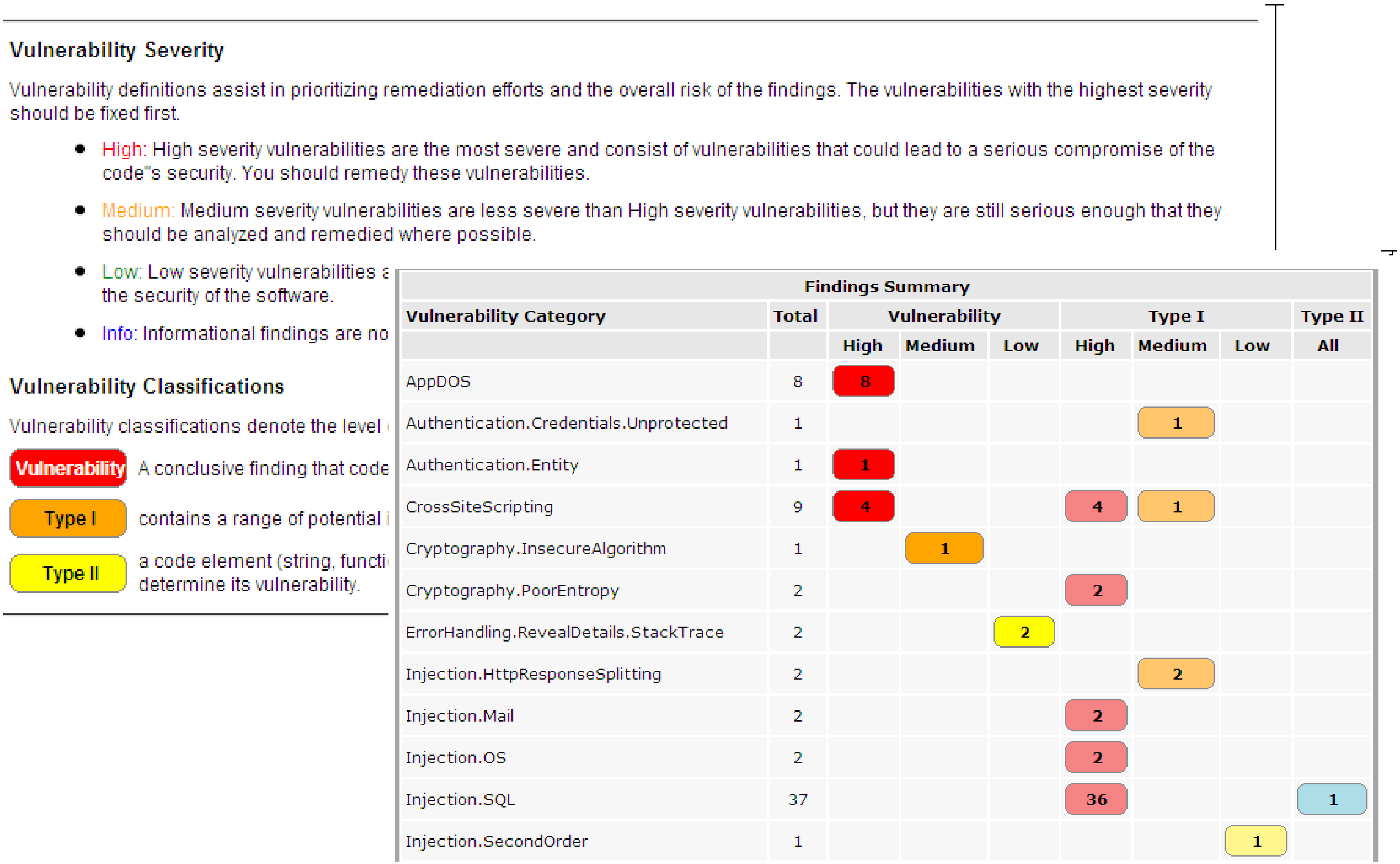
**5 Project Exit Work Session**

- Review final report with customer
- Perform detailed walkthrough of each finding
- Discuss future opportunities

**Project Planning and Management**

***Our professional services leverage market leading assessment tools to help customers reduce costs and risks by identifying and eliminating security vulnerabilities during the software development lifecycle.***

# Final Report - Score Card



### Vulnerability Severity

Vulnerability definitions assist in prioritizing remediation efforts and the overall risk of the findings. The vulnerabilities with the highest severity should be fixed first.

- High: High severity vulnerabilities are the most severe and consist of vulnerabilities that could lead to a serious compromise of the code"s security. You should remedy these vulnerabilities.
- Medium: Medium severity vulnerabilities are less severe than High severity vulnerabilities, but they are still serious enough that they should be analyzed and remedied where possible.
- Low: Low severity vulnerabilities a the security of the software.
- Info: Informational findings are no

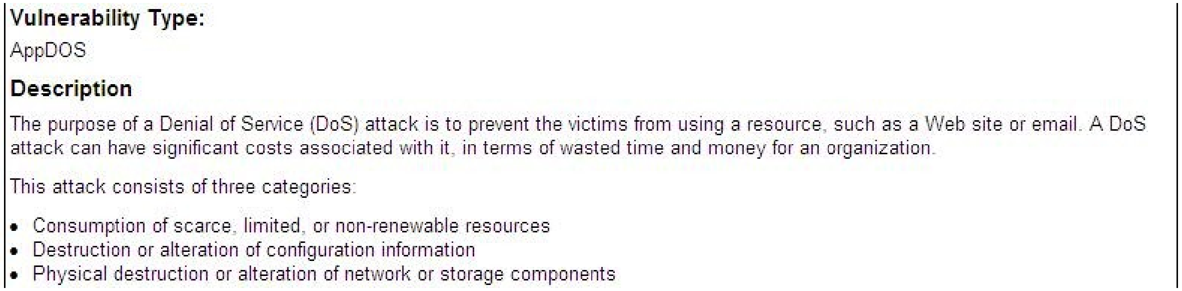
### Vulnerability Classifications

Vulnerability classifications denote the level

| | |
|---|---|
| Vulnerability | A conclusive finding that code |
| Type I | contains a range of potential i |
| Type II | a code element (string, functi determine its vulnerability. |

| Findings Summary | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Vulnerability Category | Total | Vulnerability | | | Type I | | | Type II |
| | | High | Medium | Low | High | Medium | Low | All |
| AppDOS | 8 | 8 | | | | | | |
| Authentication.Credentials.Unprotected | 1 | | | | | 1 | | |
| Authentication.Entity | 1 | 1 | | | | | | |
| CrossSiteScripting | 9 | 4 | | | 4 | 1 | | |
| Cryptography.InsecureAlgorithm | 1 | | 1 | | | | | |
| Cryptography.PoorEntropy | 2 | | | | 2 | | | |
| ErrorHandling.RevealDetails.StackTrace | 2 | | | 2 | | | | |
| Injection.HttpResponseSplitting | 2 | | | | | 2 | | |
| Injection.Mail | 2 | | | | 2 | | | |
| Injection.OS | 2 | | | | 2 | | | |
| Injection.SQL | 37 | | | | 36 | | | 1 |
| Injection.SecondOrder | 1 | | | | | | 1 | |

# Final Report – Remediation Assistance



**Vulnerability Type:**

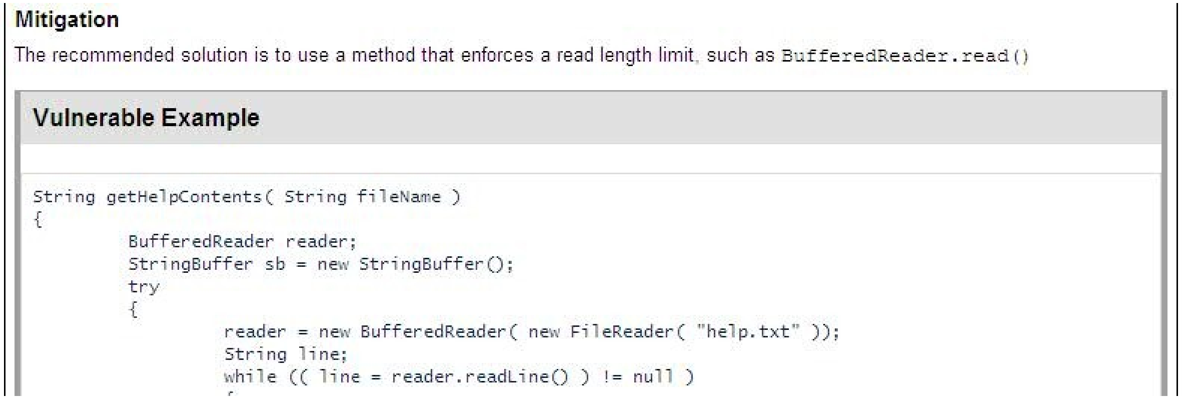
AppDOS

**Description**

The purpose of a Denial of Service (DoS) attack is to prevent the victims from using a resource, such as a Web site or email. A DoS attack can have significant costs associated with it, in terms of wasted time and money for an organization.

This attack consists of three categories:

- Consumption of scarce, limited, or non-renewable resources
- Destruction or alteration of configuration information
- Physical destruction or alteration of network or storage components

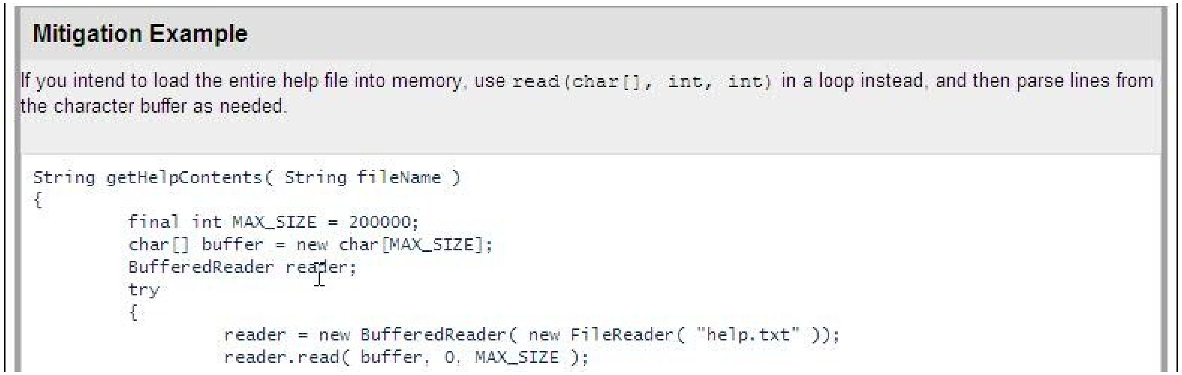


**Mitigation**

The recommended solution is to use a method that enforces a read length limit, such as `BufferedReader.read()`

**Vulnerable Example**

```
String getHelpContents( String fileName )
{
        BufferedReader reader;
        StringBuffer sb = new StringBuffer();
        try
        {
                reader = new BufferedReader( new FileReader( "help.txt" ));
                String line;
                while (( line = reader.readLine() ) != null )
```

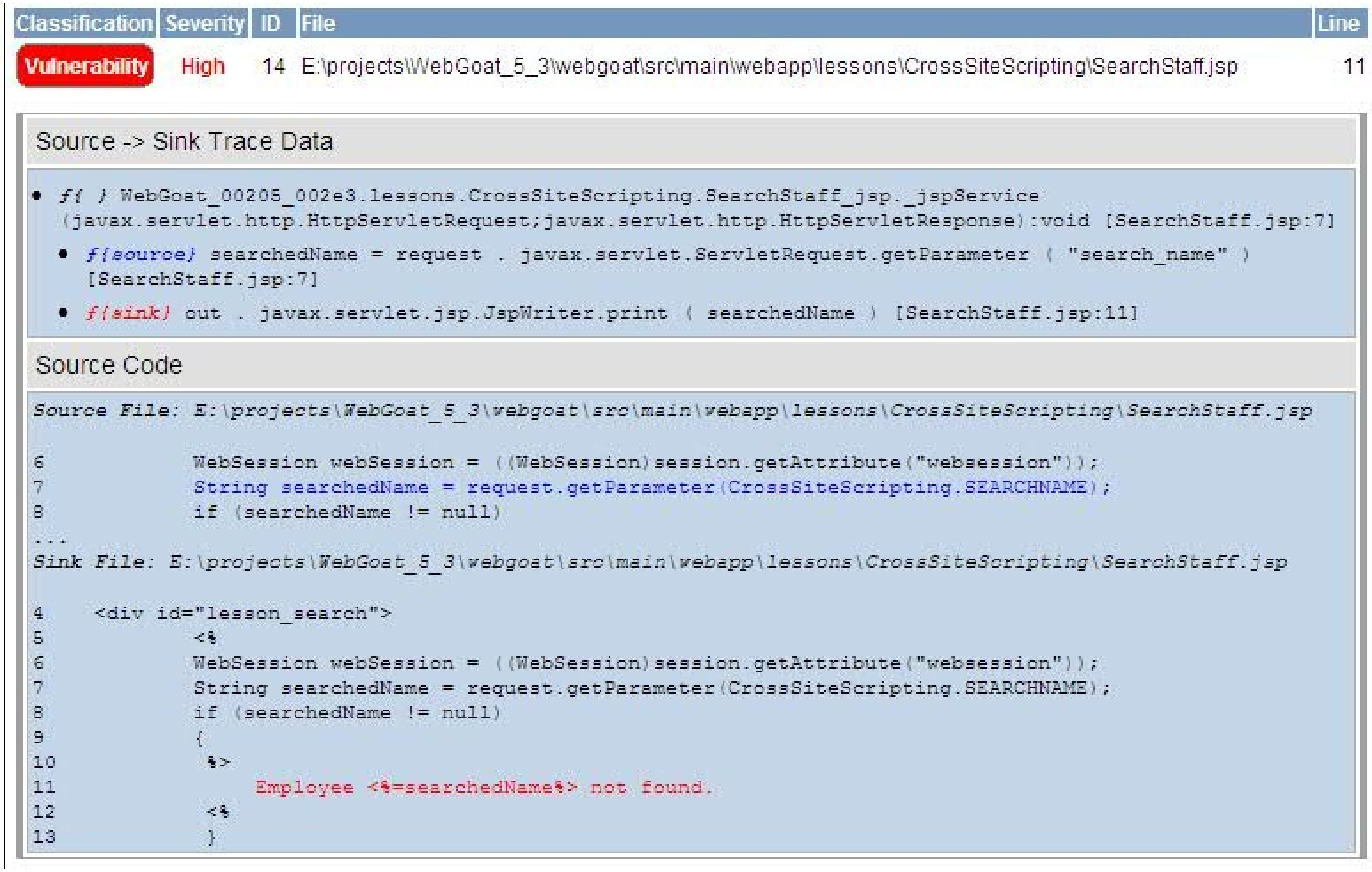


**Mitigation Example**

If you intend to load the entire help file into memory, use `read(char[], int, int)` in a loop instead, and then parse lines from the character buffer as needed.

```
String getHelpContents( String fileName )
{
        final int MAX_SIZE = 200000;
        char[] buffer = new char[MAX_SIZE];
        BufferedReader reader;
        try
        {
                reader = new BufferedReader( new FileReader( "help.txt" ));
                reader.read( buffer, 0, MAX_SIZE );
```

# Final Report – Detailed Findings



| Classification | Severity | ID | File | Line |
|---|---|---|---|---|
| Vulnerability | High | 14 | E:\projects\WebGoat_5_3\webgoat\src\main\webapp\lessons\CrossSiteScripting\SearchStaff.jsp | 11 |

Source -> Sink Trace Data

- *f{ }* WebGoat_00205_002e3.lessons.CrossSiteScripting.SearchStaff_jsp._jspService (javax.servlet.http.HttpServletRequest;javax.servlet.http.HttpServletResponse):void [SearchStaff.jsp:7]
    - *f{source}* searchedName = request . javax.servlet.ServletRequest.getParameter ( "search_name" ) [SearchStaff.jsp:7]
    - *f{sink}* out . javax.servlet.jsp.JspWriter.print ( searchedName ) [SearchStaff.jsp:11]

Source Code

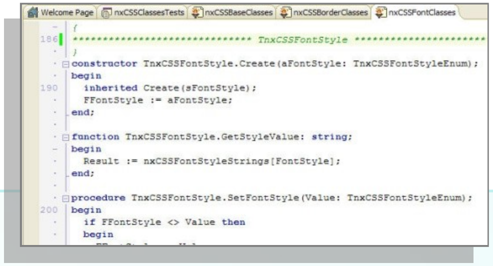
```
Source File: E:\projects\WebGoat_5_3\webgoat\src\main\webapp\lessons\CrossSiteScripting\SearchStaff.jsp

6            WebSession webSession = ((WebSession)session.getAttribute("websession"));
7            String searchedName = request.getParameter(CrossSiteScripting.SEARCHNAME);
8            if (searchedName != null)
...
Sink File: E:\projects\WebGoat_5_3\webgoat\src\main\webapp\lessons\CrossSiteScripting\SearchStaff.jsp

4    <div id="lesson_search">
5            <%
6            WebSession webSession = ((WebSession)session.getAttribute("websession"));
7            String searchedName = request.getParameter(CrossSiteScripting.SEARCHNAME);
8            if (searchedName != null)
9            {
10            %>
11                Employee <%=searchedName%> not found.
12             <%
13            }
```

# Security Testing Technologies...

## Combination Delivers a Comprehensive Solution

### Static Code Analysis = Whitebox

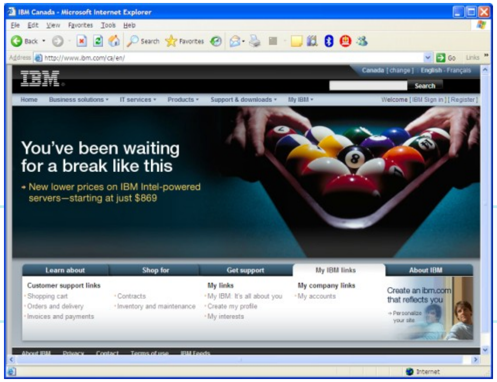
- *Assessing the source code for security issues*



```
{
****************************** TnxCSSFontStyle **********************
}
constructor TnxCSSFontStyle.Create(aFontStyle: TnxCSSFontStyleEnum);
begin
  inherited Create(sFontStyle);
  FFontStyle := aFontStyle;
end;

function TnxCSSFontStyle.GetStyleValue: string;
begin
  Result := nxCSSFontStyleStrings[FontStyle];
end;

procedure TnxCSSFontStyle.SetFontStyle(Value: TnxCSSFontStyleEnum);
begin
  if FFontStyle <> Value then
  begin
```

### Dynamic Analysis = Blackbox

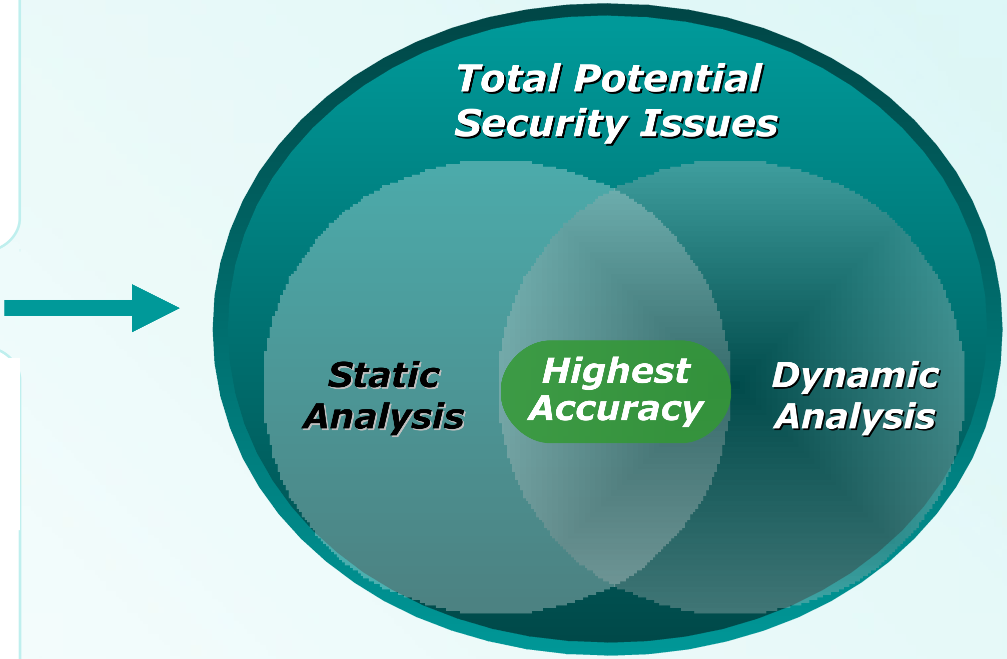
- *Performing security analysis of a compiled application*


IBM Canada - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.ibm.com/ca/en/
Go Links
IBM
Canada [ change ] English - Français
Search
Home Business solutions IT services Products Support & downloads My IBM
Welcome [ IBM Sign in ] [ Register ]
Learn about
Shop for
Get support
My IBM links
About IBM
Customer support links
Shopping cart
Orders and delivery
Invoices and payments
Contracts
Inventory and maintenance
My links
My IBM: It's all about you
Create my profile
My interests
My company links
My accounts
Personalize your site
About IBM Privacy Contact Terms of use IBM Feeds
Internet

You've been waiting for a break like this

→ New lower prices on IBM Intel-powered servers—starting at just $869

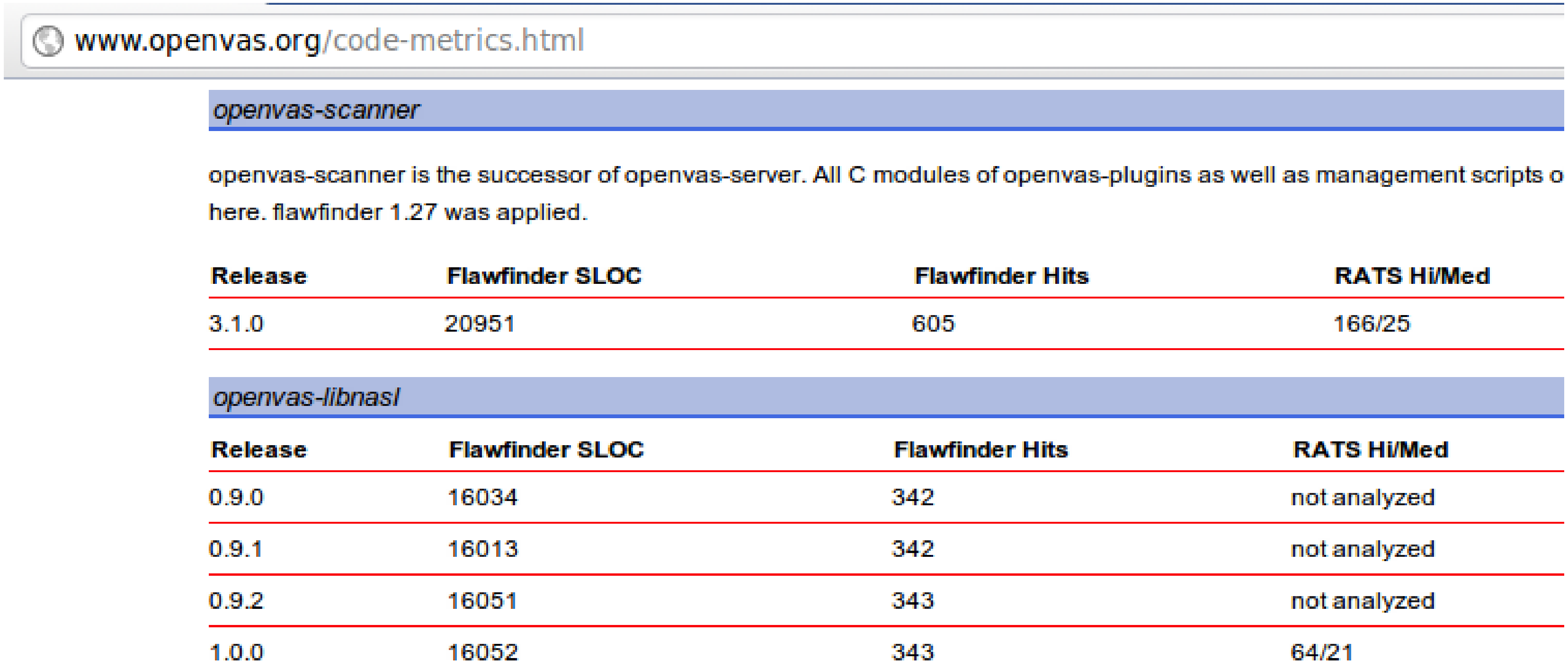
Create an ibm.com that reflects you



# Nekoliko preporuka

- Ako vodite programere
  - Uključiti sigurnost u kompletan proces razvoja
    - Dizajn!
  - Uključite testiranje u kompletan proces razvoja
- Ako ste programer
  - Ne vjeruj
    - Korisničkom unosu
  - Radi provjeru
    - Svakog ulaznog podatka u aplikaciju
  - Razmišljaj kao napadač
- Ako ste naručitelj posla
  - Uključite odgovornost za sigurnost u ugovor
    - Odgovornost je naručitelja ili izvođača?
  - Pitajte za implementirane sigurnosne zaštite
    - I Tražite ih!
    - Prihvatljiva zaštita
  - Razvijte nefunkcionalne zahtjeve za aplikaciju
    - Naravno, prvo funkcionalne ;)
- Testirajte aplikaciju
  - Različiti oblici testiranja
    - Penetracijski testovi, Pregled izvornog koda, ....
    - ...

# Example



www.openvas.org/code-metrics.html

*openvas-scanner*

openvas-scanner is the successor of openvas-server. All C modules of openvas-plugins as well as management scripts o
here. flawfinder 1.27 was applied.

| Release | Flawfinder SLOC | Flawfinder Hits | RATS Hi/Med |
|---|---|---|---|
| 3.1.0 | 20951 | 605 | 166/25 |

*openvas-libnasl*

| Release | Flawfinder SLOC | Flawfinder Hits | RATS Hi/Med |
|---|---|---|---|
| 0.9.0 | 16034 | 342 | not analyzed |
| 0.9.1 | 16013 | 342 | not analyzed |
| 0.9.2 | 16051 | 343 | not analyzed |
| 1.0.0 | 16052 | 343 | 64/21 |

# Pitanja i odgovori

<vlatko.kosturjak@hr.ibm.com>