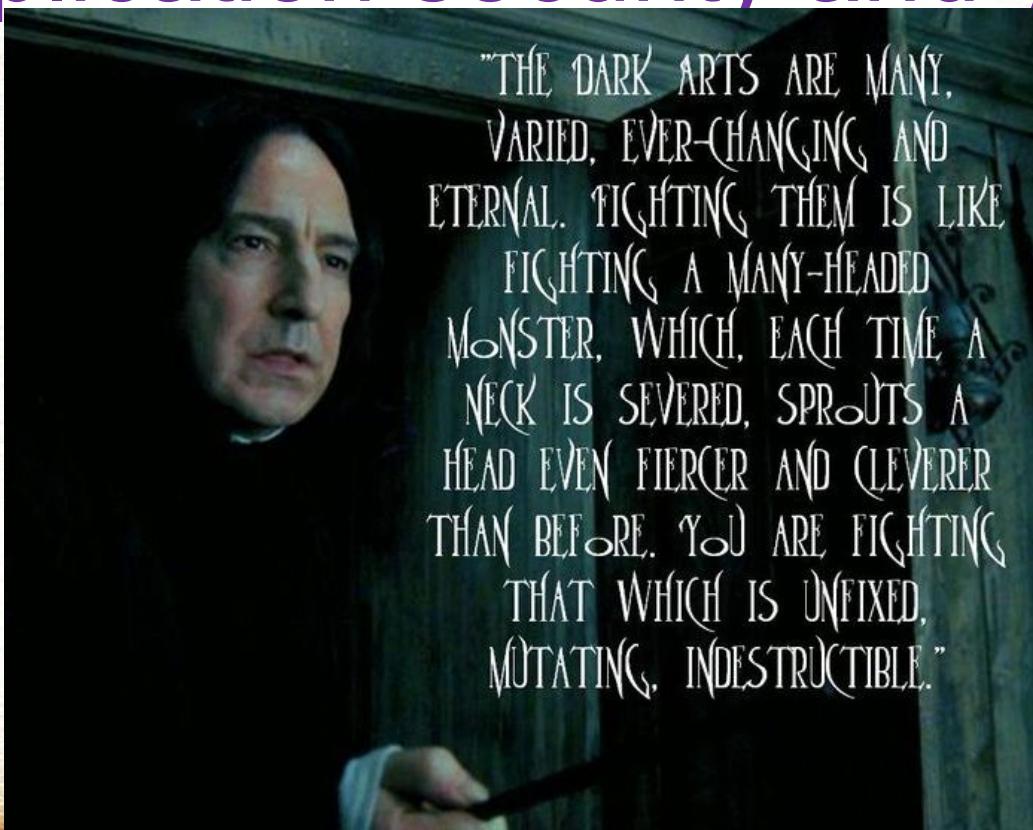


Defense Against The Dark Arts: Application security and you



<https://uk.pinterest.com/pin/335518240961936792/>

Joe Kuemerle

- Application security engineer & developer in a broad range of technologies
- Specialize in application and data security, coding best practices and regulatory compliance
- Presenter at community, regional and national events.



S. IT Budget Devoted to Securing Software

\$86 billion

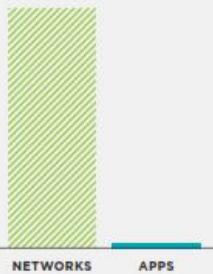
for securing networks
and endpoints

BUT ONLY

\$700 million

for securing applications

- GARTNER 2015



1% of security
spend is focused on
application security.

- GARTNER

About 70% of all
applications have at least
one vulnerability classified
as one of the top 10 web
vulnerability types.

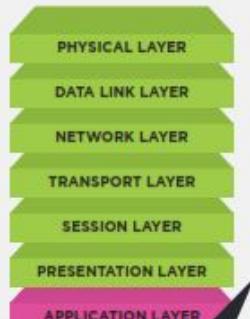
- VERACODE'S STATE OF SOFTWARE
SECURITY REPORT (VOLUME 6)



Where Your Risk Is

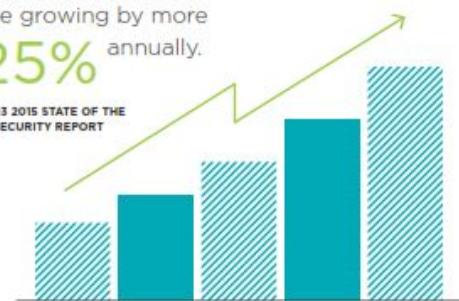
Yet... Over 70%
of security vulnerabilities
exist at the application
layer, not the network
or system layer.

- GARTNER



Attacks at the application
layer are growing by more
than 25% annually.

- AKAMAI'S Q3 2015 STATE OF THE
INTERNET SECURITY REPORT



Web application
attacks remain the most
frequent incident pattern
in confirmed breaches
and accounted for up to
40% of breaches.

- 2016 VERIZON DATA BREACH
INVESTIGATIONS REPORT

<https://www.veracode.com/resources>



<http://cdn.quotesgram.com/small/79/88/1743520670-top-10.jpg>

https://www.owasp.org/index.php/Top_10-2017_Top_10

Injection

Security Misconfiguration

Broken Authentication

Cross Site Scripting (XSS)

Sensitive Data Exposure

Insecure Deserialization

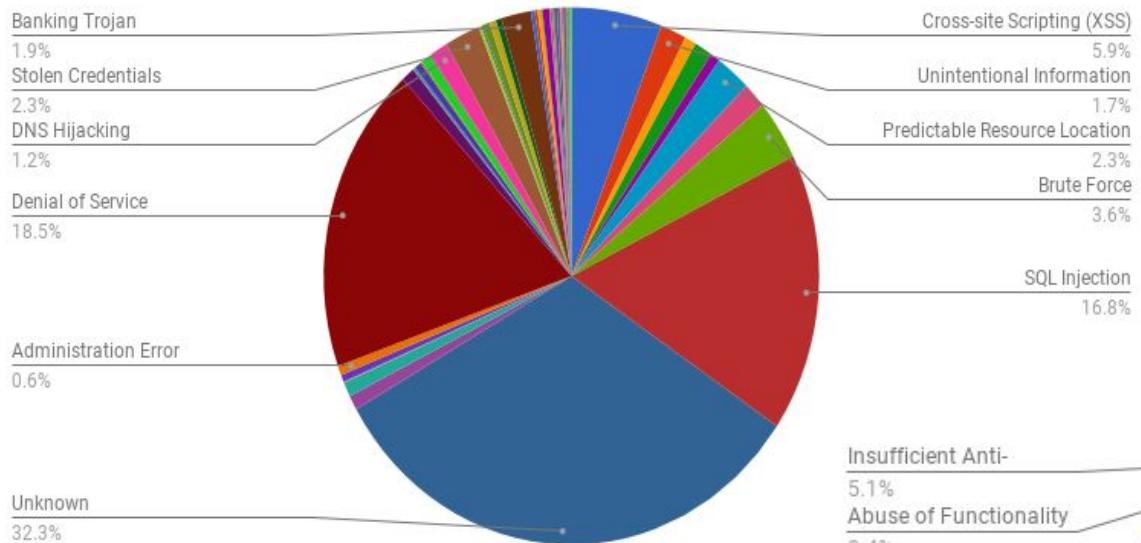
XML External Entities (XXE)

Components With Known Vulnerabilities

Broken Access Control

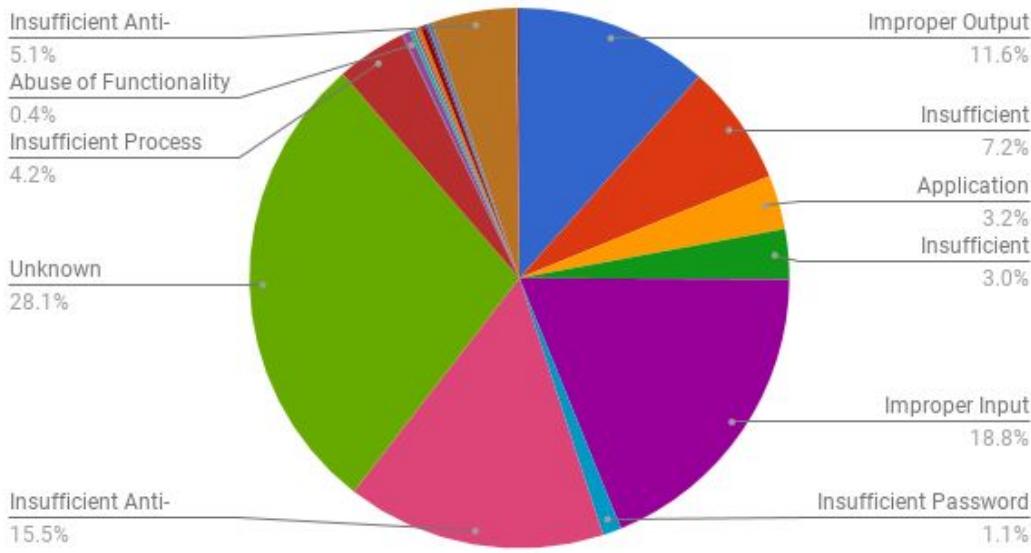
Insufficient Logging and Monitoring

Attack Methods



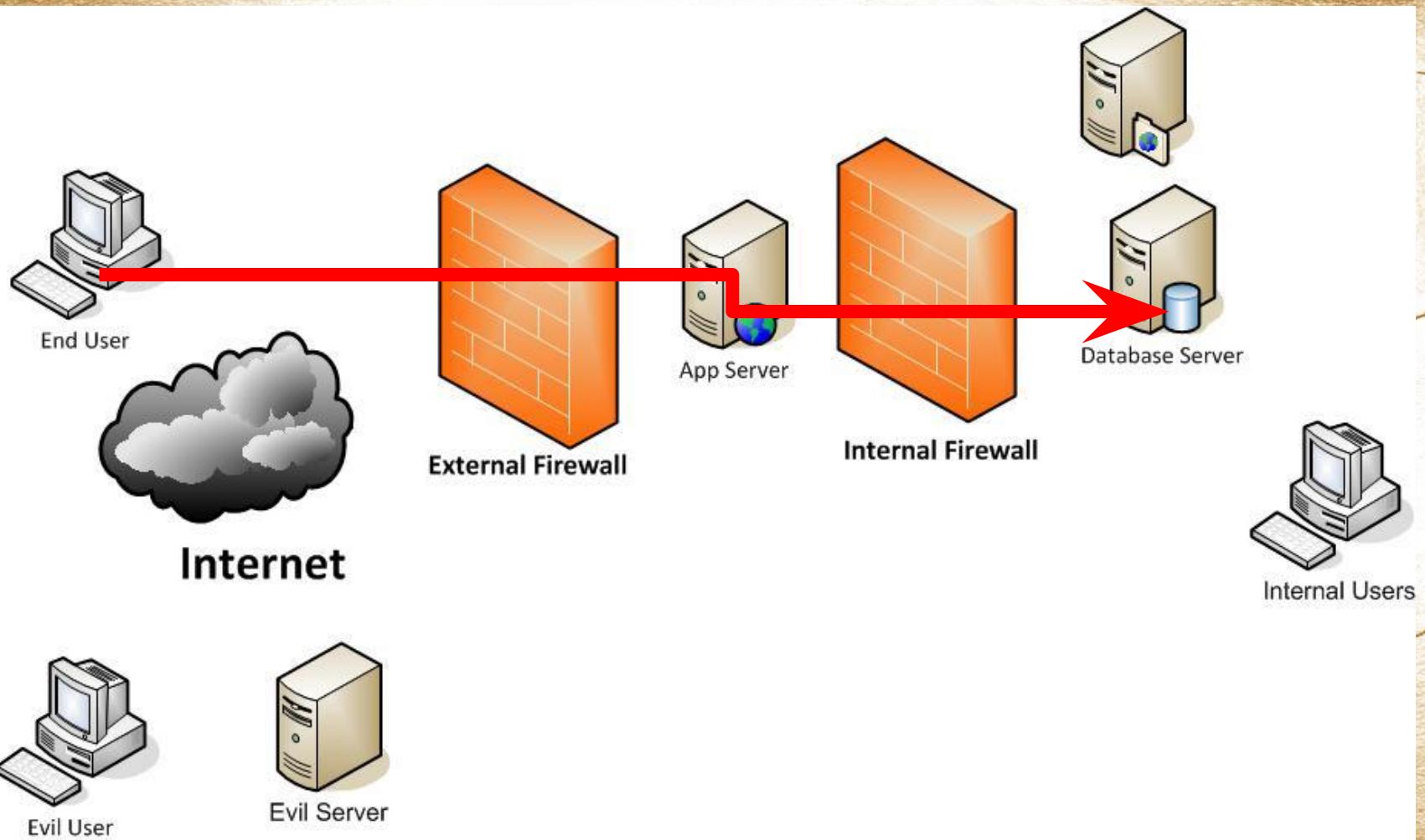
Source:
Web Hacking Incident Database
<http://tinyurl.com/AppAttackStats>

Application Weaknesses



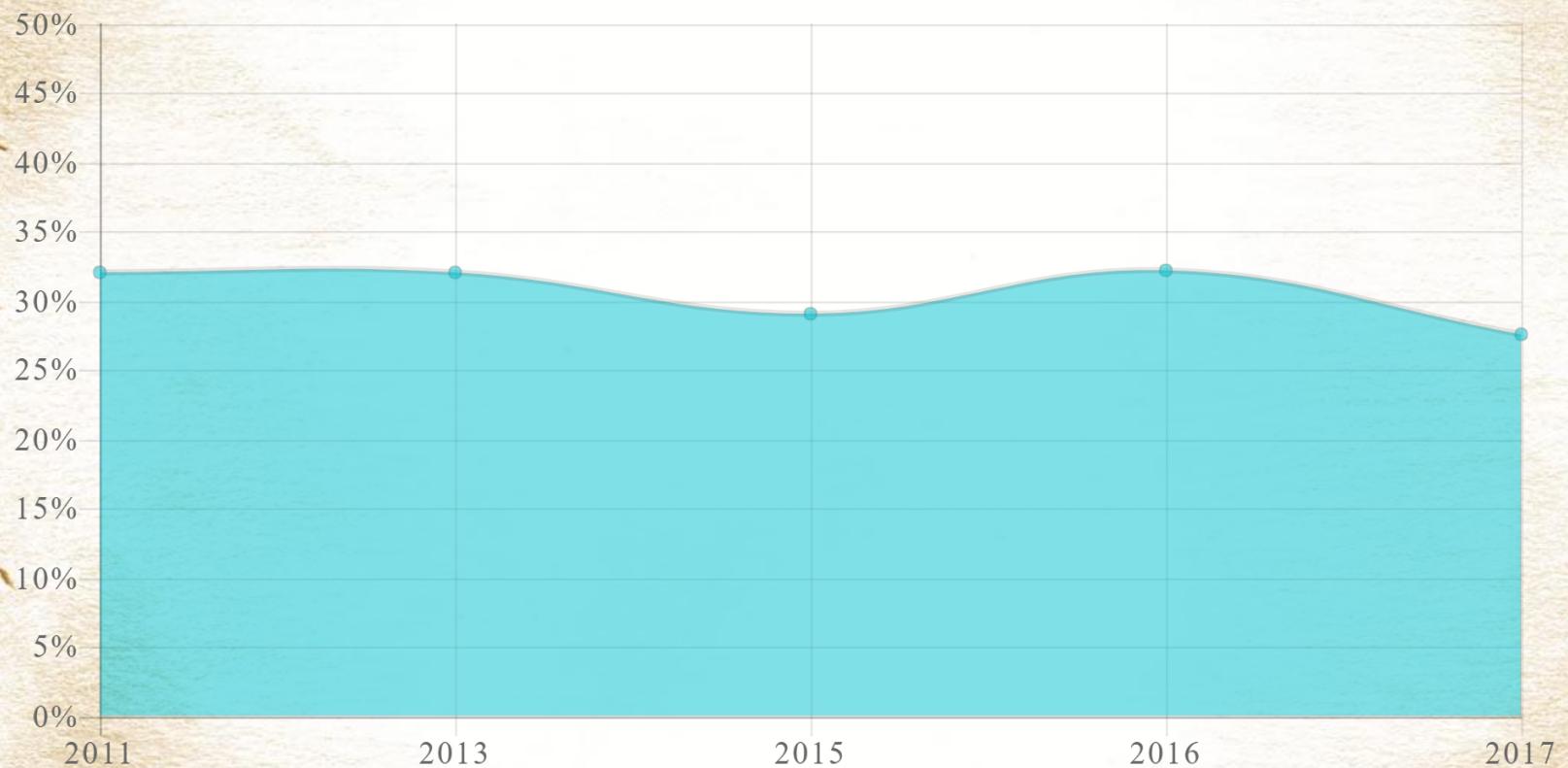
<https://www.flickr.com/photos/colmmcsky/6300431678/>





SQL INJECTION TREND

Percentage of Applications Affected



<http://www.veracode.com/resources/state-of-software-security>

Company	Date	Results	Reference
IIT-Delhi	2018-03	potential access to student data	Kerala youth picks security flaw in IIT-Delhis web server
BSNL	2018-03	47,000 employees data taken by hacker to prove problem. Second time for this site in 2 years. Fixed.	Security researcher hacks BSNL intranet, leaks details of 47,000 employees
Telangana NREGA portal (India)	2018-02	hack exposed data to show security flaw	Hacker exposes major security flaw in Telangana governments NREGA website
Joomla	2018-02	Joomla CMS sqli bugs (patch available)	Joomla 3.2.4 release addresses three XSS and SQL Injection vulnerabilities
Zoho	2018-01	Helpdesk software vulnerable to blind SQLi	Multiple critical flaws found in Zoho MANAGEENGINE
Hetzner South Africa	2017-11	Over 40,000 customer details including bank accounts leaked.	Hetzner hack - top south african web host hit
Wordpress	2017-10	SQLi vulnerability in plugins - patched in v4.8.3	If your websites use WordPress, put down that coffee and upgrade to 4.8.3
GoDaddy	2017-10	Researcher showed site security tool is easily bypassed to allow sqli	This bug let a researcher bypass GoDaddy's site security tool
Inmarsat	2017-10	AmosConnect satellite communications for ships is vulnerable.	mosConnect: Maritime Communications Security Has Its Flaws
Equifax	(long term)	Personal data for over 145 million people compromised	Equifax was warned
Jigsaw Holding in South Africa	2017-10	75m database records available for download	Revealed the real source of SAs massive data breach
Catholic United Financial	2017-10	personal info for over 130k users.	Data breach at Arden Hills-based Catholic financial services provider affects nearly 130k accounts
BPC Banking - SmartVista software	2017-10	numerous vulnerable ecommerce sites	Vendor BCP Baking Silent on Patching SQL Injection in SmartVista Ecommerce Software
EMC	2017-07	multiple vulnerabilities found, some fixed.	EMC products hit by multiple vulnerabilities including SQL injection
Wordpress Statistics plugin	2017-07	vulnerability in wordpress plugin	SQL injection vulnerability found in popular WordPress plugin, again
Siemens	2017-06	vulnerability in AMT industrial products patched after CERT warning	Siemens patches critical Intel AMT flaw in industrial products
Illinois State Board of Elections	2017-06	voter registration system breached - reported in 2017-06 but occurred in 2016	Illinois chapter in the Russian hacking saga
Peplink	2017-06	vulnerable routers - now patched	Peplink patches SD-WAN routers



<http://motherboard-images.vice.com/content-images/article/28241/1448640227075896.jpg>

```
soap:Envelope [ xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/ xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance xmlns:xsd=http://www.w3.org/2001/XMLSchema]
  soap:Body
    loginResponse [ xmlns=http://vttech_dev/ ]
      loginResult
        success
          false
        message
          No member record found.select * from member WHERE login_name='LittleMary' and password='Password'
        err_string
          alertInvalidLogin
        is_parent
          false
        is_avatar_created
          false
```

NoSQLMap-Automated NoSQL Database Pwnage

[HOME](#) [A NOTE...](#)

[An Important Message Regarding "The NoSQL Exploitation Framework"](#)

[Github repository](#)

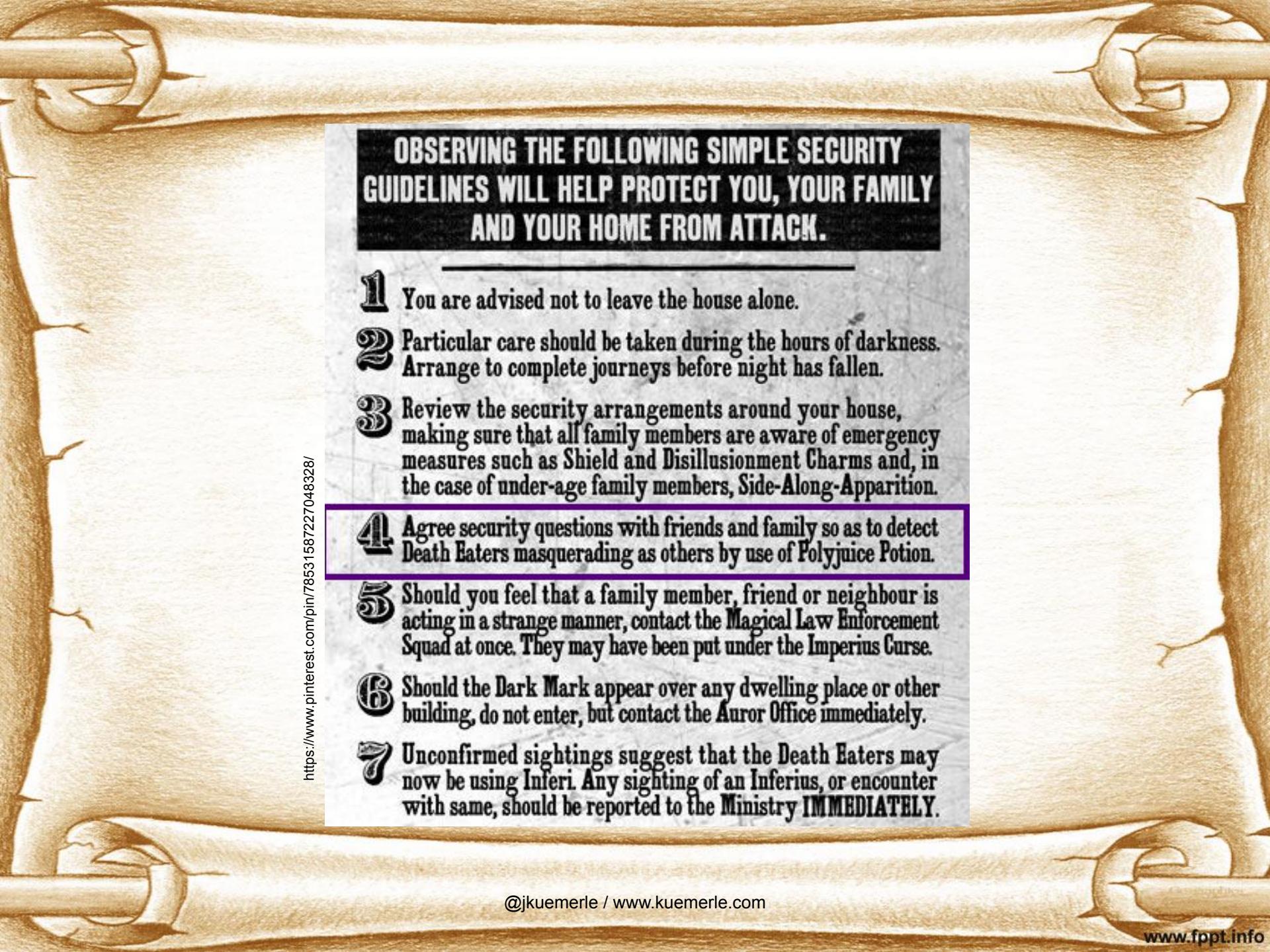
[Defcon 22 Wall of Sheep Presentation Slides](#)

[Demo Videos](#)

What is NoSQLMap?

NoSQLMap is an open source Python tool designed to audit for as well as automate injection attacks and exploit default configuration weaknesses in NoSQL databases, as well as web applications using NoSQL in order to disclose data from the database. It is named as a tribute to Bernardo Damele and Miroslav's Stampar's popular SQL injection tool SQLmap, and its concepts are based on and extensions of Ming Chow's excellent presentation at Defcon 21, "Abusing NoSQL Databases". Presently the tool's exploits are focused around MongoDB, but additional support for other NoSQL based platforms such as CouchDB, Redis, and Cassandra are planned in future releases. The current project goals are to provide a penetration testing tool to simplify attacks on MongoDB servers and web applications as well as proof of concept attacks to debunk the premise that NoSQL applications are impervious to SQL injection.

```
db.myCollection.find( { $where: function() { return obj.credits - obj.debits < 0; } } );  
db.myCollection.find( { active: true, $where: function() { return obj.credits -  
obj.debits < $userInput; } } );
```

OBSERVING THE FOLLOWING SIMPLE SECURITY GUIDELINES WILL HELP PROTECT YOU, YOUR FAMILY AND YOUR HOME FROM ATTACK.

- 1** You are advised not to leave the house alone.
- 2** Particular care should be taken during the hours of darkness.
Arrange to complete journeys before night has fallen.
- 3** Review the security arrangements around your house,
making sure that all family members are aware of emergency
measures such as Shield and Disillusionment Charms and, in
the case of under-age family members, Side-Along-Apparition.
- 4** Agree security questions with friends and family so as to detect
Death Eaters masquerading as others by use of Polyjuice Potion.
- 5** Should you feel that a family member, friend or neighbour is
acting in a strange manner, contact the Magical Law Enforcement
Squad at once. They may have been put under the Imperius Curse.
- 6** Should the Dark Mark appear over any dwelling place or other
building, do not enter, but contact the Auror Office immediately.
- 7** Unconfirmed sightings suggest that the Death Eaters may
now be using Inferi. Any sighting of an Inferius, or encounter
with same, should be reported to the Ministry IMMEDIATELY.

<https://www.flickr.com/photos/foilman/27016860369/>



@jkuemerle / www.kuemerle.com

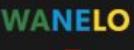
www.fppt.info

412 pwned websites 8,513,925,254 pwned accounts 103,309 pastes 123,092,693 paste accounts

Largest breaches

	772,904,991	Collection #1 accounts
	763,117,241	Verifications.io accounts
	711,477,622	Onliner Spambot accounts
	593,427,119	Exploit.In accounts
	457,962,538	Anti Public Combo List accounts
	393,430,309	River City Media Spam List accounts
	359,420,698	MySpace accounts
	234,842,089	NetEase accounts
	164,611,595	LinkedIn accounts
	161,749,950	Dubsmash accounts

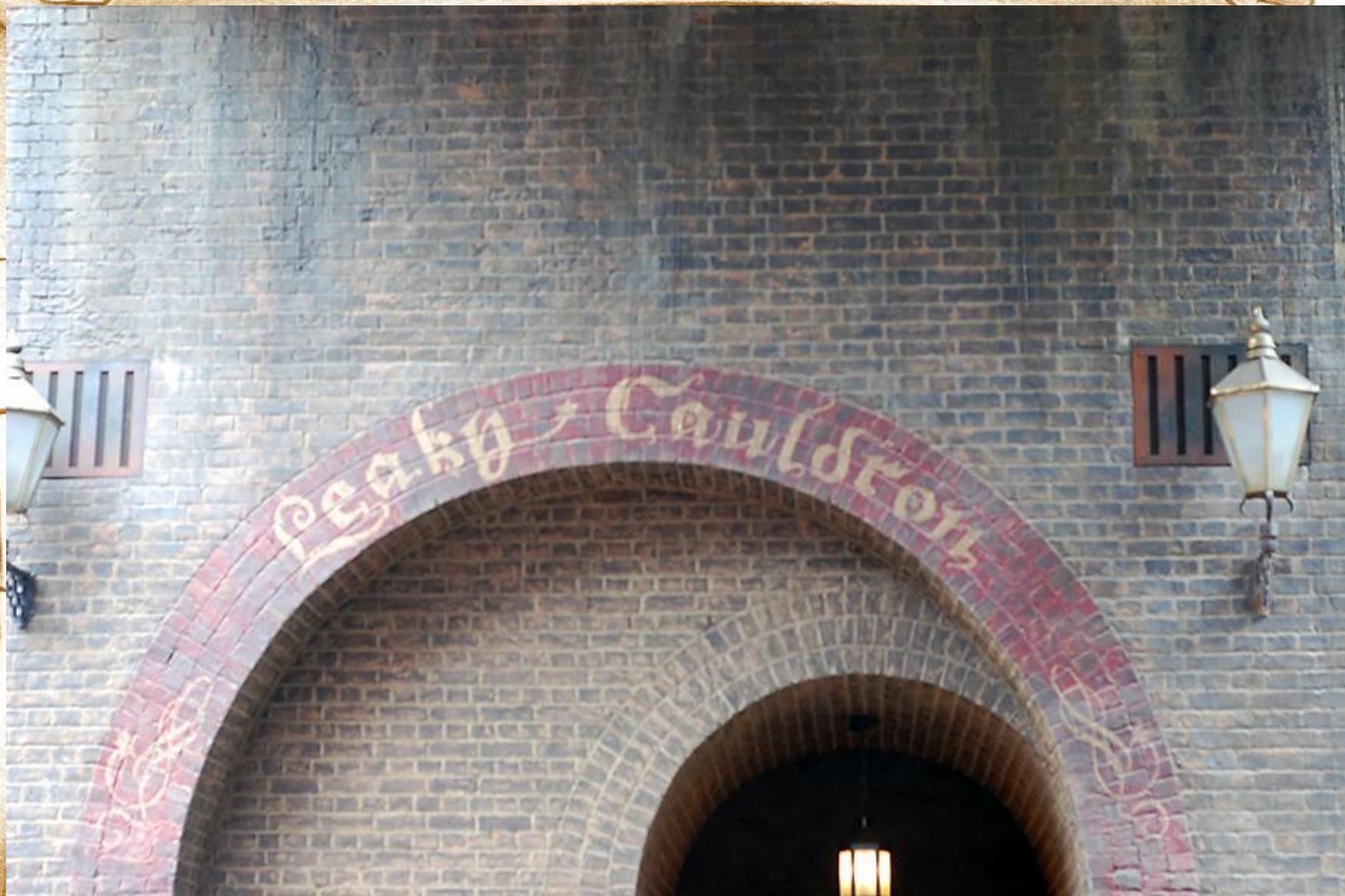
Recently added breaches

	6,002,694	ToonDoo accounts
	686,899	Vedantu accounts
	290,955	Hookers.nl accounts
	71,407	Zooville accounts
	988,230	StreetEasy accounts
	780,073	Sephora accounts
	23,165,793	Wanelo accounts
	15,453,048	Lumin PDF accounts
	4,606	KiwiFarms accounts
	396,533	Minehut accounts



<https://www.flickr.com/photos/bryanaalexander/17611523750/>

@jkueemerle / www.kueemerle.com



<https://www.flickr.com/photos/yaketyyaketyak/7325939128/>





<https://graph.facebook.com/1138975844>

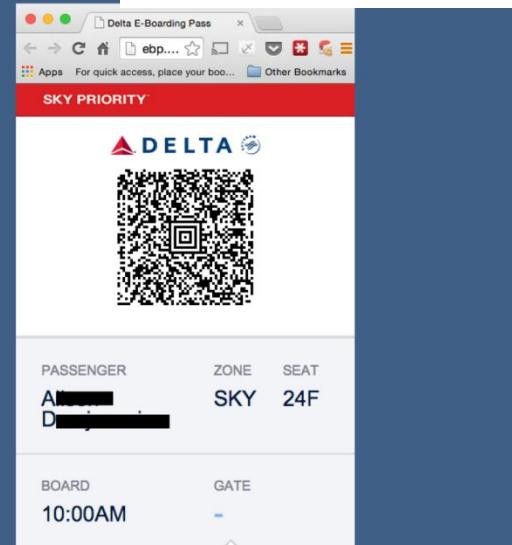
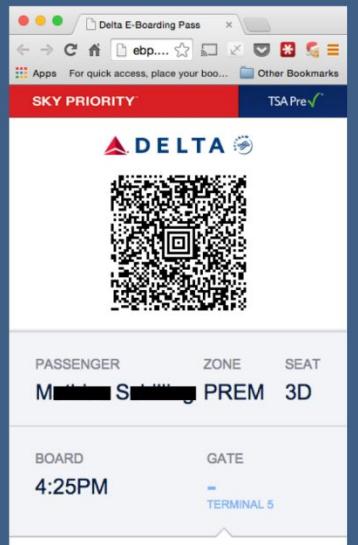
That's me, obviously.

```
{  
  "id": "1138975844",  
  "name": "Bill Sempf",  
  "first_name": "Bill",  
  "last_name": "Sempf",  
  "username": "billsempf",  
  "gender": "male",  
  "locale": "en_US"  
}
```

<https://graph.facebook.com/1138975845>

That could get interesting.

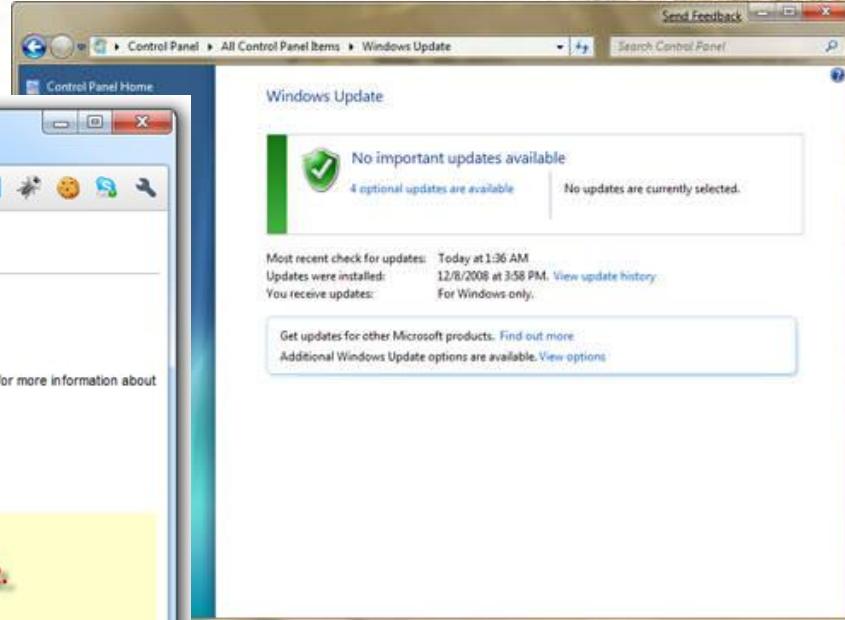
```
1 {  
2   "id": "1138975845",  
3   "name": "Mary Loaiza",  
4   "first_name": "Mary",  
5   "last_name": "Loaiza",  
6   "link": "http://www.facebook.com/people/Mary-Loaiza/1138975845",  
7   "gender": "female",  
8   "locale": "es_LA"  
9 }
```



<http://www.itnews.com.au/News/398892,delta-site-flaw-lets-passengers-access-other-boarding-passes.aspx>



<https://www.flickr.com/photos/saeru/872702709/>



The screenshot shows the Windows Control Panel with the path "Control Panel > All Control Panel Items > Windows Update". A "Send Feedback" button is at the top right. Below it is a "Control Panel Home" link. The main area displays a green shield icon with a checkmark, indicating "No important updates available". It also shows "4 optional updates are available" and "No updates are currently selected". A message states: "Most recent check for updates: Today at 1:36 AM. Updates were installed: 12/8/2008 at 3:58 PM. View update history. You receive updates: For Windows only." At the bottom, there are links to "Get updates for other Microsoft products: Find out more" and "Additional Windows Update options are available: View options".

Server Error in '/' Application.

Value cannot be null. Parameter name: String 1.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.ArgumentNullException: Value cannot be null.
Parameter name: String

Source Error:

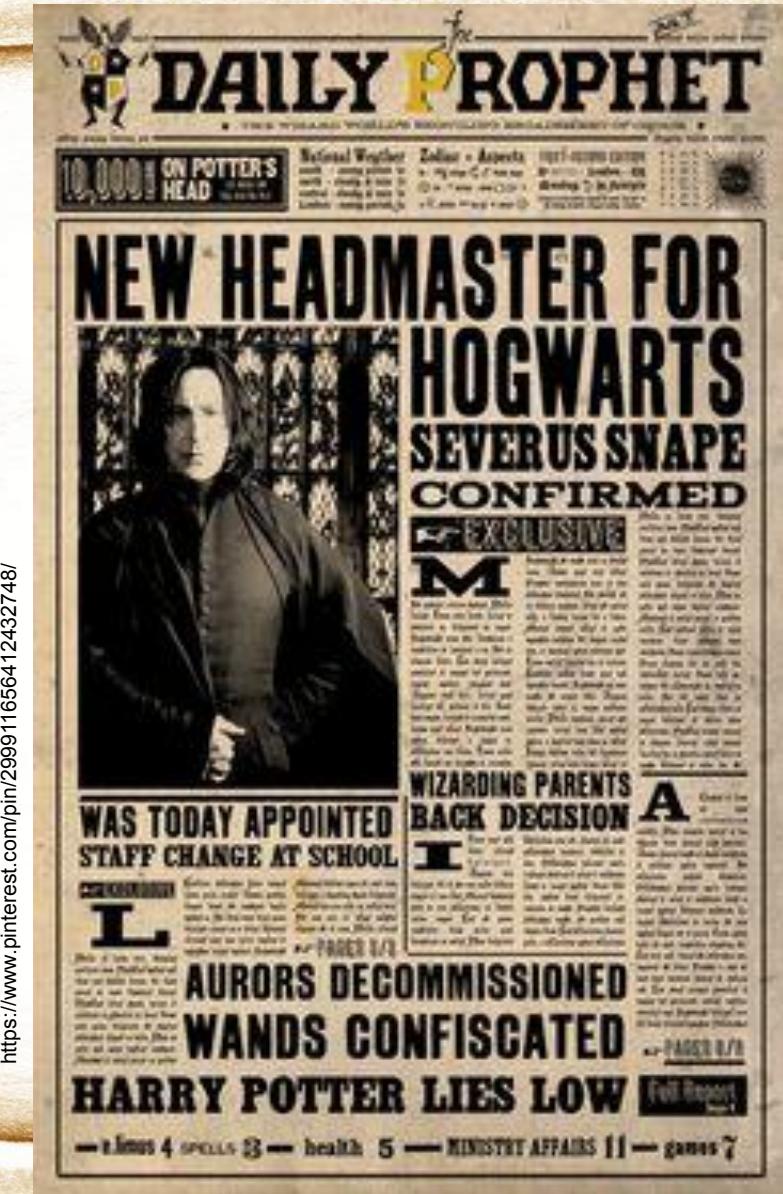
```
Line 12:     protected void Page_Load(object sender, EventArgs e)
Line 13:     {
Line 14:         int id = int.Parse(Request.QueryString["Id"]); 2.
Line 15:         // The admin ID has special rights over the system.
Line 16:         var hasAdminRights = id == 837235272; 3.
```

Source File: c:\Projects\VulnerableApplication\Web\Default.aspx.cs Line: 14 4.

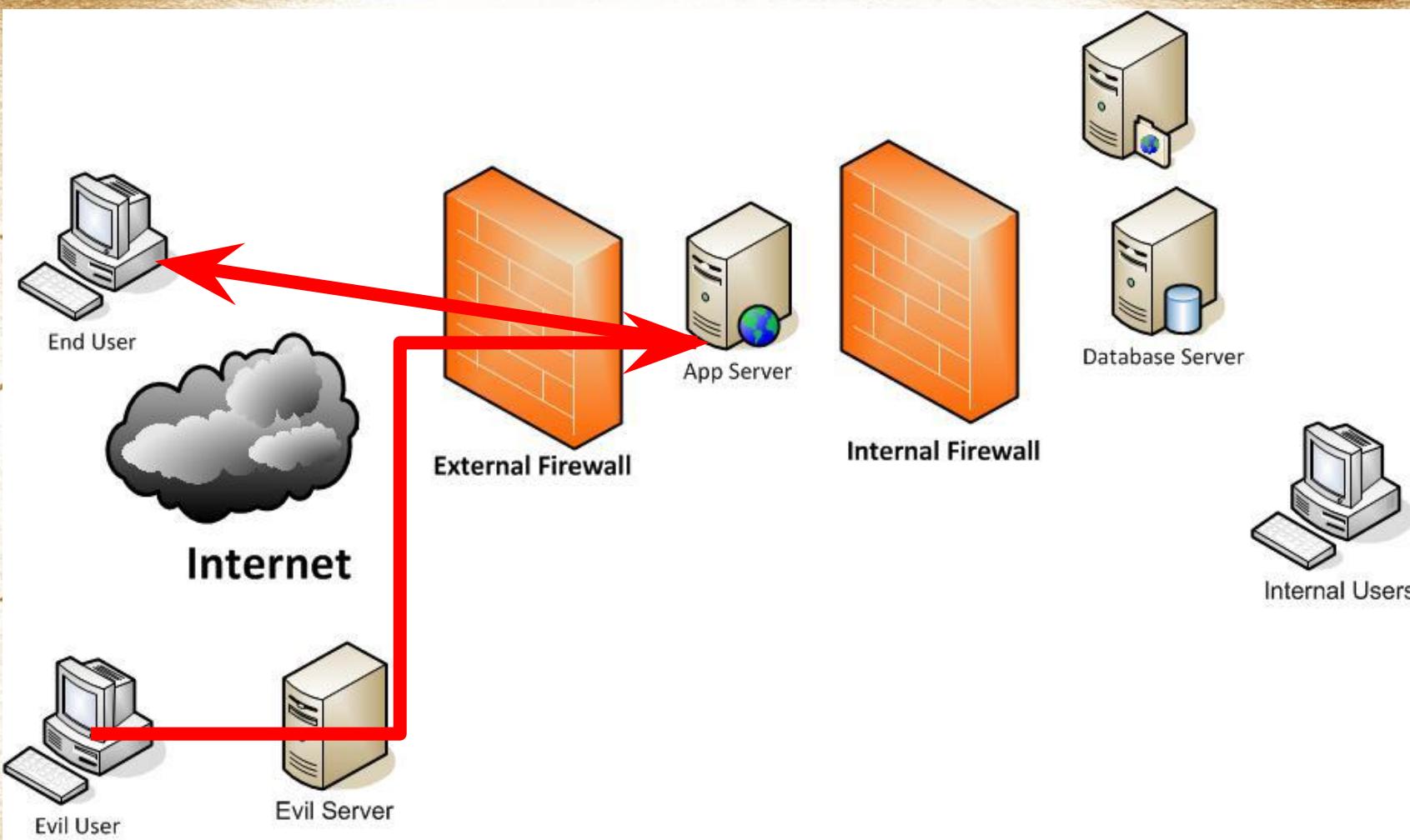
Stack Trace:

```
[ArgumentNullException: Value cannot be null.
Parameter name: String]
System.Number.StringToNumber(String str, NumberStyles options, NumberBuffer& number, NumberFormatInfo info) +224
System.Number.ParseInt32(String s, NumberStyles style, NumberFormatInfo info) +224
Web._Default.Page_Load(Object sender, EventArgs e) in c:\Projects\VulnerableApplication\Web\Default.aspx.cs:14
System.Web.Util.CalliHelper.EventArgFunctionCaller(IntPtr fp, Object o, Object t, EventArgs e) +71
System.Web.UI.Control.LoadRecursive() +71
System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includesPostBackData) +244
```

Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.0.30319.1 6.



<https://www.pinterest.com/pin/299911656412432748/>



XSS Filter Evasion Cheat Sheet

Contents [PDF]

- 1 Introduction
- 2 Tags
 - 2.1 XSS Locator
 - 2.2 XSS iFrame
 - 2.3 XSS in Evasion
 - 2.4 Image XSS using the JavaScript directive
 - 2.5 No quotes and no semicolon
 - 2.6 Double quote XSS attack vector
 - 2.7 HTML entities
 - 2.8 Oracle accent obfuscation
 - 2.9 Microsoft IIS tags
 - 2.10 Obfuscation
 - 2.11 UTF-8 Unicode encoding
 - 2.12 Long UTF-8 Unicode encoding without semicolons
 - 2.13 Head encoding without semicolons
 - 2.14 Script tags
 - 2.15 Embedded newline to break up XSS
 - 2.16 Embedded carriage return to break up XSS
 - 2.17 Not the first character
 - 2.18 Spaces and meta char before the JavaScript in Images for XSS
 - 2.19 Nonhexadecimal digit XSS
 - 2.20 JavaScript in attributes
 - 2.21 XSS in attributes
 - 2.22 No closing script tags
 - 2.23 Protocol resolution in script tags
 - 2.24 Half open HTML/JavaScript XSS vector
 - 2.25 Single quote single brackets
 - 2.26 Escaped JavaScript escapes
 - 2.27 End tag tag
 - 2.28 INPUT tag
 - 2.29 SELECT tag
 - 2.30 IFRAME
 - 2.31 IFRAME
 - 2.32 Library tag
 - 2.33 XSS in an image
 - 2.34 Livescript (older versions of Netscape only)
 - 2.35 BODY tag
 - 2.36 BGSOUND
 - 2.37 BGSOUND
 - 2.38 & JavaScript includes
 - 2.39 STYLE tag
 - 2.40 Remote style sheet
 - 2.41 Remote style sheet part 2
 - 2.42 Remote style sheet part 3
 - 2.43 Remote style sheet part 4
 - 2.44 Using a comment as a placeholder for XSS
 - 2.45 NOVIELE attribute using a comment to break up expression
 - 2.46 NOVIELE with expression
 - 2.47 NOVIELE tag (older versions of Netscape only)
 - 2.48 NOVIELE tag using background image
 - 2.49 NOVIELE tag using background
 - 2.50 Anonymous HTML with NOVIELE attribute
 - 2.51 NOVIELE file
 - 2.52 URL-ASCII encoding
 - 2.53 META
 - 2.53.1 META using data
 - 2.53.2 META with additional URL parameter
 - 2.54 FRAME
 - 2.55 TABLE
 - 2.56 TD
 - 2.57 DIV
 - 2.57.1 DIV background-image
 - 2.57.2 DIV background-image with unquoted XSS exploit
 - 2.57.3 DIV background-image plus extra characters
 - 2.57.4 DIV expression
 - 2.58 Document/hidden block
 - 2.59 Flash
 - 2.60 OBJECT tag
 - 2.61 Using an EMBED tag you can embed a Flash movie that contains XSS
 - 2.62 You can EMBED BVD which can control your XSS vector
 - 2.63 URL with a space that will obfuscate your XSS vector
 - 2.64 XML data island with CDATA obfuscation
 - 2.65 Locally hosted XML with embedded JavaScript that is generated using an XML data island
 - 2.66 XML-TIME in XML
 - 2.67 Against you can only fit in a few characters and it filters against ">"
 - 2.68 BBP (Beamer BBP includes)
 - 2.69 PHP
 - 2.70 URL-ASCII encoding
 - 2.70.1 URL Encoded commands part II
 - 2.71 Cookie manipulation
 - 2.72 UTF-7 encoding
 - 2.73 URL-8 encoding, quote encapsulation
 - 2.74 URL string evasion
 - 2.74.1 IP versus hostname
 - 2.74.2 URL encoding
 - 2.74.3 URL encoding
 - 2.74.4 Hex encoding
 - 2.74.5 Octal encoding
 - 2.74.6 Base64 encoding

https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet

<https://www.flickr.com/photos/yaketyakyak/7306685420/>



@jkueemerle / www.kuemerle.com

Quarterly Trend for Cross-Site Scripting (XSS) Prevalence (Percentage of Affected Web Applications)

p-value = 0.441

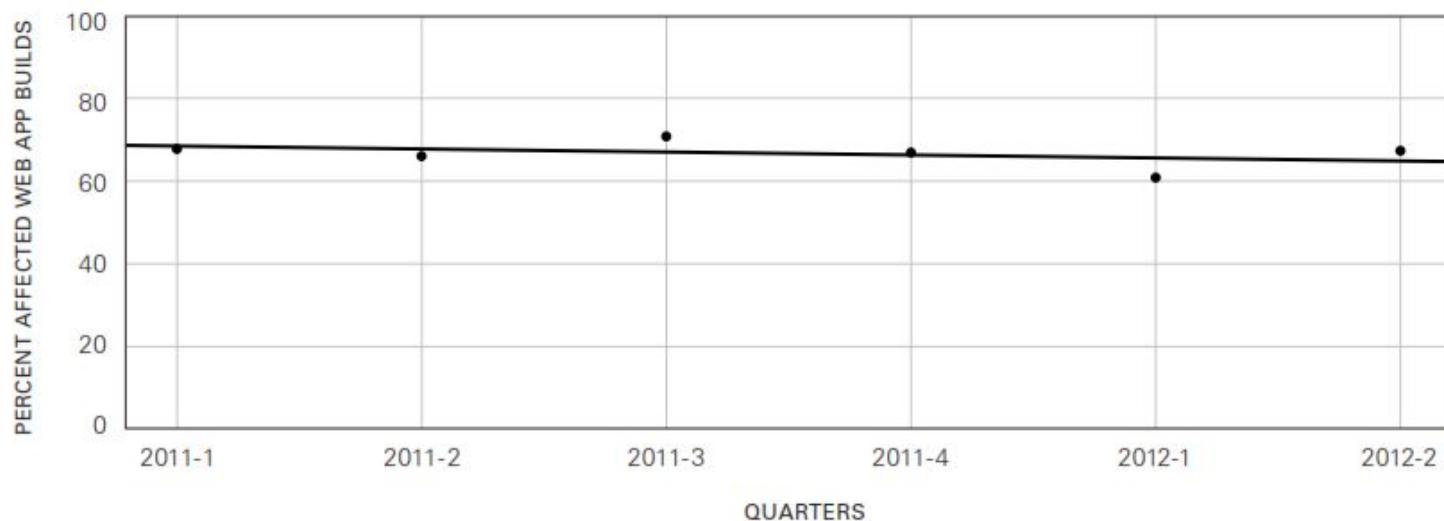


Figure 27: Quarterly Trend for Cross-Site Scripting (XSS) Prevalence (Percentage of Affected Web Applications)

Vulnerability Prevalence in ColdFusion Applications (Percentage of Applications Affected)

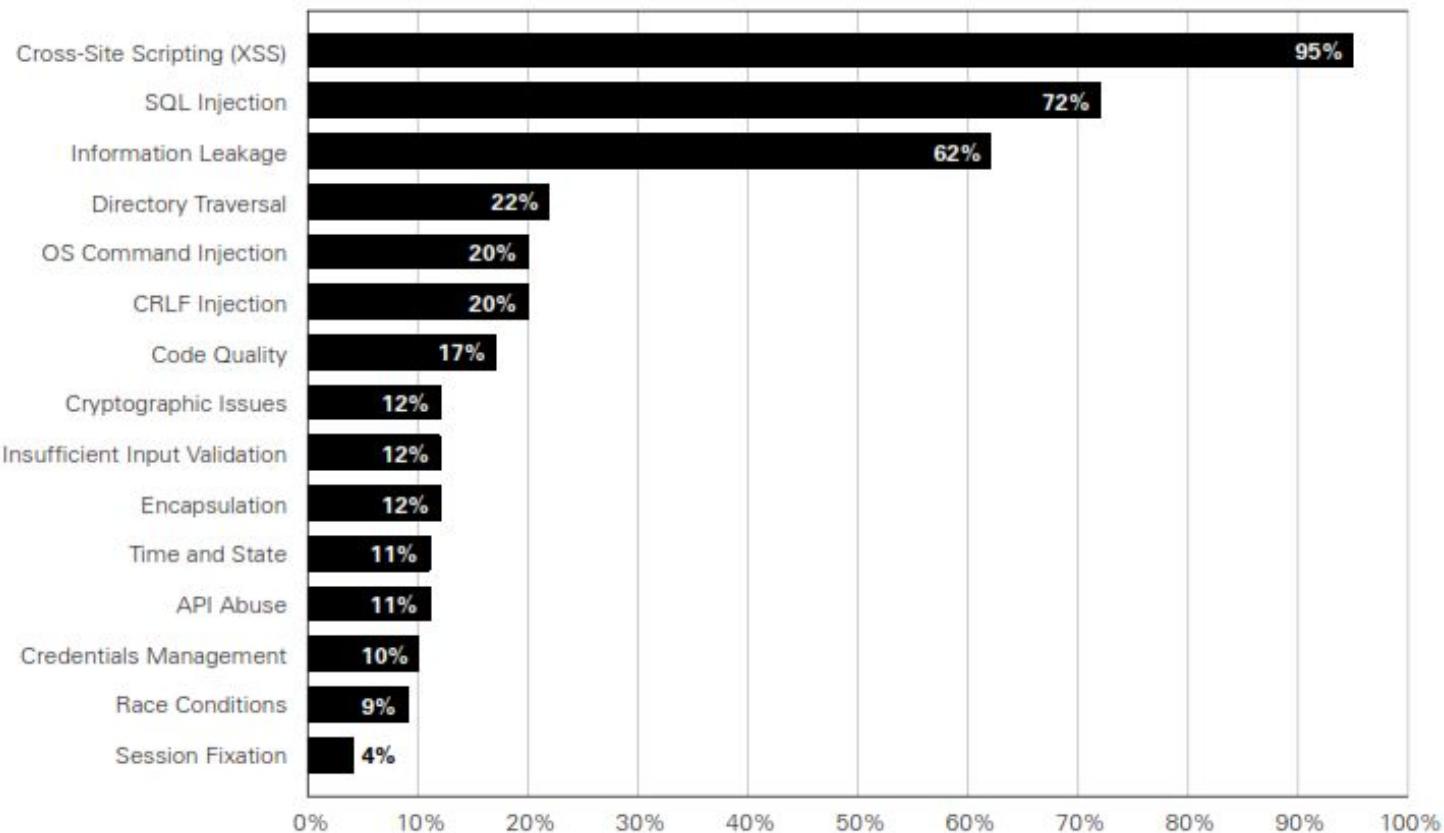
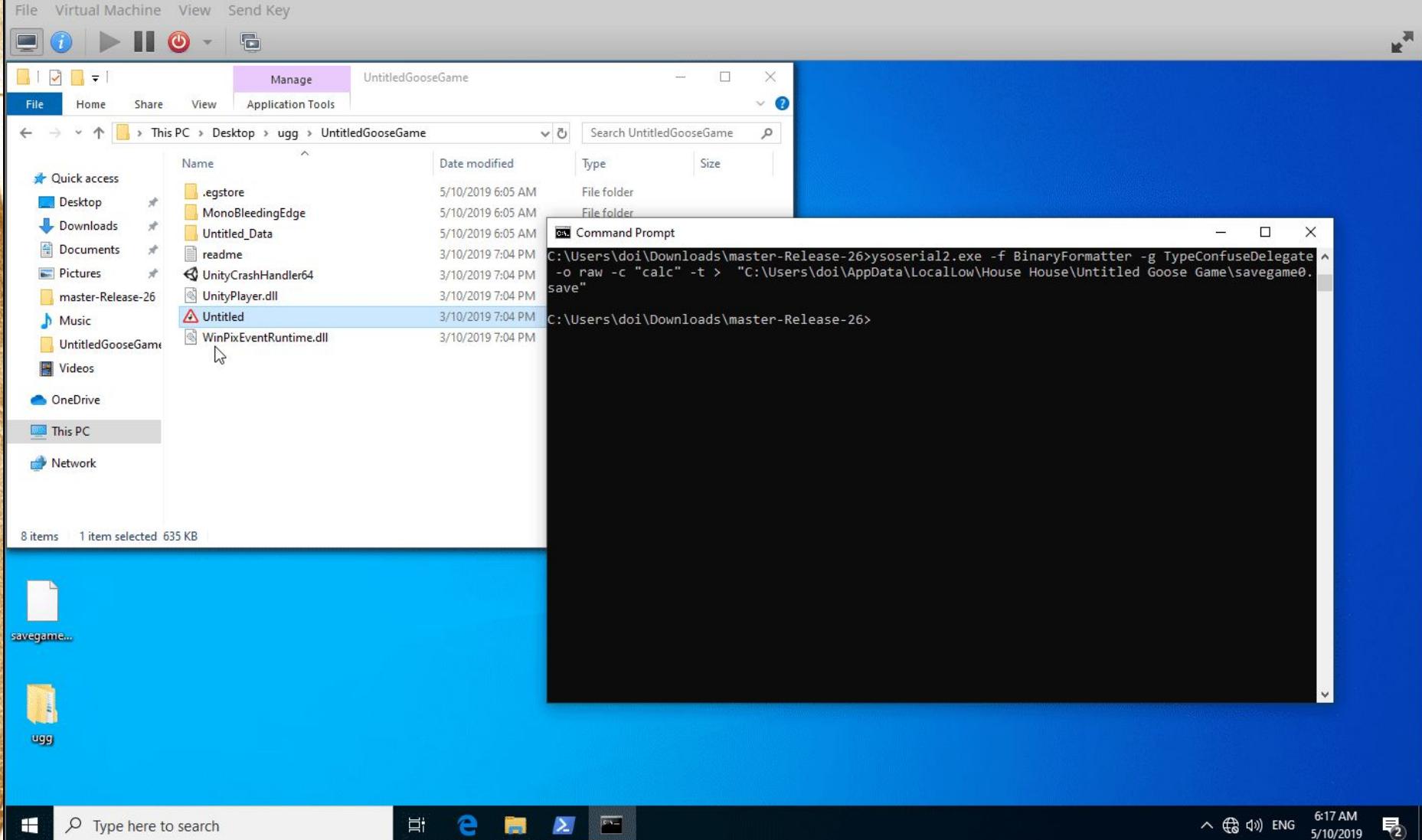


Figure 19: Vulnerability Prevalence in ColdFusion Applications (Percentage of Applications Affected)



<https://www.flickr.com/photos/20056268@N00/2094726257>

HONK on QEMU/KVM





www.jhoch.com

@jkuemerle / www.kuemerle.com

www.fppt.info

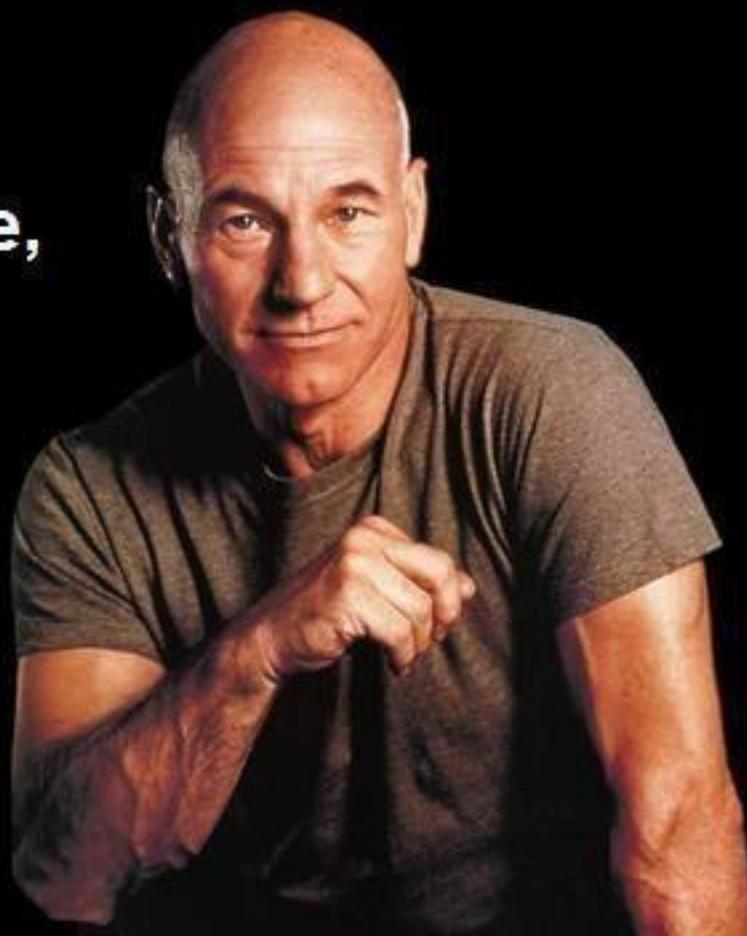


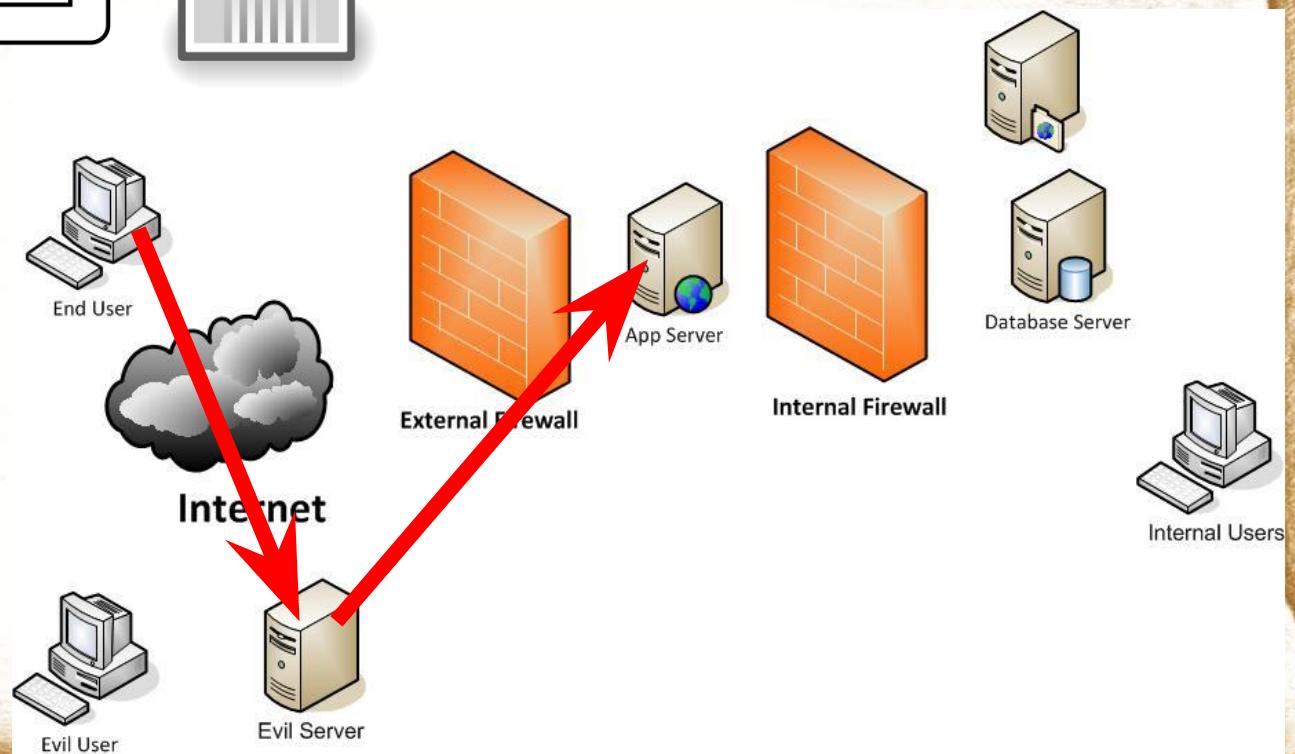
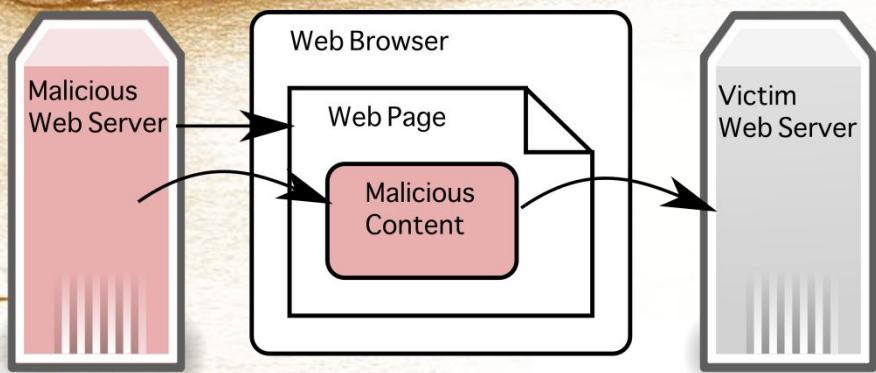


<https://www.flickr.com/photos/foilman/38557740656/>

**"Use the force,
Harry"**

- Gandalf





Injection

Security Misconfiguration

Broken Authentication

Cross Site Scripting (XSS)

Sensitive Data Exposure

Insecure Deserialization

XML External Entities (XXE)

Components With Known Vulnerabilities

Broken Access Control

Insufficient Logging and Monitoring

M1 - Improper Platform Usage	This category covers misuse of a platform feature or failure to use platform security controls. It might include Android intents, platform permissions, misuse of TouchID, the Keychain, or some other security control that is part of the mobile operating system. There are several ways that mobile apps can experience this risk.
M2 - Insecure Data Storage	This new category is a combination of M2 + M4 from Mobile Top Ten 2014. This covers insecure data storage and unintended data leakage.
M3 - Insecure Communication	This covers poor handshaking, incorrect SSL versions, weak negotiation, cleartext communication of sensitive assets, etc.
M4 - Insecure Authentication	This category captures notions of authenticating the end user or bad session management. This can include: <ul style="list-style-type: none">• Failing to identify the user at all when that should be required• Failure to maintain the user's identity when it is required• Weaknesses in session management
M5 - Insufficient Cryptography	The code applies cryptography to a sensitive information asset. However, the cryptography is insufficient in some way. Note that anything and everything related to TLS or SSL goes in M3. Also, if the app fails to use cryptography at all when it should, that probably belongs in M2. This category is for issues where cryptography was attempted, but it wasn't done correctly.
M6 - Insecure Authorization	This is a category to capture any failures in authorization (e.g., authorization decisions in the client side, forced browsing, etc.). It is distinct from authentication issues (e.g., device enrolment, user identification, etc.). If the app does not authenticate users at all in a situation where it should (e.g., granting anonymous access to some resource or service when authenticated and authorized access is required), then that is an authentication failure not an authorization failure.
M7 - Client Code Quality	This was the "Security Decisions Via Untrusted Inputs", one of our lesser-used categories. This would be the catch-all for code-level implementation problems in the mobile client. That's distinct from server-side coding mistakes. This would capture things like buffer overflows, format string vulnerabilities, and various other code-level mistakes where the solution is to rewrite some code that's running on the mobile device.
M8 - Code Tampering	This category covers binary patching, local resource modification, method hooking, method swizzling, and dynamic memory modification. Once the application is delivered to the mobile device, the code and data resources are resident there. An attacker can either directly modify the code, change the contents of memory dynamically, change or replace the system APIs that the application uses, or modify the application's data and resources. This can provide the attacker a direct method of subverting the intended use of the software for personal or monetary gain.
M9 - Reverse Engineering	This category includes analysis of the final core binary to determine its source code, libraries, algorithms, and other assets. Software such as IDA Pro, Hopper, otool, and other binary inspection tools give the attacker insight into the inner workings of the application. This may be used to exploit other latent vulnerabilities in the application, as well as revealing information about back end servers, cryptographic constants and ciphers, and intellectual property.
M10 - Extraneous Functionality	Often, developers include hidden backdoor functionality or other internal development security controls that are not intended to be released into a production environment. For example, a developer may accidentally include a password as a comment in a hybrid app. Another example includes disabling of 2-factor authentication during testing.

Vulnerability Distribution for Mobile Platforms (Share of Total Vulnerabilities Found)

Android		iOS	Java ME		
CRLF Injection	37%	Information Leakage	62%	Cryptographic Issues	47%
Cryptographic Issues	33%	Error Handling	20%	Information Leakage	47%
Information Leakage	10%	Cryptographic Issues	7%	Directory Traversal	3%
SQL Injection	9%	Directory Traversal	6%	Insufficient Input Validation	2%
Time and State	4%	Buffer Management Errors	3%	Credentials Management	<1%

Figure 20: Vulnerability Distribution for Mobile Platforms (Share of Total Vulnerabilities Found)

Android Vulnerability Prevalence (Percentage of Applications Affected)

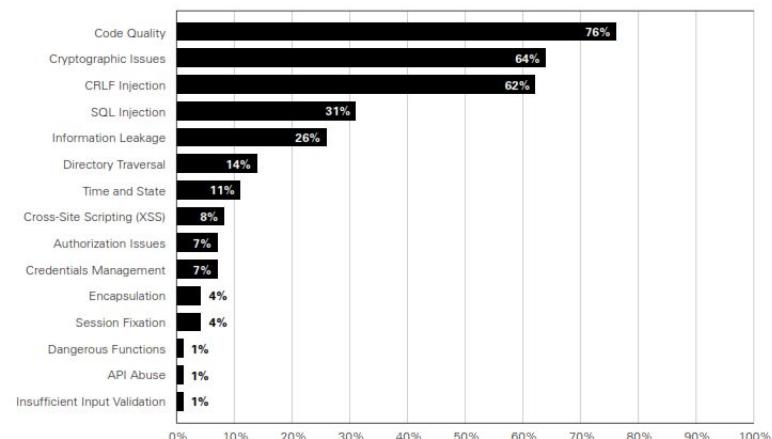


Figure 21: Android Vulnerability Prevalence (Percentage of Applications Affected)

iOS (ObjectiveC) Vulnerability Prevalence (Percentage of Applications Affected)

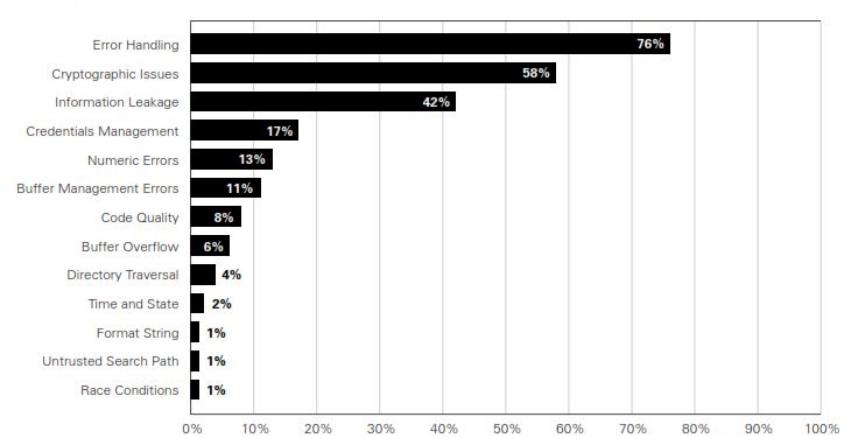


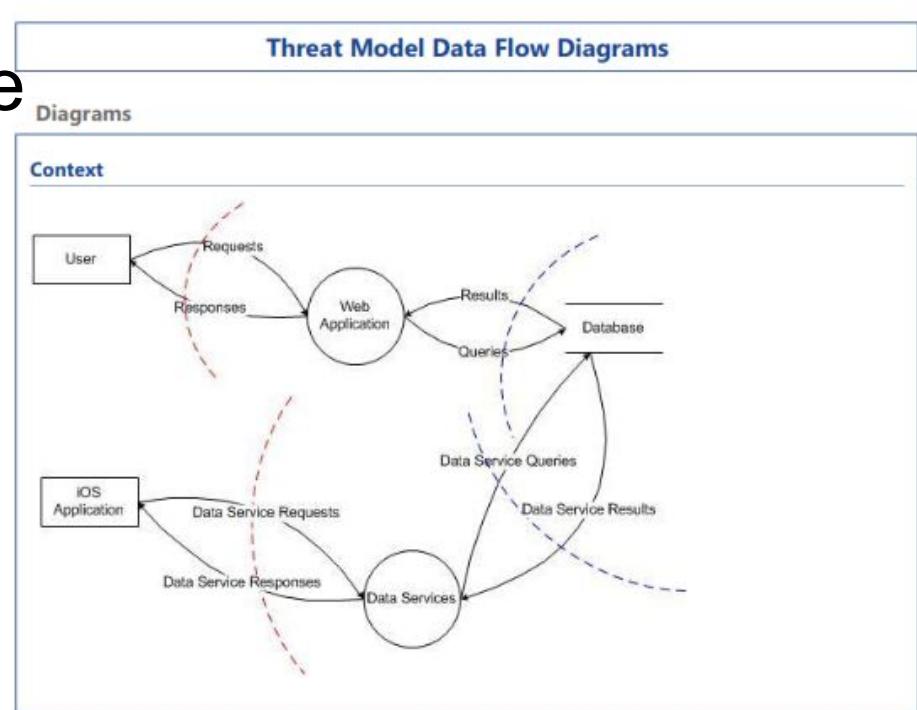
Figure 22: iOS (ObjectiveC) Vulnerability Prevalence (Percentage of Applications Affected)

<https://github.com/skylot/jadx>

The screenshot shows the jadx-gui interface. On the left is a tree view of the class hierarchy under 'jad'. The right side shows the decompiled Java code for the `JadxDecompiler` class. The code is annotated with various colors: blue for keywords like `private`, `void`, `int`, etc.; red for strings and comments; and green for class names like `JavaClass`. The code itself is as follows:

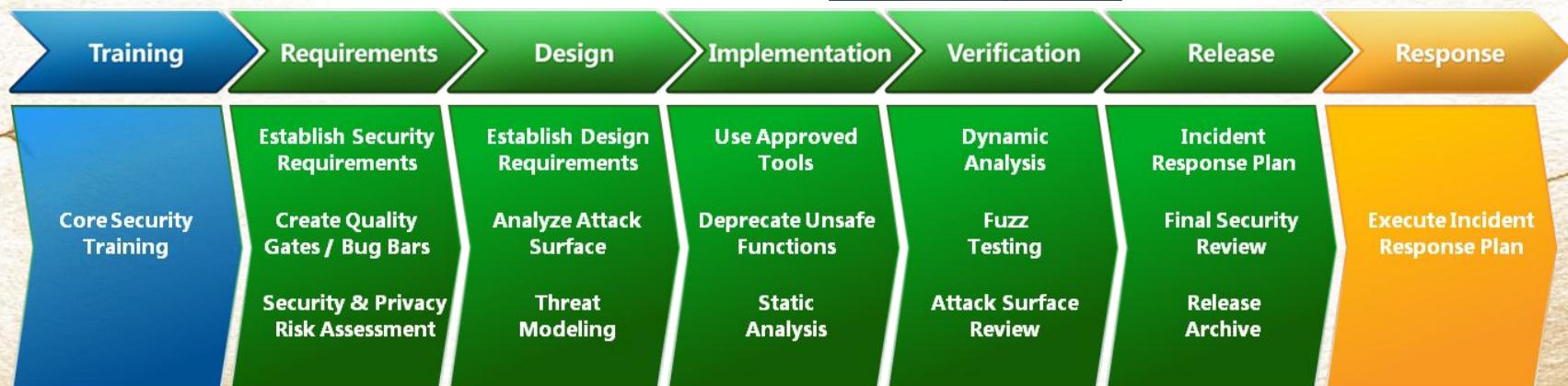
```
112     decompile();
113     return this.cls.getCode().getAnnotations();
114 }
115
116 private void load() {
117     Iterator i$;
118     int inClsCount = this.cls.getInnerClasses().size();
119     if (inClsCount != 0) {
120         List<JavaClass> list = new ArrayList(inClsCount);
121         i$ = this.cls.getInnerClasses().iterator();
122         while (i$.hasNext()) {
123             ClassNode inner = (ClassNode) i$.next();
124             if (!inner.contains(AFlag.DONT_GENERATE)) {
125                 JavaClass javaClass = new JavaClass(inner, this);
126                 javaClass.load();
127                 list.add(javaClass);
128             }
129         }
130         this.innerClasses = Collections.unmodifiableList(list);
131     }
132     int fieldsCount = this.cls.getFields().size();
133     if (fieldsCount != 0) {
134         List<JavaField> flds = new ArrayList(fieldsCount);
135         i$ = this.cls.getFields().iterator();
136         while (i$.hasNext()) {
137             FieldNode f = (FieldNode) i$.next();
138             if (!f.contains(AFlag.DONT_GENERATE)) {
139                 flds.add(new JavaField(f, this));
140             }
141         }
142         this.fields = Collections.unmodifiableList(flds);
143     }
144     int methodsCount = this.cls.getMethods().size();
145 }
```

Spoofing Tampering Repudiation Information Disclosure Denial of Service Elevation of Privilege





Exploring Information Security podcast,



wful
rots
a.
The
ndy
84A.
—
asa
bar
r. A

~~Need not apply.
Camdom 818.~~

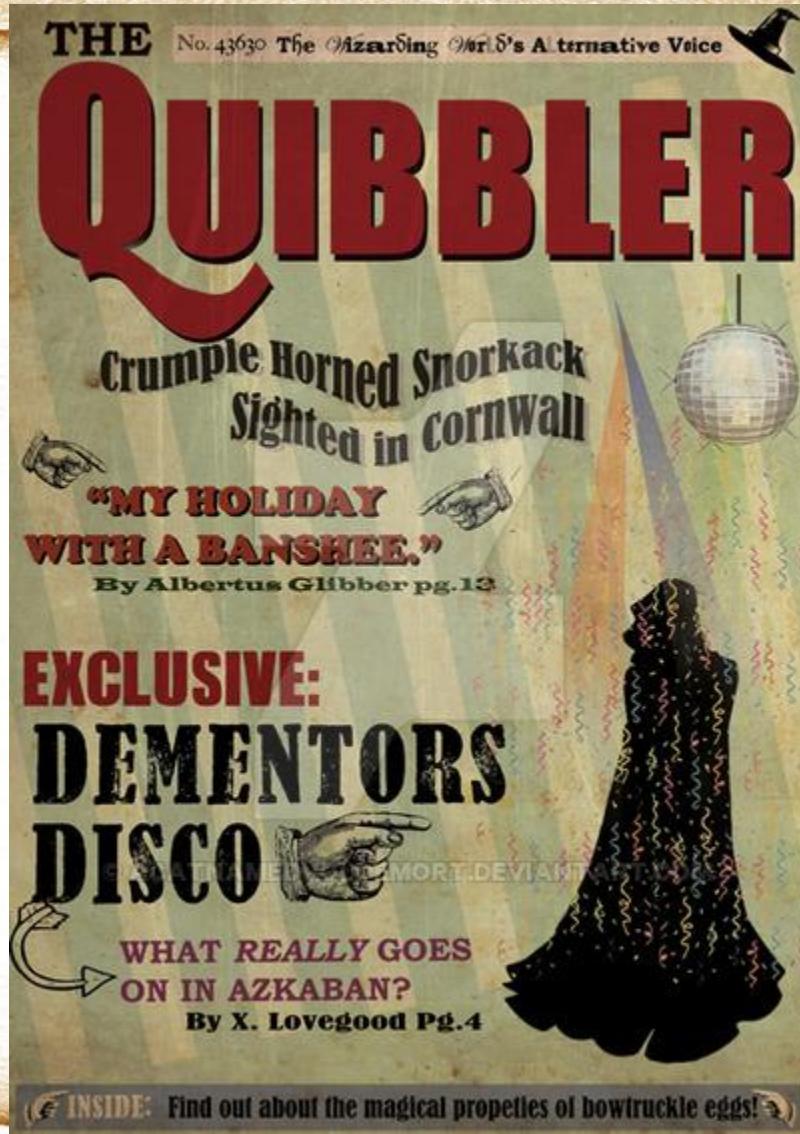
Evil? Eager to be
minion? Want rad
tattoo? Contact
Mortimer D'Vol
weeknights after
8.

~~SWW seeks SWW.
Must love dogs~~

rent. Full
and kitch
bedroom
Quarrel.
—
Snakes a
with som
rats but
carrying
Death. C
oid Mar



<http://www.potterworldmc.com/forum/m/23366915/viewthread/24561719-quibbler>



References

- <http://www.troyhunt.com>
 - <http://www.troyhunt.com/2011/12/free-ebook-owasp-top-10-for-net.html>
- <http://www.owasp.org>
 - <http://www.youtube.com/user/AppsecTutorialSeries?feature=watch>
- <http://www.microsoft.com/security/sdl/default.aspx>
- <http://blogs.msdn.com/b/sdl>
- <http://bsimm.com>
- <http://www.amazon.com/Writing-Secure-Second-Michael-Howard/dp/0735617228>
- <http://www.nosqlmap.net/index.html>
- <http://www.veracode.com/resources/state-of-software-security>

Tools

- <https://www.kali.org/>
- OWASP ZAP
https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
- <https://www.microsoft.com/en-us/download/details.aspx?id=49168> (Threat Model designer)
- JuiceShop
<https://github.com/bkimminich/juice-shop>
- ysoserial.NET
<https://github.com/pwntester/ysoserial.net>