Web Application (OWASP Top 10) Scan Report

Report Generated: December 14, 2015

1 Introduction

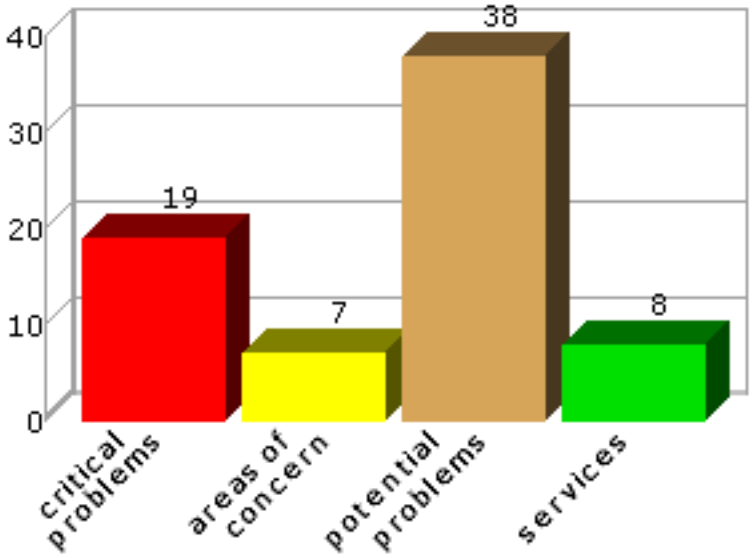
On December 14, 2015, at 4:48 PM, an OWASP Top 10 vulnerability assessment was conducted using the SAINT 8.9.28 vulnerability scanner. The scan discovered a total of one live host, and detected 19 critical problems, seven areas of concern, and 38 potential problems. The hosts and problems detected are discussed in greater detail in the following sections.

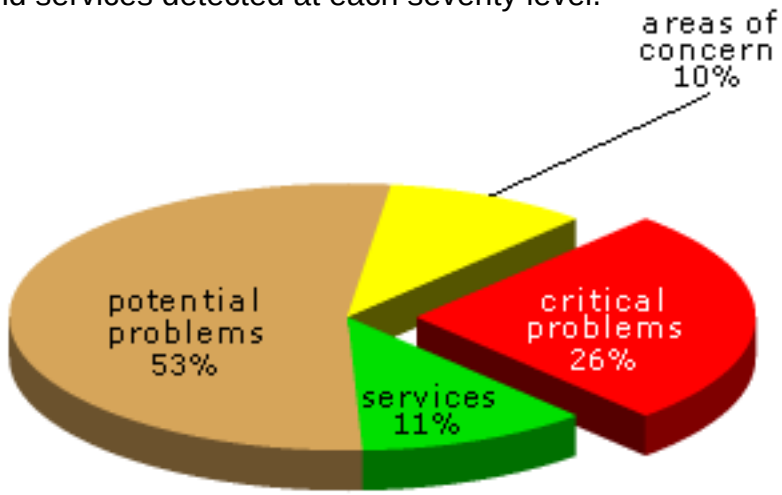
2 Summary

The sections below summarize the results of the scan.

2.1 Vulnerabilities by Severity

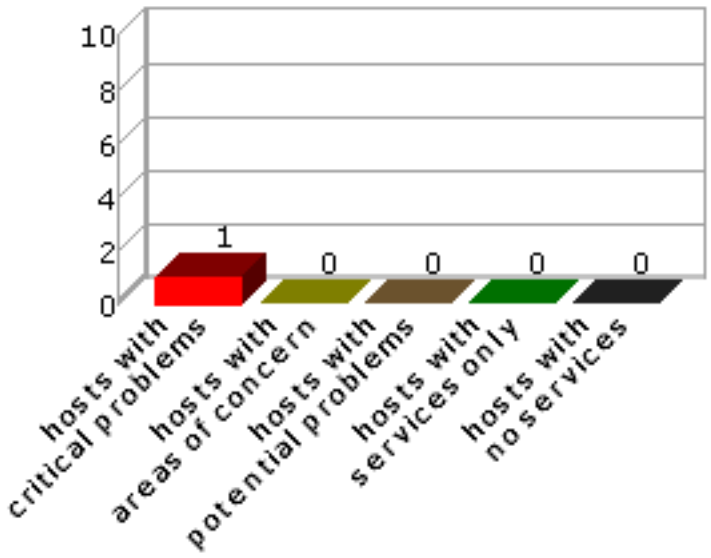
This section shows the overall number of vulnerabilities and services detected at each severity level.





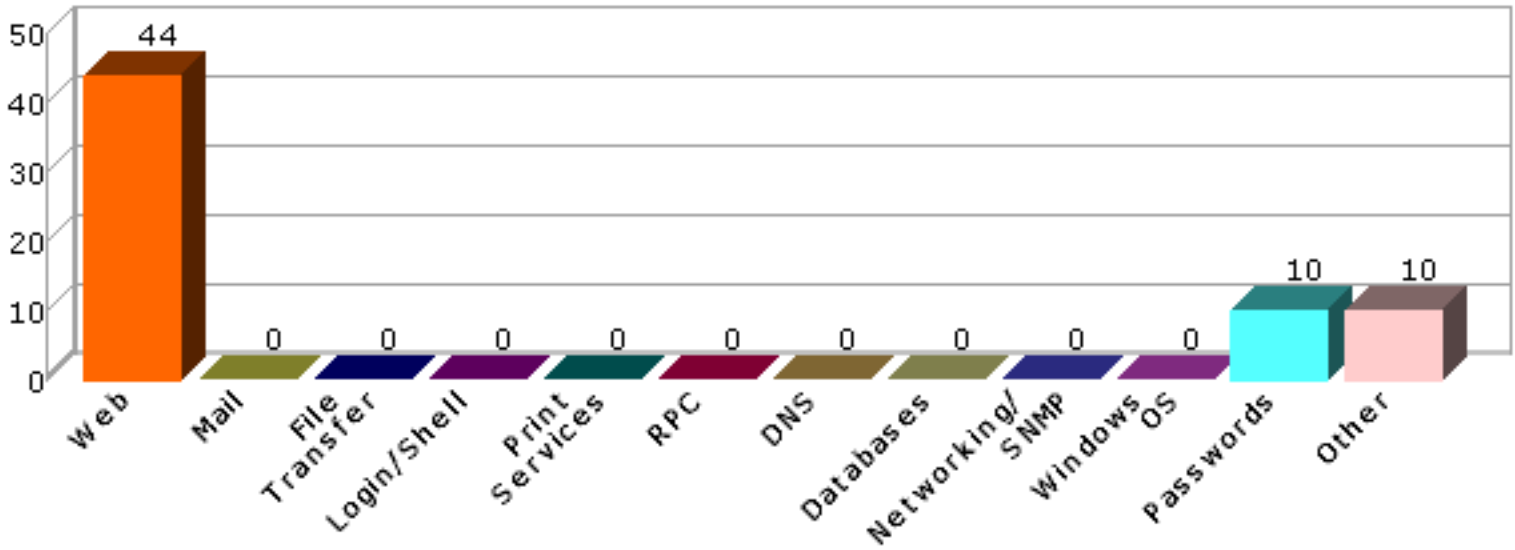
2.2 Hosts by Severity

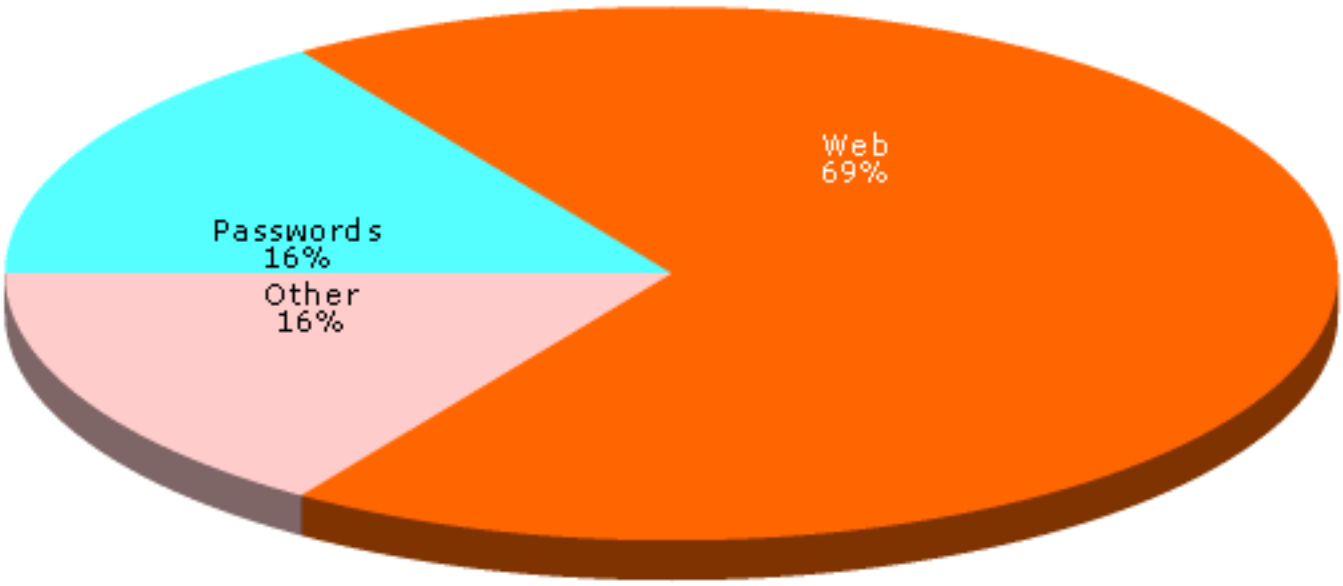
This section shows the overall number of hosts detected at each severity level. The severity level of a host is defined as the highest vulnerability severity level detected on that host.



2.3 Vulnerabilities by Class

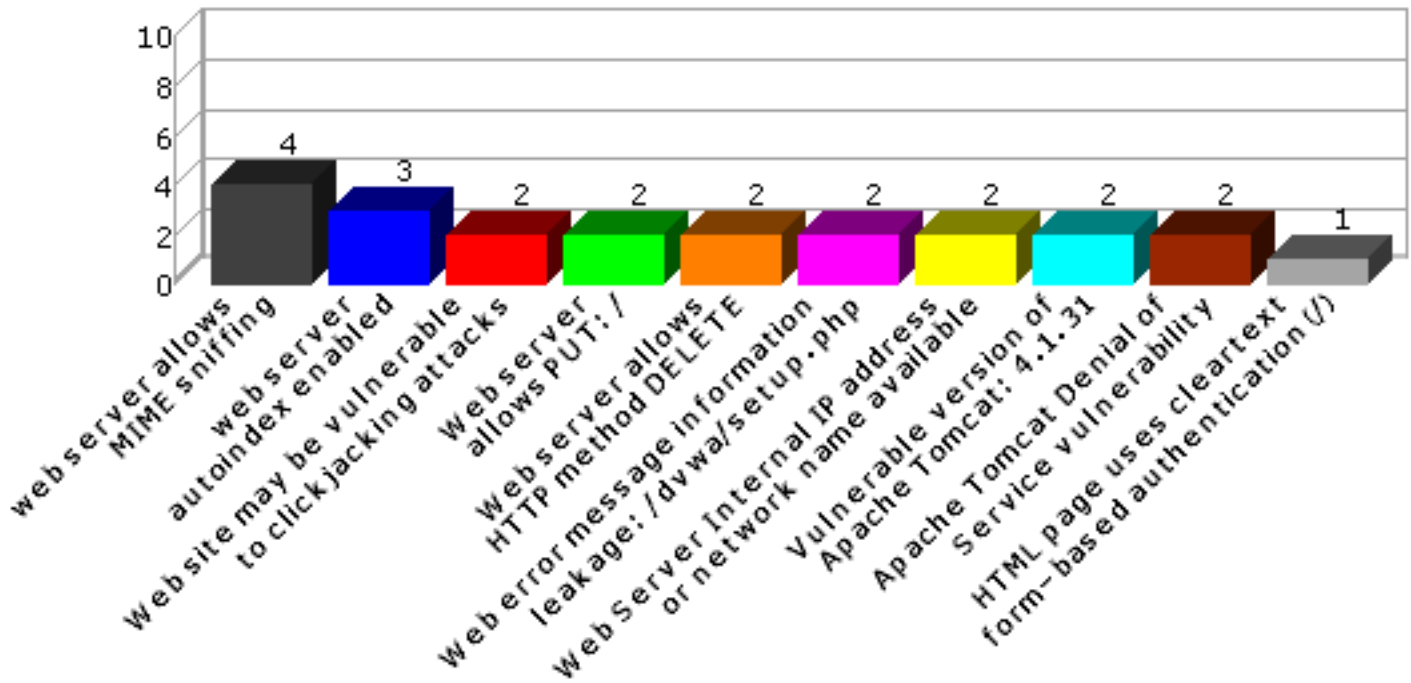
This section shows the number of vulnerabilities detected in each vulnerability class.





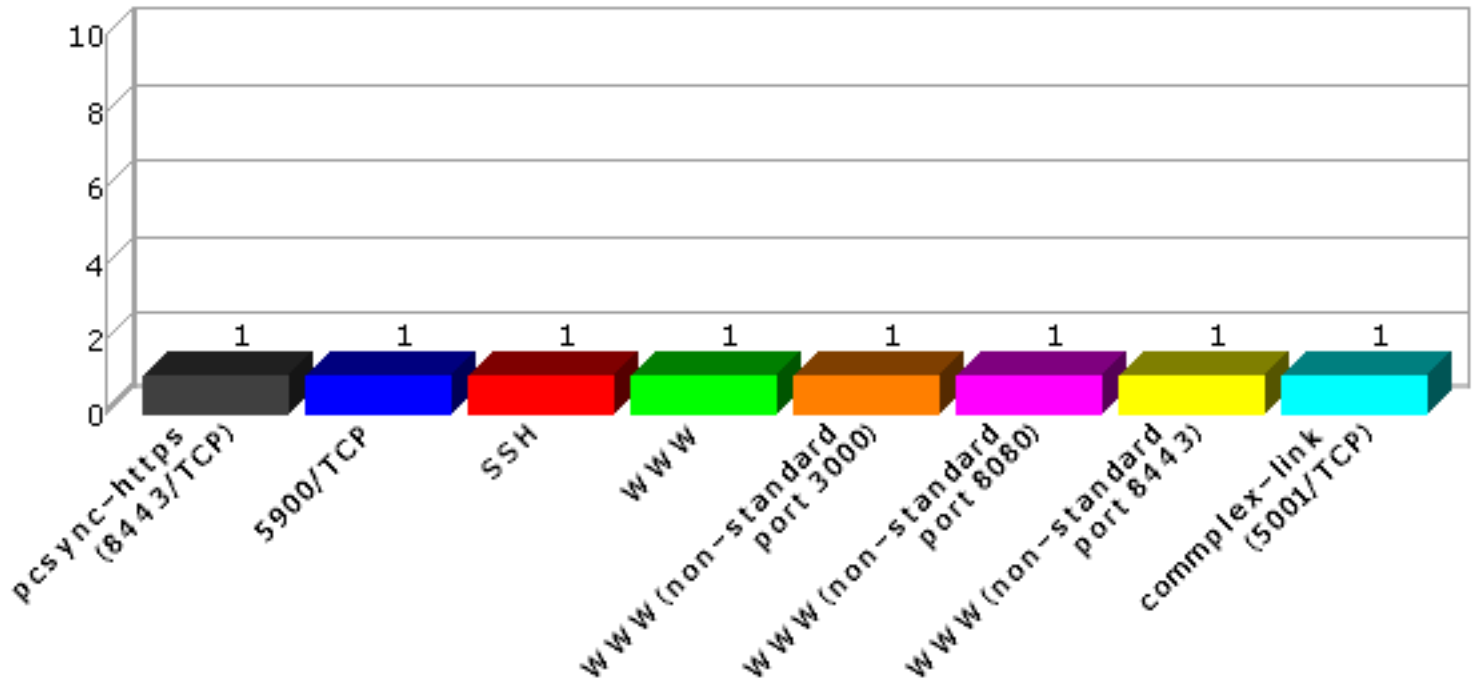
2.4 Top 10 Vulnerabilities

This section shows the most common vulnerabilities detected, and the number of occurrences.



2.5 Top 10 Services

This section shows the most common services detected, and the number of hosts on which they were detected.



3 Overview

The following tables present an overview of the hosts discovered on the network and the vulnerabilities contained therein.

3.1 Host List

This table presents an overview of the hosts discovered on the network.

Host Name	Netbios Name	IP Address	Host Type	Critical Problems	Areas of Concern	Potential Problems
dojo1.sainttest.local		10.8.0.186	Ubuntu 10.04	19	7	38

3.2 Vulnerability List

This table presents an overview of the vulnerabilities detected on the network.

Host Name	Port	Severity	Vulnerability / Service	Class	CVE	Max. CVSSv2 Base Score
dojo1.sainttest.local	8080 /tcp	critical	Apache Tomcat Denial of Service vulnerability	Web	CVE-2005-3510	5.0
dojo1.sainttest.local	8443 /tcp	critical	Apache Tomcat Denial of Service vulnerability	Web	CVE-2005-3510	5.0
dojo1.sainttest.local	8080 /tcp	critical	Vulnerable version of Apache Tomcat: 4.1.31	Web	CVE-2005-2090 CVE-2005-4838 CVE-2006-7196 CVE-2006-7197 CVE-2007-1355 CVE-2007-1358 CVE-2007-1858 CVE-2007-2449 CVE-2007-2450 CVE-2007-3382 CVE-2007-3383 CVE-2007-3385 CVE-2007-5333 CVE-2007-5461 CVE-2008-1232 CVE-2008-2370 CVE-2008-2938 CVE-2008-3271 CVE-2008-5515 CVE-2009-0033 CVE-2009-0580 CVE-2009-0781 CVE-2009-0783 CVE-2011-2204 CVE-2011-2526 CVE-2012-5568 CVE-2013-4286 CVE-2013-4322 CVE-2013-4590	7.8

dojo1.sainttest.local	8443 /tcp	critical	Vulnerable version of Apache Tomcat: 4.1.31	Web	CVE-2005-2090 CVE-2005-4838 CVE-2006-7196 CVE-2006-7197 CVE-2007-1355 CVE-2007-1358 CVE-2007-1858 CVE-2007-2449 CVE-2007-2450 CVE-2007-3382 CVE-2007-3383 CVE-2007-3385 CVE-2007-5333 CVE-2007-5461 CVE-2008-1232 CVE-2008-2370 CVE-2008-2938 CVE-2008-3271 CVE-2008-5515 CVE-2009-0033 CVE-2009-0580 CVE-2009-0781 CVE-2009-0783 CVE-2011-2204 CVE-2011-2526 CVE-2012-5568 CVE-2013-4286 CVE-2013-4322 CVE-2013-4590	7.8
dojo1.sainttest.local	80 /tcp	critical	guessed password to web form: /dvwa/login.php (admin:password)	Passwords		9.0
dojo1.sainttest.local	80 /tcp	critical	guessed password to web form: /dvwa/vulnerabilities/brute/../.. /login.php (admin:password)	Passwords		9.0
dojo1.sainttest.local	80 /tcp	critical	guessed password to web form: /dvwa/vulnerabilities/csrf/../.. /login.php (admin:password)	Passwords		9.0
dojo1.sainttest.local	80 /tcp	critical	guessed password to web form: /dvwa/vulnerabilities/exec/../.. /login.php (admin:password)	Passwords		9.0
dojo1.sainttest.local	80 /tcp	critical	guessed password to web form: /dvwa/vulnerabilities/fi/../.. /login.php (admin:password)	Passwords		9.0
dojo1.sainttest.local	80 /tcp	critical	guessed password to web form: /dvwa/vulnerabilities/sqli/../.. /login.php (admin:password)	Passwords		9.0
dojo1.sainttest.local	80 /tcp	critical	guessed password to web form: /dvwa/vulnerabilities/sqli_blind/../.. /login.php (admin:password)	Passwords		9.0
dojo1.sainttest.local	80 /tcp	critical	guessed password to web form: /dvwa/vulnerabilities/upload/../.. /login.php (admin:password)	Passwords		9.0
dojo1.sainttest.local	80 /tcp	critical	guessed password to web form: /dvwa/vulnerabilities/xss_r/../.. /login.php (admin:password)	Passwords		9.0
dojo1.sainttest.local	80 /tcp	critical	guessed password to web form: /dvwa/vulnerabilities/xss_s/../.. /login.php (admin:password)	Passwords		9.0
dojo1.sainttest.local	8080 /tcp	critical	Web server allows HTTP method DELETE	Web		7.8

dojo1.sainttest.local	8443 /tcp	critical	Web server allows HTTP method DELETE	Web	7.8
dojo1.sainttest.local	8443 /tcp	critical	Web server allows PUT: /	Web	7.8
dojo1.sainttest.local	8080 /tcp	critical	Web server allows PUT: /	Web	7.8

dojo1.sainttest.local	80 /tcp	critical	vulnerable PHP version: 5.3.2	Web	CVE-2006-7243 CVE-2007-1581 CVE-2010-1860 CVE-2010-1861 CVE-2010-1862 CVE-2010-1864 CVE-2010-1866 CVE-2010-1868 CVE-2010-1914 CVE-2010-1915 CVE-2010-1917 CVE-2010-2094 CVE-2010-2097 CVE-2010-2100 CVE-2010-2101 CVE-2010-2190 CVE-2010-2191 CVE-2010-2225 CVE-2010-2531 CVE-2010-3065 CVE-2010-3436 CVE-2010-3709 CVE-2010-3710 CVE-2010-3870 CVE-2010-4150 CVE-2010-4156 CVE-2010-4409 CVE-2010-4645 CVE-2010-4697 CVE-2011-0708 CVE-2011-0753 CVE-2011-0754 CVE-2011-0755 CVE-2011-1092 CVE-2011-1148 CVE-2011-1398 CVE-2011-1464 CVE-2011-1466 CVE-2011-1467 CVE-2011-1468 CVE-2011-1469 CVE-2011-1470 CVE-2011-1471 CVE-2011-1657 CVE-2011-1938 CVE-2011-2202 CVE-2011-2483 CVE-2011-3182 CVE-2011-3267 CVE-2011-3268 CVE-2011-4718 CVE-2011-4885 CVE-2012-0057 CVE-2012-0788 CVE-2012-0789 CVE-2012-1823 CVE-2012-2311 CVE-2012-2688 CVE-2012-3365 CVE-2012-3450 CVE-2013-1635 CVE-2013-1643 CVE-2013-2110

CVE-2013-4113	10.0
CVE-2013-4248	
CVE-2013-4636	
CVE-2013-6420	
CVE-2014-0207	
CVE-2014-3478	
CVE-2014-3479	
CVE-2014-3480	
CVE-2014-3487	
CVE-2014-3515	
CVE-2014-3668	
CVE-2014-3669	
CVE-2014-3670	
CVE-2014-3710	
CVE-2014-3981	
CVE-2014-4049	
CVE-2014-4670	
CVE-2014-4698	
CVE-2014-8142	
CVE-2014-9426	
CVE-2014-9427	
CVE-2014-9709	
CVE-2015-0231	
CVE-2015-0232	
CVE-2015-0273	
CVE-2015-1351	
CVE-2015-1352	
CVE-2015-2301	
CVE-2015-2348	
CVE-2015-2783	
CVE-2015-2787	
CVE-2015-3152	
CVE-2015-3307	
CVE-2015-3329	
CVE-2015-4021	
CVE-2015-4022	
CVE-2015-4024	
CVE-2015-4025	
CVE-2015-4026	
CVE-2015-4147	
CVE-2015-4148	
CVE-2015-4598	
CVE-2015-4642	
CVE-2015-4643	
CVE-2015-4644	
CVE-2015-5589	
CVE-2015-5590	
CVE-2015-6834	
CVE-2015-6835	
CVE-2015-6836	
CVE-2015-6837	
CVE-2015-6838	
CVE-2015-7803	
CVE-2015-7804	

dojo1.sainttest.local	80 /tcp	concern	vulnerable Apache version: 2.2.14	Web	CVE-2009-3560 CVE-2009-3720 CVE-2010-0425 CVE-2010-0434 CVE-2010-1452 CVE-2010-1623 CVE-2010-2068 CVE-2011-0419 CVE-2011-1928 CVE-2011-3192 CVE-2011-3348 CVE-2011-3607 CVE-2011-4415 CVE-2012-0031 CVE-2012-0053 CVE-2012-0883 CVE-2012-2687 CVE-2012-3499 CVE-2012-4558 CVE-2013-1862 CVE-2013-1896 CVE-2013-5704 CVE-2013-6438 CVE-2014-0098 CVE-2014-3581 CVE-2014-3583 CVE-2014-8109	10.0
dojo1.sainttest.local	3000 /tcp	concern	.htaccess file is accessible	Web		2.6
dojo1.sainttest.local	8080 /tcp	concern	Web Server Internal IP address or network name available	Web	CVE-2000-0649 CVE-2002-0419	5.0
dojo1.sainttest.local	8443 /tcp	concern	Web Server Internal IP address or network name available	Web	CVE-2000-0649 CVE-2002-0419	5.0
dojo1.sainttest.local	80 /tcp	concern	Web error message information leakage: /dvwa/instructions.php	Web		2.6
dojo1.sainttest.local	80 /tcp	concern	Web error message information leakage: /dvwa/setup.php	Web		2.6
dojo1.sainttest.local	80 /tcp	concern	Web error message information leakage: /dvwa/setup.php	Web		2.6
dojo1.sainttest.local	80 /tcp	potential	Apache ETag header discloses inode numbers	Web	CVE-2003-1418	4.3
dojo1.sainttest.local	80 /tcp	potential	Apache mod_proxy_ajp 2.2.14 is vulnerable	Web	CVE-2010-0408	5.0
dojo1.sainttest.local	80 /tcp	potential	mod_proxy vulnerability in Apache version: 2.2.14	Web	CVE-2011-3368 CVE-2011-3639 CVE-2011-4317	5.0
dojo1.sainttest.local	80 /tcp	potential	Web site may be vulnerable to clickjacking attacks	Web		2.6
dojo1.sainttest.local	3000 /tcp	potential	Web site may be vulnerable to clickjacking attacks	Web		2.6
dojo1.sainttest.local	80 /tcp	potential	Potentially sensitive information found on web page (/dvwa/)	Other		2.6
dojo1.sainttest.local	80 /tcp	potential	Potentially sensitive information found on web page (/dvwa/.)	Other		2.6
dojo1.sainttest.local	80 /tcp	potential	Potentially sensitive information found on web page (/dvwa /index.php)	Other		2.6
dojo1.sainttest.local	8443 /tcp	potential	weak https cache policy	Web		2.6

dojo1.sainttest.local	80 /tcp	potential	CGI Gives Information about System (phpinfo.php)	Web		2.6
dojo1.sainttest.local	3000 /tcp	potential	web server allows MIME sniffing	Web		2.6
dojo1.sainttest.local	8080 /tcp	potential	web server allows MIME sniffing	Web		2.6
dojo1.sainttest.local	80 /tcp	potential	web server allows MIME sniffing	Web		2.6
dojo1.sainttest.local	8443 /tcp	potential	web server allows MIME sniffing	Web		2.6
dojo1.sainttest.local	8443 /tcp	potential	web server autoindex enabled	Web	CVE-1999-0569	10.0
dojo1.sainttest.local	80 /tcp	potential	web server autoindex enabled	Web	CVE-1999-0569	10.0
dojo1.sainttest.local	8080 /tcp	potential	web server autoindex enabled	Web	CVE-1999-0569	10.0
dojo1.sainttest.local	80 /tcp	potential	Cookie without HTTPOnly attribute can be accessed by scripts: PHPSESSID	Web		2.6
dojo1.sainttest.local	3000 /tcp	potential	Cookie without HTTPOnly attribute can be accessed by scripts: _session_id	Web		2.6
dojo1.sainttest.local	3000 /tcp	potential	Autocomplete enabled for password input (/)	Web		2.6
dojo1.sainttest.local	3000 /tcp	potential	HTML page uses cleartext form-based authentication (/)	Web		2.6
dojo1.sainttest.local	80 /tcp	potential	HTML page uses cleartext form-based authentication (/dvwa /login.php)	Web		2.6
dojo1.sainttest.local	80 /tcp	potential	HTML page uses cleartext form-based authentication (/dvwa /vulnerabilities/brute/../../login.php)	Web		2.6
dojo1.sainttest.local	80 /tcp	potential	HTML page uses cleartext form-based authentication (/dvwa /vulnerabilities/csrf/../../login.php)	Web		2.6
dojo1.sainttest.local	80 /tcp	potential	HTML page uses cleartext form-based authentication (/dvwa /vulnerabilities/exec/../../login.php)	Web		2.6
dojo1.sainttest.local	80 /tcp	potential	HTML page uses cleartext form-based authentication (/dvwa /vulnerabilities/fi/../../login.php)	Web		2.6
dojo1.sainttest.local	80 /tcp	potential	HTML page uses cleartext form-based authentication (/dvwa /vulnerabilities/sqli/../../login.php)	Web		2.6
dojo1.sainttest.local	80 /tcp	potential	HTML page uses cleartext form-based authentication (/dvwa /vulnerabilities/sqli_blind/../.. /login.php)	Web		2.6
dojo1.sainttest.local	80 /tcp	potential	HTML page uses cleartext form-based authentication (/dvwa /vulnerabilities/upload/../.. /login.php)	Web		2.6
dojo1.sainttest.local	80 /tcp	potential	HTML page uses cleartext form-based authentication (/dvwa /vulnerabilities/xss_r/../../login.php)	Web		2.6

dojo1.sainttest.local	80 /tcp	potential	HTML page uses cleartext form-based authentication (/dvwa /vulnerabilities/xss_s/../../login.php)	Web		2.6
dojo1.sainttest.local	8443 /tcp	potential	weak RSA public key	Other		2.6
dojo1.sainttest.local	8443 /tcp	potential	SSL certificate is expired	Other		2.6
dojo1.sainttest.local	8443 /tcp	potential	SSL certificate is self signed	Other		2.6
dojo1.sainttest.local	8443 /tcp	potential	SSL certificate subject does not match target	Other		2.6
dojo1.sainttest.local	8443 /tcp	potential	SSL server accepts weak ciphers	Other		2.6
dojo1.sainttest.local	8443 /tcp	potential	TLS/SSL server accepts export ciphers (FREAK/Logjam attack)	Other	CVE-2015-0204 CVE-2015-4000	4.3
dojo1.sainttest.local	8443 /tcp	potential	SSL certificate is signed with weak hash function: MD5	Other	CVE-2004-2761	5.0
dojo1.sainttest.local	5900 /tcp	service	5900/TCP			
dojo1.sainttest.local	22 /tcp	service	SSH			
dojo1.sainttest.local	80 /tcp	service	WWW			
dojo1.sainttest.local	3000 /tcp	service	WWW (non-standard port 3000)			
dojo1.sainttest.local	8080 /tcp	service	WWW (non-standard port 8080)			
dojo1.sainttest.local	8443 /tcp	service	WWW (non-standard port 8443)			
dojo1.sainttest.local	5001 /tcp	service	commplex-link (5001/TCP)			
dojo1.sainttest.local	8443 /tcp	service	pcsync-https (8443/TCP)			
dojo1.sainttest.local	3000 /tcp	info	Web Directory: /			
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa/			
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa/dvwa/			
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa/dvwa/css/			
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa/dvwa /images/			
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa/dvwa /includes/			
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa/dvwa /includes/DBMS/			
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa/dvwa/js/			
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa /vulnerabilities/brute/			
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa /vulnerabilities/csrf/			
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa /vulnerabilities/exec/			
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa /vulnerabilities/fi/			

dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa /vulnerabilities/sqli/
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa /vulnerabilities/sqli_blind/
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa /vulnerabilities/upload/
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa /vulnerabilities/xss_r/
dojo1.sainttest.local	80 /tcp	info	Web Directory: /dvwa /vulnerabilities/xss_s/
dojo1.sainttest.local	3000 /tcp	info	Web Directory: /images/web/
dojo1.sainttest.local	8080 /tcp	info	Web Directory: /insecure/
dojo1.sainttest.local	8443 /tcp	info	Web Directory: /insecure/
dojo1.sainttest.local	8080 /tcp	info	Web Directory: /insecure /Read-Me/
dojo1.sainttest.local	8443 /tcp	info	Web Directory: /insecure /Read-Me/
dojo1.sainttest.local	8080 /tcp	info	Web Directory: /insecure /Read-Me/source/
dojo1.sainttest.local	8443 /tcp	info	Web Directory: /insecure /Read-Me/source/
dojo1.sainttest.local	3000 /tcp	info	Web Directory: /javascripts/
dojo1.sainttest.local	3000 /tcp	info	Web Directory: /stylesheets/

4 Details

The following sections provide details on the specific vulnerabilities detected on each host.

4.1 dojo1.sainttest.local

IP Address: 10.8.0.186 **Host type:** Ubuntu 10.04
Scan time: Dec 14 16:48:50 2015

Apache Tomcat Denial of Service vulnerability

Severity: Critical Problem **CVE:** CVE-2005-3510

Impact

A remote attacker could view directory listings, view source code of JSP files, view passwords, gain read access to files which are normally inaccessible, cause a denial of service, or possibly write files with the permissions of the user running the server.

Resolutions

Upgrade Tomcat to version 6.0.44 for 6.0, or 7.0.59 for 7.0, or 8.0.17 for 8.0, or higher when available.

Where can I read more about this?

The 404 Error Page Cross Site Scripting vulnerability was reported in Bugtraq ID 37149.

The high stress Denial of Service attack was reported in Bugtraq ID 15325.

Technical Details

Service: 8080:TCP
<html><head><title>Apache Tomcat/4.1.31 - Error report</title><STYLE><!--H1{font-family : sans-serif,Arial,Tahoma;color : white;background-color : #0086b2;} H3{font-family : sans-serif,Arial,Tahoma;color : white;background-color : #0086b2;} BODY{font-family : sans-serif,Arial,Tahoma;color : black;background-color : white;} B{color : white;background-color : #0086b2;} HR{color : #0086b2;} --></STYLE> < /head><body><h1>HTTP Status 404 - /n0nexi5tent_fi1e.asp</h1><HR size="1" noshade="noshade"><p>type S

Apache Tomcat Denial of Service vulnerability

Severity: Critical Problem **CVE:** CVE-2005-3510

Impact

A remote attacker could view directory listings, view source code of JSP files, view passwords, gain read access to files which are normally inaccessible, cause a denial of service, or possibly write files with the permissions of the user running the server.

Resolutions

Upgrade Tomcat to version 6.0.44 for 6.0, or 7.0.59 for 7.0, or 8.0.17 for 8.0, or higher when available.

Where can I read more about this?

The 404 Error Page Cross Site Scripting vulnerability was reported in Bugtraq ID 37149.

The high stress Denial of Service attack was reported in Bugtraq ID 15325.

Technical Details

Service: pcsync-https
<html><head><title>Apache Tomcat/4.1.31 - Error report</title><STYLE><!--H1{font-family : sans-serif,Arial,Tahoma;color : white;background-color : #0086b2;} H3{font-family : sans-serif,Arial,Tahoma;color : white;background-color : #0086b2;} BODY{font-family : sans-serif,Arial,Tahoma;color : black;background-color : white;} B{color : white;background-color : #0086b2;} HR{color : #0086b2;} --></STYLE> < /head><body><h1>HTTP Status 404 - /n0nexi5tent_fi1e.asp</h1><HR size="1" noshade="noshade"><p>type S

Vulnerable version of Apache Tomcat: 4.1.31

Severity: Critical Problem **CVE:** CVE-2005-2090 CVE-2005-4838 CVE-2006-7196 CVE-2006-7197 CVE-2007-1355 CVE-2007-1358 CVE-2007-1858 CVE-2007-2449 CVE-2007-2450 CVE-2007-3382 CVE-2007-3383 CVE-2007-3385 CVE-2007-5333 CVE-2007-5461 CVE-2008-1232 CVE-2008-2370 CVE-2008-2938 CVE-2008-3271 CVE-2008-5515 CVE-2009-0033

CVE-2009-0580 CVE-2009-0781
CVE-2009-0783 CVE-2011-2204
CVE-2011-2526 CVE-2012-5568
CVE-2013-4286 CVE-2013-4322
CVE-2013-4590

Impact

A remote attacker could view directory listings, view source code of JSP files, view passwords, gain read access to files which are normally inaccessible, cause a denial of service, or possibly write files with the permissions of the user running the server.

Resolutions

Upgrade Tomcat to version 6.0.44 for 6.0, or 7.0.59 for 7.0, or 8.0.17 for 8.0, or higher when available.

Where can I read more about this?

The XML External Entity vulnerability, Chunked Transfer vulnerability, and request smuggling vulnerability were reported in Apache Tomcat 6.x vulnerabilities.

The `Slowloris` denial of service was reported in Bugtraq ID 56686.

The AJP Connector information disclosure vulnerability was reported in Bugtraq ID 28477.

The `sendfile` Security Bypass and Denial of Service vulnerabilities were reported in Secunia Advisory SA45232.

The **`MemoryUserDatabase`** password disclosure vulnerability was reported in Secunia Advisory 44981.

The `404` Error Page Cross Site Scripting vulnerability was reported in Bugtraq ID 37149.

The multiple vulnerabilities fixed in Apache Tomcat 6.0.20 were reported in Secunia Advisory SA35326, and Secunia Advisory SA35344.

The `RemoteFilterValve` Security Bypass vulnerability was reported in Bugtraq ID 31698.

The `HttpServletResponse.sendError()` Cross Site Scripting vulnerability was reported in Bugtraq ID 30496.

The `RequestDispatcher` Information Disclosure vulnerability was reported in Bugtraq ID 30494.

The UTF-8 Directory Traversal vulnerability was reported in Bugtraq ID 30633.

Apache Tomcat security fixes are reported by Apache in Apache Tomcat 3.x vulnerabilities, Apache Tomcat 4.x vulnerabilities, Apache Tomcat 5.x vulnerabilities, and Apache Tomcat 6.x vulnerabilities.

The WebDAV servlet arbitrary file access was reported in Secunia Advisory SA27398.

The quote issues and cross-site scripting vulnerabilities were reported in Secunia Advisory SA26465 and Secunia Advisory SA26466.

The Apache Tomcat security fixes were reported in Secunia Advisory SA24732.

Technical Details

Service: 8080:TCP
<html><head><title>Apache Tomcat/4.1.31 - Error report</title><STYLE><!--H1{font-family : sans-serif,Arial,Tahoma;color : white;background-color : #0086b2;} H3{font-family : sans-serif,Arial,Tahoma;color : white;background-color : #0086b2;} BODY{font-family : sans-serif,Arial,Tahoma;color : black;background-color : white;} B{color : white;background-color : #0086b2;} HR{color : #0086b2;} --></STYLE> < /head><body><h1>HTTP Status 404 - /n0nexi5tent_fi1e.asp</h1><HR size="1" noshade="noshade"><p>type S

Vulnerable version of Apache Tomcat: 4.1.31

Severity: Critical Problem

CVE: CVE-2005-2090 CVE-2005-4838 CVE-2006-7196 CVE-2006-7197 CVE-2007-1355 CVE-2007-1358 CVE-2007-1858 CVE-2007-2449 CVE-2007-2450 CVE-2007-3382 CVE-2007-3383 CVE-2007-3385 CVE-2007-5333 CVE-2007-5461 CVE-2008-1232 CVE-2008-2370 CVE-2008-2938 CVE-2008-3271 CVE-2008-5515 CVE-2009-0033 CVE-2009-0580 CVE-2009-0781 CVE-2009-0783 CVE-2011-2204 CVE-2011-2526 CVE-2012-5568 CVE-2013-4286 CVE-2013-4322 CVE-2013-4590

Impact

A remote attacker could view directory listings, view source code of JSP files, view passwords, gain read access to files which are normally inaccessible, cause a denial of service, or possibly write files with the permissions of the user running the server.

Resolutions

Upgrade Tomcat to version 6.0.44 for 6.0, or 7.0.59 for 7.0, or 8.0.17 for 8.0, or higher when available.

Where can I read more about this?

The XML External Entity vulnerability, Chunked Transfer vulnerability, and request smuggling vulnerability were reported in Apache Tomcat 6.x vulnerabilities.

The `Slowloris` denial of service was reported in Bugtraq ID 56686.

The AJP Connector information disclosure vulnerability was reported in Bugtraq ID 28477.

The `sendfile` Security Bypass and Denial of Service vulnerabilities were reported in Secunia Advisory SA45232.

The **`MemoryUserDatabase`** password disclosure vulnerability was reported in Secunia Advisory 44981.

The `404` Error Page Cross Site Scripting vulnerability was reported in Bugtraq ID 37149.

The multiple vulnerabilities fixed in Apache Tomcat 6.0.20 were reported in Secunia Advisory SA35326, and Secunia Advisory SA35344.

The `RemoteFilterValve` Security Bypass vulnerability was reported in Bugtraq ID 31698.

The `HttpServletResponse.sendError()` Cross Site Scripting vulnerability was reported in Bugtraq ID 30496.

The `RequestDispatcher` Information Disclosure vulnerability was reported in Bugtraq ID 30494.

The UTF-8 Directory Traversal vulnerability was reported in Bugtraq ID 30633.

Apache Tomcat security fixes are reported by Apache in Apache Tomcat 3.x vulnerabilities, Apache Tomcat 4.x vulnerabilities, Apache Tomcat 5.x vulnerabilities, and Apache Tomcat 6.x vulnerabilities.

The WebDAV servlet arbitrary file access was reported in Secunia Advisory SA27398.

The quote issues and cross-site scripting vulnerabilities were reported in Secunia Advisory SA26465 and Secunia Advisory SA26466.

The Apache Tomcat security fixes were reported in Secunia Advisory SA24732.

Technical Details

Service: pcsync-https
<html><head><title>Apache Tomcat/4.1.31 - Error report</title><STYLE><!--H1{font-family : sans-serif,Arial,Tahoma;color : white;background-color : #0086b2;} H3{font-family : sans-serif,Arial,Tahoma;color : white;background-color : #0086b2;} BODY{font-family : sans-serif,Arial,Tahoma;color : black;background-color : white;} B{color : white;background-color : #0086b2;} HR{color : #0086b2;} --></STYLE> < /head><body><h1>HTTP Status 404 - /n0nexi5tent_fi1e.asp</h1><HR size="1" noshade="noshade"><p>type S

guessed password to web form: /dvwa/login.php (admin:password)

Severity: Critical Problem

Impact

An attacker who is able to guess the password to a user account could gain shell access to the system with the privileges of the user. From there it is often trivial to gain complete control of the system.

Resolution

Protect all accounts with a password that cannot be guessed. Require users to choose passwords which are eight characters long, including numeric and non-alphanumeric characters, and which are not based on the login name or any other personal information about the user. Enforce this policy using a utility such as npasswd in place of the default UNIX `passwd` program. Check the strength of all account passwords periodically using a password cracking utility such as Crack for Unix.

For Cisco 2700 Series Wireless Location Appliance, change the password or mitigate as described in cisco-air-20061013-wla.

Where can I read more about this?

Walter Belgers' paper, UNIX password security, is a good reference on strengthening passwords.

The Cisco 2700 Series WLA default password was described in cisco-sa-2006-1012-wla and Bugtraq ID 20490.

The IBM Totalstorage DS400 default password was posted to Full Disclosure.

Technical Details

Service: 80:TCP
Sent:
POST /dvwa/login.php HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/5.0
Content-length: 44
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
username=admin&password=password&Login=Login
Received:
HTTP/1.1 200 OK
Did Not Receive:
???<label for="pass">Password</label> <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

guessed password to web form: /dvwa/vulnerabilities/brute/../../login.php (admin:password)

Severity: Critical Problem

Impact

An attacker who is able to guess the password to a user account could gain shell access to the system with the privileges of the user. From there it is often trivial to gain complete control of the system.

Resolution

Protect all accounts with a password that cannot be guessed. Require users to choose passwords which are eight characters long, including numeric and non-alphanumeric characters, and which are not based on the login name or any other personal information about the user. Enforce this policy using a utility such as npasswd in place of the default UNIX `passwd` program. Check the strength of all account passwords periodically using a password cracking utility such as Crack for Unix.

For Cisco 2700 Series Wireless Location Appliance, change the password or mitigate as described in cisco-air-20061013-wla.

Where can I read more about this?

Walter Belgers' paper, UNIX password security, is a good reference on strengthening passwords.

The Cisco 2700 Series WLA default password was described in cisco-sa-2006-1012-wla and Bugtraq ID 20490.

The IBM Totalstorage DS400 default password was posted to Full Disclosure.

Technical Details

Service: 80:TCP
Sent:
POST /dvwa/vulnerabilities/brute/../../login.php HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/5.0
Content-length: 44
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
username=admin&password=password&Login=Login
Received:
HTTP/1.1 200 OK
Did Not Receive:
???<label for="pass">Password</label> <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

guessed password to web form: /dvwa/vulnerabilities/csrf/../../login.php (admin:password)

Severity: Critical Problem

Impact

An attacker who is able to guess the password to a user account could gain shell access to the system with the privileges of the user. From there it is often trivial to gain complete control of the system.

Resolution

Protect all accounts with a password that cannot be guessed. Require users to choose passwords which are eight characters long, including numeric and non-alphanumeric characters, and which are not based on the login name or any other personal information about the user. Enforce this policy using a utility such as npasswd in place of the default UNIX `passwd` program. Check the strength of all account passwords periodically using a password cracking utility such as Crack for Unix.

For Cisco 2700 Series Wireless Location Appliance, change the password or mitigate as described in cisco-air-20061013-wla.

Where can I read more about this?

Walter Belgers' paper, UNIX password security, is a good reference on strengthening passwords.

The Cisco 2700 Series WLA default password was described in cisco-sa-2006-1012-wla and Bugtraq ID 20490.

The IBM Totalstorage DS400 default password was posted to Full Disclosure.

Technical Details

Service: 80:TCP

Sent:
POST /dvwa/vulnerabilities/csrf/../../login.php HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/5.0
Content-length: 44
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
username=admin&password=password&Login=Login
Received:
HTTP/1.1 200 OK
Did Not Receive:
???<label for="pass">Password</label> <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

guessed password to web form: /dvwa/vulnerabilities/exec/../../login.php (admin:password)

Severity: Critical Problem

Impact

An attacker who is able to guess the password to a user account could gain shell access to the system with the privileges of the user. From there it is often trivial to gain complete control of the system.

Resolution

Protect all accounts with a password that cannot be guessed. Require users to choose passwords which are eight characters long, including numeric and non-alphanumeric characters, and which are not based on the login name or any other personal information about the user. Enforce this policy using a utility such as npasswd in place of the default UNIX `passwd` program. Check the strength of all account passwords periodically using a password cracking utility such as Crack for Unix.

For Cisco 2700 Series Wireless Location Appliance, change the password or mitigate as described in cisco-air-20061013-wla.

Where can I read more about this?

Walter Belgers' paper, UNIX password security, is a good reference on strengthening passwords.

The Cisco 2700 Series WLA default password was described in cisco-sa-2006-1012-wla and Bugtraq ID 20490.

The IBM Totalstorage DS400 default password was posted to Full Disclosure.

Technical Details

Service: 80:TCP
Sent:
POST /dvwa/vulnerabilities/exec/../../login.php HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/5.0
Content-length: 44
Content-Type: application/x-www-form-urlencoded

Connection: Keep-Alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
username=admin&password=password&Login=Login
Received:
HTTP/1.1 200 OK
Did Not Receive:
???<label for="pass">Password</label> <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

guessed password to web form: /dvwa/vulnerabilities/fi/../../login.php (admin:password)

Severity: Critical Problem

Impact

An attacker who is able to guess the password to a user account could gain shell access to the system with the privileges of the user. From there it is often trivial to gain complete control of the system.

Resolution

Protect all accounts with a password that cannot be guessed. Require users to choose passwords which are eight characters long, including numeric and non-alphanumeric characters, and which are not based on the login name or any other personal information about the user. Enforce this policy using a utility such as npasswd in place of the default UNIX `passwd` program. Check the strength of all account passwords periodically using a password cracking utility such as Crack for Unix.

For Cisco 2700 Series Wireless Location Appliance, change the password or mitigate as described in cisco-air-20061013-wla.

Where can I read more about this?

Walter Belgers' paper, UNIX password security, is a good reference on strengthening passwords.

The Cisco 2700 Series WLA default password was described in cisco-sa-2006-1012-wla and Bugtraq ID 20490.

The IBM Totalstorage DS400 default password was posted to Full Disclosure.

Technical Details

Service: 80:TCP
Sent:
POST /dvwa/vulnerabilities/fi/../../login.php HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/5.0
Content-length: 44
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
username=admin&password=password&Login=Login
Received:
HTTP/1.1 200 OK

Did Not Receive:
???<label for="pass">Password</label> <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

guessed password to web form: /dvwa/vulnerabilities/sqli/../../login.php (admin:password)

Severity: Critical Problem

Impact

An attacker who is able to guess the password to a user account could gain shell access to the system with the privileges of the user. From there it is often trivial to gain complete control of the system.

Resolution

Protect all accounts with a password that cannot be guessed. Require users to choose passwords which are eight characters long, including numeric and non-alphanumeric characters, and which are not based on the login name or any other personal information about the user. Enforce this policy using a utility such as npasswd in place of the default UNIX `passwd` program. Check the strength of all account passwords periodically using a password cracking utility such as Crack for Unix.

For Cisco 2700 Series Wireless Location Appliance, change the password or mitigate as described in cisco-air-20061013-wla.

Where can I read more about this?

Walter Belgers' paper, UNIX password security, is a good reference on strengthening passwords.

The Cisco 2700 Series WLA default password was described in cisco-sa-2006-1012-wla and Bugtraq ID 20490.

The IBM Totalstorage DS400 default password was posted to Full Disclosure.

Technical Details

Service: 80:TCP
Sent:
POST /dvwa/vulnerabilities/sqli/../../login.php HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/5.0
Content-length: 44
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
username=admin&password=password&Login=Login
Received:
HTTP/1.1 200 OK
Did Not Receive:
???<label for="pass">Password</label> <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

guessed password to web form: /dvwa/vulnerabilities/sqli_blind/../../login.php (admin:password)

Severity: Critical Problem

Impact

An attacker who is able to guess the password to a user account could gain shell access to the system with the privileges of the user. From there it is often trivial to gain complete control of the system.

Resolution

Protect all accounts with a password that cannot be guessed. Require users to choose passwords which are eight characters long, including numeric and non-alphanumeric characters, and which are not based on the login name or any other personal information about the user. Enforce this policy using a utility such as npasswd in place of the default UNIX `passwd` program. Check the strength of all account passwords periodically using a password cracking utility such as Crack for Unix.

For Cisco 2700 Series Wireless Location Appliance, change the password or mitigate as described in cisco-air-20061013-wla.

Where can I read more about this?

Walter Belgers' paper, UNIX password security, is a good reference on strengthening passwords.

The Cisco 2700 Series WLA default password was described in cisco-sa-2006-1012-wla and Bugtraq ID 20490.

The IBM Totalstorage DS400 default password was posted to Full Disclosure.

Technical Details

Service: 80:TCP
Sent:
POST /dvwa/vulnerabilities/sqli_blind/../../login.php HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/5.0
Content-length: 44
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
username=admin&password=password&Login=Login
Received:
HTTP/1.1 200 OK
Did Not Receive:
???<label for="pass">Password</label> <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

guessed password to web form: /dvwa/vulnerabilities/upload/../../login.php (admin:password)

Severity: Critical Problem

Impact

An attacker who is able to guess the password to a user account could gain shell access to the system with the privileges of the user. From there it is often trivial to gain complete control of the system.

Resolution

Protect all accounts with a password that cannot be guessed. Require users to choose passwords which are eight characters long, including numeric and non-alphanumeric characters, and which are not based on the login name or any other personal information about the user. Enforce this policy using a utility such as npasswd in place of the default UNIX `passwd` program. Check the strength of all account passwords periodically using a password cracking utility such as Crack for Unix.

For Cisco 2700 Series Wireless Location Appliance, change the password or mitigate as described in cisco-air-20061013-wla.

Where can I read more about this?

Walter Belgers' paper, UNIX password security, is a good reference on strengthening passwords.

The Cisco 2700 Series WLA default password was described in cisco-sa-2006-1012-wla and Bugtraq ID 20490.

The IBM Totalstorage DS400 default password was posted to Full Disclosure.

Technical Details

Service: 80:TCP
Sent:
POST /dvwa/vulnerabilities/upload/../../login.php HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/5.0
Content-length: 44
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
username=admin&password=password&Login=Login
Received:
HTTP/1.1 200 OK
Did Not Receive:
???<label for="pass">Password</label> <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

guessed password to web form: /dvwa/vulnerabilities/xss_r/../../login.php (admin:password)

Severity: Critical Problem

Impact

An attacker who is able to guess the password to a user account could gain shell access to the system with the privileges of the user. From there it is often trivial to gain complete control of the system.

Resolution

Protect all accounts with a password that cannot be guessed. Require users to choose passwords which are eight characters long, including numeric and non-alphanumeric characters, and which are not based on the login name or any other personal information about the user. Enforce this policy using a utility such as npasswd in

place of the default UNIX passwd program. Check the strength of all account passwords periodically using a password cracking utility such as Crack for Unix.

For Cisco 2700 Series Wireless Location Appliance, change the password or mitigate as described in cisco-air-20061013-wla.

Where can I read more about this?

Walter Belgers' paper, UNIX password security, is a good reference on strengthening passwords.

The Cisco 2700 Series WLA default password was described in cisco-sa-2006-1012-wla and Bugtraq ID 20490.

The IBM Totalstorage DS400 default password was posted to Full Disclosure.

Technical Details

Service: 80:TCP
Sent:
POST /dvwa/vulnerabilities/xss_r/../../login.php HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/5.0
Content-length: 44
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
username=admin&password=password&Login=Login
Received:
HTTP/1.1 200 OK
Did Not Receive:
???<label for="pass">Password</label> <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

guessed password to web form: /dvwa/vulnerabilities/xss_s/../../login.php (admin:password)

Severity: Critical Problem

Impact

An attacker who is able to guess the password to a user account could gain shell access to the system with the privileges of the user. From there it is often trivial to gain complete control of the system.

Resolution

Protect all accounts with a password that cannot be guessed. Require users to choose passwords which are eight characters long, including numeric and non-alphanumeric characters, and which are not based on the login name or any other personal information about the user. Enforce this policy using a utility such as npasswd in place of the default UNIX passwd program. Check the strength of all account passwords periodically using a password cracking utility such as Crack for Unix.

For Cisco 2700 Series Wireless Location Appliance, change the password or mitigate as described in cisco-air-20061013-wla.

Where can I read more about this?

Walter Belgers' paper, UNIX password security, is a good reference on strengthening passwords.

The Cisco 2700 Series WLA default password was described in cisco-sa-2006-1012-wla and Bugtraq ID 20490.

The IBM Totalstorage DS400 default password was posted to Full Disclosure.

Technical Details

Service: 80:TCP
Sent:
POST /dvwa/vulnerabilities/xss_s/../../login.php HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/5.0
Content-length: 44
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
username=admin&password=password&Login=Login
Received:
HTTP/1.1 200 OK
Did Not Receive:
???<label for="pass">Password</label> <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

Web server allows HTTP method DELETE

Severity: Critical Problem

Impact

The HTTP DELETE method may allow an attacker to delete arbitrary content from the Web Server.

Resolution

Disable the DELETE method in the Web Server configuration. If this is not an option, use one of the following workarounds:

- **Apache:** Disable the DELETE method by including the following in the Apache configuration:

```
<Limit DELETE>
Order Deny, Allow
Deny from All
</Limit>
```

- **IIS:** Use the UrlScan Security Tool or the IIS Lockdown Tool provided by Microsoft which allows you to specify which HTTP verbs are allowed or disallowed.

- **Java System Web Server, Sun ONE Web Server, iPlanet:** In the server.xml configuration file, add the following lines to restrict the DELETE method to a particular user(s):

```
acl "uri=/dir/*";
deny(all)
user="anyone";
allow(read, list, execute, info)
user="all";
allow (read, list, execute, info, write, delete)
user = "username";
```

Where can I read more about this?

Information on the HTTP Method DELETE vulnerability was reported in OSVDB 5646.

Technical Details

Service: 8080:TCP
Received:
Allow: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS

Web server allows HTTP method DELETE

Severity: Critical Problem

Impact

The HTTP DELETE method may allow an attacker to delete arbitrary content from the Web Server.

Resolution

Disable the DELETE method in the Web Server configuration. If this is not an option, use one of the following workarounds:

- **Apache:** Disable the DELETE method by including the following in the Apache configuration:

```
<Limit DELETE>
Order Deny, Allow
Deny from All
</Limit>
```

- **IIS:** Use the UrlScan Security Tool or the IIS Lockdown Tool provided by Microsoft which allows you to specify which HTTP verbs are allowed or disallowed.

- **Java System Web Server, Sun ONE Web Server, iPlanet:** In the server.xml configuration file, add the following lines to restrict the DELETE method to a particular user(s):

```
acl "uri=/dir/*";
deny(all)
user="anyone";
allow(read, list, execute, info)
user="all";
allow (read, list, execute, info, write, delete)
user = "username";
```

Where can I read more about this?

Information on the HTTP Method DELETE vulnerability was reported in OSVDB 5646.

Technical Details

Service: pcsync-https
Received:
Allow: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS

Web server allows PUT: /

Severity: Critical Problem

Impact

An attacker may be able to upload files onto the web server.

Resolution

Configure the web server not to accept **PUT** requests. If you require the functionality of **PUT** for web publishing, use a put script which can only be run by authorized users, which ensures that the script can update only web content files, and which ensures that users can only update their own pages.

Where can I read more about this?

A good tutorial on the proper use of the **PUT** method is available from Apache Week.

Technical Details

Service: pcsync-https
Received:
Allow: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS

Web server allows PUT: /

Severity: Critical Problem

Impact

An attacker may be able to upload files onto the web server.

Resolution

Configure the web server not to accept **PUT** requests. If you require the functionality of **PUT** for web publishing, use a put script which can only be run by authorized users, which ensures that the script can update only web content files, and which ensures that users can only update their own pages.

Where can I read more about this?

A good tutorial on the proper use of the **PUT** method is available from Apache Week.

Technical Details

Service: 8080:TCP
Received:

Allow: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS

vulnerable PHP version: 5.3.2

Severity: Critical Problem

CVE: CVE-2006-7243 CVE-2007-1581
CVE-2010-1860 CVE-2010-1861
CVE-2010-1862 CVE-2010-1864
CVE-2010-1866 CVE-2010-1868
CVE-2010-1914 CVE-2010-1915
CVE-2010-1917 CVE-2010-2094
CVE-2010-2097 CVE-2010-2100
CVE-2010-2101 CVE-2010-2190
CVE-2010-2191 CVE-2010-2225
CVE-2010-2531 CVE-2010-3065
CVE-2010-3436 CVE-2010-3709
CVE-2010-3710 CVE-2010-3870
CVE-2010-4150 CVE-2010-4156
CVE-2010-4409 CVE-2010-4645
CVE-2010-4697 CVE-2011-0708
CVE-2011-0753 CVE-2011-0754
CVE-2011-0755 CVE-2011-1092
CVE-2011-1148 CVE-2011-1398
CVE-2011-1464 CVE-2011-1466
CVE-2011-1467 CVE-2011-1468
CVE-2011-1469 CVE-2011-1470
CVE-2011-1471 CVE-2011-1657
CVE-2011-1938 CVE-2011-2202
CVE-2011-2483 CVE-2011-3182
CVE-2011-3267 CVE-2011-3268
CVE-2011-4718 CVE-2011-4885
CVE-2012-0057 CVE-2012-0788
CVE-2012-0789 CVE-2012-1823
CVE-2012-2311 CVE-2012-2688
CVE-2012-3365 CVE-2012-3450
CVE-2013-1635 CVE-2013-1643
CVE-2013-2110 CVE-2013-4113
CVE-2013-4248 CVE-2013-4636
CVE-2013-6420 CVE-2014-0207
CVE-2014-3478 CVE-2014-3479
CVE-2014-3480 CVE-2014-3487
CVE-2014-3515 CVE-2014-3668
CVE-2014-3669 CVE-2014-3670
CVE-2014-3710 CVE-2014-3981
CVE-2014-4049 CVE-2014-4670
CVE-2014-4698 CVE-2014-8142
CVE-2014-9426 CVE-2014-9427
CVE-2014-9709 CVE-2015-0231
CVE-2015-0232 CVE-2015-0273
CVE-2015-1351 CVE-2015-1352
CVE-2015-2301 CVE-2015-2348
CVE-2015-2783 CVE-2015-2787
CVE-2015-3152 CVE-2015-3307
CVE-2015-3329 CVE-2015-4021
CVE-2015-4022 CVE-2015-4024

	CVE-2015-4025 CVE-2015-4026 CVE-2015-4147 CVE-2015-4148 CVE-2015-4598 CVE-2015-4642 CVE-2015-4643 CVE-2015-4644 CVE-2015-5589 CVE-2015-5590 CVE-2015-6834 CVE-2015-6835 CVE-2015-6836 CVE-2015-6837 CVE-2015-6838 CVE-2015-7803 CVE-2015-7804

Impact

Remote attackers may be able to gain unauthorized access to the web server, cause a denial of service or information disclosure, or execute arbitrary code.

Resolution

PHP should be upgraded to version 5.5.30 for 5.5.x, or 5.6.14 for 5.6.x, or higher than 5.4.45 for 5.4.x when available, or 7.0 Beta 2 dev for development.

Where can I read more about this?

The two vulnerabilities fixed in 5.5.30 and 5.6.14 were reported in Version 5.6.14.

The multiple vulnerabilities fixed in 5.4.45, 5.5.29, and 5.6.13 were reported in Version 5.6.13.

The multiple vulnerabilities fixed in 5.4.43, 5.5.27, and 5.6.11 were reported in Version 5.6.11, Bugtraq ID 75974, and Bugtraq ID 75970.

The multiple vulnerabilities fixed in 5.4.42, 5.5.26, and 5.6.10 were reported in Version 5.6.10.

The PHP `SoapClient's` Type Confusion Flaws were reported in PHP Sec Bug 69085.

The multiple vulnerabilities fixed in 5.4.41, 5.5.25, and 5.6.9 were reported in Version 5.6.9.

The `phar_parse_tarfile` memory corruption vulnerability was reported in OSVDB 122125.

The multiple vulnerabilities fixed in PHP 5.4.40, 5.5.24, and 5.6.8 were reported in PHP released 2015-04-16-3.

The **`move_uploaded_file`** access control bypass and the **`unserialize`** use-after-free vulnerability were reported in PHP 5.6.7 ChangeLog.

The PHP `DateTimeZone` vulnerability was reported in PHP 5.4.38 ChangeLog, PHP 5.5.22 ChangeLog, and PHP 5.6.6 ChangeLog.

The Use-after-free in `opcache` and NULL Pointer Dereference in pgsql were reported in OSVDB 117589 and OSVDB 117588.

The multiple vulnerabilities fixed in PHP 5.4.37, 5.5.21, and 5.6.5 were reported in PHP 5.4.37 ChangeLog, PHP 5.5.21 ChangeLog, and PHP 5.6.5 ChangeLog.

The Out of bounds read in `sapi/cgi/cgi_main.c` was reported in PHP Sec Bug 68618.

The PHP fileinfo `apprentice_load` function invalid free vulnerability was reported in OSVDB 116500.

The use-after-free remote code execution vulnerability was reported in PHP 5.4.36 ChangeLog, PHP 5.5.20 ChangeLog, and PHP 5.6.4 ChangeLog.

The PHP fileinfo in elf note headers vulnerability was reported in PHP 5.4.35 ChangeLog and PHP 5.5.19 ChangeLog.

The multiple vulnerabilities fixed in PHP 5.4.34, 5.5.18, and 5.6.2 were reported in PHP 5.6.2 ChangeLog, PHP 5.5.18 ChangeLog, and PHP 5.4.34 ChangeLog.

The PHP Fileinfo denial of service vulnerability was reported in OSVDB 94064.

The local denial-of-service vulnerabilities were reported in OSVDB 108946 and OSVDB 108947.

The multiple vulnerabilities fixed in PHP 5.4.30 and 5.5.14 reported in PHP 5.4.30 ChangeLog and PHP 5.5.14ChangeLog.

The heap-based buffer overflow in `DNS TXT` record parsing was reported in Red Hat Bugzilla Bug 1108447.

The PHP OpenSSL Module Buffer Overflow Vulnerability was reported in Secunia Advisory SA56055.

The Session fixation vulnerability was reported in Secunia Advisory SA54562.

The SSL Module "`subjectAltNames`" NULL Byte Handling vulnerability was reported in Secunia Advisory SA54480.

The XML Parsing Buffer Overflow vulnerability was reported in Secunia Advisory SA54069.

The "`php_quot_print_encode()`" Buffer Overflow vulnerability was reported in Secunia Advisory SA53736.

The vulnerability fixed in 5.3.4 was reported in Bugtraq ID 44951.

The two vulnerabilities fixed in PHP 5.3.22 and 5.4.13 were reported in Secunia CVE Reference CVE-2013-1635 and Secunia CVE Reference CVE-2013-1643.

The "`header()`" HTTP header injection vulnerability was reported in Bugtraq ID 55297.

The PHP 5.3.15 and 5.4.5 fixed potential overflow was reported in Bugtraq ID 54638.

The PHP 5.3.14 and 5.4.4 fixed denial of service was reported in Bugtraq ID 54777.

The "open_basedir" security bypass vulnerability was reported in Secunia Advisory SA49969.

The NULL pointer dereference and `FILTER_VALIDATE_EMAIL` vulnerabilities were reported in Bugtraq ID 44718, Bugtraq ID 43926, and PHP 5.3.4 Released.

The "`PHP-CGI`" query string parameter vulnerability was reported in Secunia Advisory SA49014.

The Web Form Hash Collision Denial of Service vulnerability was reported in Secunia Advisory SA47404.

The "`is_a()`" Change in Functional Behaviour Security vulnerability was reported in Secunia Advisory

SA46107.

The multiple vulnerabilities fixed in PHP 5.3.7 were reported in Secunia Advisory SA44874.

The `phar` Extension Integer Overflow vulnerability was reported in Secunia Advisory SA44335.

The `Exif` Extension `'exif_read_data()'` Function Remote Denial of Service vulnerability was reported in Bugtraq ID 46365.

The vulnerabilities fixed in 5.3.4 were reported in Bugtraq ID 46168.

The `Zend` Engine Use-after-free Heap Corruption vulnerability was reported in Bugtraq ID 45952.

The `zend_strtod()` Function Floating-Point Value Denial of Service vulnerability was reported in Bugtraq ID 45668.

The `'getSymbol()'` Function Denial of Service vulnerability was reported in Bugtraq ID 45119.

The `'ext/imap/php_imap.c'` Use After Free Denial of Service vulnerability was reported in Bugtraq ID 44980.

The NULL Character Security Bypass vulnerability was reported in net-security 14385.

The `'mb_strcut()'` Function Information Disclosure vulnerability was reported in Bugtraq ID 44727.

The `'open_basedir'` Security-Bypass vulnerability was reported in Bugtraq ID 44723.

The `'xml_utf8_decode()'` `UTF-8` Input Validation vulnerability was reported in Bugtraq ID 44605.

The `ibase_gen_id()` Function off-by-one Buffer Overflow vulnerability was reported in Bugtraq ID 42516.

The vulnerabilities fixed in 5.2.14 were reported in PHP 5 ChangeLog. The PHP project confirms that CVE-2010-2531 was fixed in PHP 5.3.3 as well as PHP 5.2.14.

The `strrchr()` Function Information Disclosure vulnerability was reported in Bugtraq ID 41265.

The `SplObjectStorage` Unserializer Arbitrary Code Execution vulnerability was reported in Bugtraq ID 40948.

The `Mysqlnd` Extension Information Disclosure and Multiple Buffer Overflow vulnerabilities were reported in Bugtraq ID 40461.

The `ext/phar/stream.c` and `ext/phar/dirstream.c` Multiple Format String vulnerabilities were reported in Bugtraq ID 40173.

The vulnerabilities in PHP 5.2.13 and 5.3.2 were reported in MOPS-2010-009, OpenSuSE summary report SR:2010:17 and OpenSuSE summary report SR:2010:18.

The `'sqlite_single_query()'` and `'sqlite_array_query()'` Arbitrary Code Execution vulnerabilities were reported in Bugtraq ID 40013.

The `php_dechunk()` HTTP Chunked Encoding Integer Overflow vulnerability was reported in Bugtraq ID 39877.

The Multiple Month of PHP Bugs vulnerabilities were reported in the month of PHP bugs.

Technical Details

Service: http
Sent: GET /dvwa/index.php HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/4.0
Received: X-Powered-By: PHP/5.3.2-1ubuntu4.15

vulnerable Apache version: 2.2.14

Severity: Area of Concern

CVE: CVE-2009-3560 CVE-2009-3720 CVE-2010-0425 CVE-2010-0434 CVE-2010-1452 CVE-2010-1623 CVE-2010-2068 CVE-2011-0419 CVE-2011-1928 CVE-2011-3192 CVE-2011-3348 CVE-2011-3607 CVE-2011-4415 CVE-2012-0031 CVE-2012-0053 CVE-2012-0883 CVE-2012-2687 CVE-2012-3499 CVE-2012-4558 CVE-2013-1862 CVE-2013-1896 CVE-2013-5704 CVE-2013-6438 CVE-2014-0098 CVE-2014-3581 CVE-2014-3583 CVE-2014-8109

Impact

A remote attacker could crash the web server, disclose certain sensitive information, or execute arbitrary commands.

Resolutions

Upgrade Apache 2.4.x to 2.4.16 or higher when available. 2.2.x to 2.2.29 or higher when available, or install an updated package from your Linux vendor.

Note: Apache httpd 2.4.1, includes fixes for all vulnerabilities which have been resolved in Apache httpd 2.2.22 and all older releases. Apache 2.0.x is no longer maintained.

Where can I read more about this?

The multiple vulnerabilities fixed in Apache HTTP Server 2.4.12 were reported in Apache HTTP Server 2.4.12 Released.

The Apache HTTP Server `mod_proxy_fcgi` response handling vulnerability was reported in OSVDB 114570.

The Apache HTTP Server NULL pointer dereference vulnerability was reported in OSVDB 112168.

The Two Apache HTTP Server Denial of Service Vulnerabilities were reported in Secunia Advisory SA37537.

The Apache HTTP Server two denial of service vulnerabilities were reported in Secunia Advisory SA57399.

The `mod_dav` and `mod_session_dbd` vulnerabilities were reported in Secunia Advisory SA54241.

The `mod_rewrite` vulnerability was reported in Secunia Advisory SA53154.

The multiple Cross-Site Scripting vulnerabilities fixed in 2.2.24 were reported in Secunia Advisory SA52394.

The Information Disclosure and Cross-Site Scripting vulnerabilities were reported in Secunia Advisory SA50363.

The `LD_LIBRARY_PATH` vulnerability was reported in Secunia Advisory SA48849.

The `"httpOnly"` Cookie Disclosure and Denial of Service vulnerabilities were reported in Secunia Advisory SA47779.

The Scoreboard Invalid Free Security Bypass vulnerability was reported in Secunia Advisory SA47410.

The `"ap_pregsub()"` Denial of Service vulnerability was reported in Secunia Advisory SA46823.

The `"ap_pregsub()"` Privilege Escalation vulnerability was reported in Secunia Advisory SA45793.

The `mod_proxy_ajp` Denial of Service vulnerability was reported in Secunia Advisory SA46013.

The `ByteRange` Filter Denial of Service vulnerability was reported in Secunia Advisory SA45606.

The `APR "apr_fnmatch()"` Infinite Loop Denial of Service vulnerability was reported in Secunia Advisory SA44661.

The `APR apr_fnmatch` Denial of Service vulnerability was reported in Secunia Advisory SA44574.

The `APR apr_brigade_split_line` Denial of Service vulnerability was reported in Bugtraq ID 43673.

The multiple Remote Denial of Service vulnerabilities were reported in Bugtraq ID 41963.

The `Apache` Memory Corruption Vulnerability was reported in Bugtraq ID: 38494. The `mod_proxy_http` Timeout Handling Information Disclosure vulnerability was reported in Bugtraq ID 40827.

The `mod_isapi` Dangling Pointer Remote Code Execution vulnerability was reported in Bugtraq ID 38494.

The HTTP-Basic Authentication Bypass vulnerability was reported in Bugtraq ID 35840.

The Apache HTTP Server OS Fingerprinting Unspecified Security vulnerability was reported in Bugtraq ID 31805.

Technical Details

Service: http
Received: Server: Apache/2.2.14 (Ubuntu)

.htaccess file is accessible

Severity: Area of Concern

Impact

A remote attacker may be able to enumerate users or view sensitive configuration information.

Resolution

Most web servers which support **`.htaccess`** files prevent them from being viewed over the web, so if this vulnerability was detected, then either the web server does not support **`.htaccess`** files and they should be removed, or the web server is misconfigured. Check your web server software's documentation for more information.

Where can I read more about this?

For more information on **`.htaccess`** files, see the Htaccess FAQ.

Technical Details

Service: hbci
Sent:
GET /.htaccess HTTP/1.0
Host: dojo1.sainttest.local:3000
User-Agent: Mozilla/4.0
Connection: Keep-alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; security=high
Received:
Options +FollowSymLinks +ExecCGI

Web Server Internal IP address or network name available

Severity: Area of Concern **CVE:** CVE-2000-0649 CVE-2002-0419

Impact

An attacker could determine information about your internal network structure from information in http headers.

Resolutions

For IIS 4.0, 5.0 or 5.1, fix as designated in Q218180 by changing the he w3svc/UseHostName value in the metabase from False to True.

For IIS 6.0, fix as designated in 834141.

For other web servers, contact the vendor.

Where can I read more about this?

More information on the Web Server Internal IP address available for the IIS web servers is available at Bugtraq ID 1499.

Technical Details

Service: 8080:TCP
Received:

Location: http://dojo1.local:8080/index.html

Web Server Internal IP address or network name available

Severity: Area of Concern **CVE:** CVE-2000-0649 CVE-2002-0419

Impact

An attacker could determine information about your internal network structure from information in http headers.

Resolutions

For IIS 4.0, 5.0 or 5.1, fix as designated in Q218180 by changing the he w3svc/UseHostName value in the metabase from False to True.

For IIS 6.0, fix as designated in 834141.

For other web servers, contact the vendor.

Where can I read more about this?

More information on the Web Server Internal IP address available for the IIS web servers is available at Bugtraq ID 1499.

Technical Details

Service: pcsync-https
Received:
Location: https://dojo1.local:8443/index.html

Web error message information leakage: /dvwa/instructions.php

Severity: Area of Concern

Impact

A remote attacker could view information about the internal workings of the web application.

Resolution

Modify the web server or application to disable detailed error messages.

Where can I read more about this?

More information about information leakage is available from the Web Application Security Consortium.

Technical Details

Service: 80:TCP
Sent:
GET /dvwa
/instructions.php?n0nex1st=%3Cscript%3Ealert%28%22SAINTL2R2d2EvaW5zdHJ1Y3Rpb25zLnBocCBuMG5leDFzdA==%22%29%3C%2Fscript%3E HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/5.0

Connection: Keep-Alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
Received:
error while trying to create your database, make sure your database credentials are correct within /config /config.inc.php

Web error message information leakage: /dvwa/setup.php

Severity: Area of Concern

Impact

A remote attacker could view information about the internal workings of the web application.

Resolution

Modify the web server or application to disable detailed error messages.

Where can I read more about this?

More information about information leakage is available from the Web Application Security Consortium.

Technical Details

Service: http
Sent:
GET /dvwa/setup.php HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/4.0
Received:
error make sure you have the correct user credentials in /config/config.inc.php</p>

Web error message information leakage: /dvwa/setup.php

Severity: Area of Concern

Impact

A remote attacker could view information about the internal workings of the web application.

Resolution

Modify the web server or application to disable detailed error messages.

Where can I read more about this?

More information about information leakage is available from the Web Application Security Consortium.

Technical Details

Service: 80:TCP
Sent:
POST /dvwa/setup.php HTTP/1.0
Host: dojo1.sainttest.local

User-Agent: Mozilla/5.0
Content-length: 35
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
create_db=Create+%2F+Reset+Database
Received:
error make sure you have the correct user credentials in /config/config.inc.php</p>

Apache ETag header discloses inode numbers

Severity: Potential Problem **CVE:** CVE-2003-1418

Impact

A remote attacker could determine inode numbers on the server.

Resolution

Use the `FileETag` directive to remove the INode component from the calculation of the ETag. For example, place the following line in the Apache configuration file to calculate the ETag based only on the file's modification time and size:

```
FileETag MTime Size
```

Where can I read more about this?

This vulnerability was reported in Bugtraq ID 6939.

Technical Details

Service: http
Sent:
GET /dvwa/dvwa/js/dvwaPage.js HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/5.0
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
Received:
ETag: "4008b-307-4a6d8f85f8940"

Apache mod_proxy_ajp 2.2.14 is vulnerable

Severity: Potential Problem **CVE:** CVE-2010-0408

Impact

A remote attacker may be able to crash the Apache process or execute arbitrary commands.

Resolutions

To resolve the vulnerability in mod_proxy_ajp, upgrade to Apache version 2.2.15 or higher when available.

To resolve the vulnerabilities in mod_ntlm, upgrade to version 0.5 or higher for Apache 1.3 or version 0.2 or

higher for Apache 2.0. These versions will presumably contain a fix. If these versions are not yet available, it would be advisable to disable mod_ntlm in the Apache configuration file, and use Basic HTTP authentication instead of NTLM.

Where can I read more about this?

The `mod_proxy_ajp` vulnerability was reported in Bugtraq ID 38491.

The vulnerabilities in mod_ntlm were posted to Bugtraq archive 319239.

Technical Details

Service: http
Received:
Server: Apache/2.2.14 (Ubuntu)

mod_proxy vulnerability in Apache version: 2.2.14

Severity: Potential Problem **CVE:** CVE-2011-3368 CVE-2011-3639 CVE-2011-4317

Impact

A remote attacker may be able to crash the Apache process or execute arbitrary commands.

Resolutions

To resolve the vulnerabilities in mod_proxy, upgrade to version 2.3.3 or higher for Apache 2.3, a version higher than 2.2.21 for Apache 2.2 when available, or a version higher than 2.0.64 for Apache 2.0 when available.

To resolve the vulnerabilities in mod_ntlm, upgrade to version 0.5 or higher for Apache 1.3 or version 0.2 or higher for Apache 2.0. These versions will presumably contain a fix. If these versions are not yet available, it would be advisable to disable mod_ntlm in the Apache configuration file, and use Basic HTTP authentication instead of NTLM.

Where can I read more about this?

The `mod_proxy` Reverse Proxy Mode Security Bypass vulnerability was reported in Secunia Advisory SA46288 and Secunia Advisory SA46987.

The vulnerabilities in mod_ntlm were posted to Bugtraq archive 319239.

Technical Details

Service: http
Received:
Server: Apache/2.2.14 (Ubuntu)

Web site may be vulnerable to clickjacking attacks

Severity: Potential Problem

Impact

An attacker could trick a legitimate user into taking undesired actions on the web site.

Resolution

Prevent unauthorized sites from using your web pages in iframes by configuring the web server to send the X-Frame-Options response header. Mozilla has provided specific instructions for common web server software.

To protect against clickjacking in older browsers which don't support the X-Frame-Options header, various javascript defenses have been suggested, as described in OWASP's Clickjacking Defense Cheat Sheet.

Where can I read more about this?

More information about clickjacking is available from OWASP.

Technical Details

Service: http
Site allows authentication and has no X-Frame-Options header
Received: <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

Web site may be vulnerable to clickjacking attacks

Severity: Potential Problem

Impact

An attacker could trick a legitimate user into taking undesired actions on the web site.

Resolution

Prevent unauthorized sites from using your web pages in iframes by configuring the web server to send the X-Frame-Options response header. Mozilla has provided specific instructions for common web server software.

To protect against clickjacking in older browsers which don't support the X-Frame-Options header, various javascript defenses have been suggested, as described in OWASP's Clickjacking Defense Cheat Sheet.

Where can I read more about this?

More information about clickjacking is available from OWASP.

Technical Details

Service: hbci
Site allows authentication and has no X-Frame-Options header
Received: <input type="password" name="user_password" id="user_password" size="30"/>

Potentially sensitive information found on web page (/dvwa/)

Severity: Potential Problem

Impact

Sensitive information may be at risk of exposure.

Resolution

Consider encrypting sensitive information.

Technical Details

Service: http
Matches:
Payment card number: 4465-4204-9361-1370
[USA] Social Security Number: 324-45-0987
[USA] Social Security Number: 324-45-0999
[USA] Social Security Number: 295-51-0900

Potentially sensitive information found on web page (/dvwa/.)

Severity: Potential Problem

Impact

Sensitive information may be at risk of exposure.

Resolution

Consider encrypting sensitive information.

Technical Details

Service: http
Matches:
Payment card number: 4465-4204-9361-1370
[USA] Social Security Number: 324-45-0987
[USA] Social Security Number: 324-45-0999
[USA] Social Security Number: 295-51-0900

Potentially sensitive information found on web page (/dvwa/index.php)

Severity: Potential Problem

Impact

Sensitive information may be at risk of exposure.

Resolution

Consider encrypting sensitive information.

Technical Details

Service: http
Matches:
Payment card number: 4465-4204-9361-1370
[USA] Social Security Number: 324-45-0987
[USA] Social Security Number: 324-45-0999
[USA] Social Security Number: 295-51-0900

weak https cache policy

Severity: Potential Problem

Impact

The confidentiality provided by **`https`** sessions could be compromised due to stored copies of sensitive pages in a shared cache or browser cache.

Resolution

Set the **`Cache-Control`** header to one or more of the following values:

- **private**: allows caching in the browser, but not shared caches
- **no-cache**: forces the cache to re-validate the authenticated session with the server before delivering a cached page
- **no-store**: prohibits the storing of cached pages

Setting **`Cache-Control`** to **`no-cache, no-store`** provides the greatest protection.

The **`Cache-Control`** header can be set programmatically using PHP's **`header()`** function, Java's **`HttpServletResponse.addHeader()`** method, or ASP's **`Response.AddHeader()`** method.

The **`Cache-Control`** header can also be set in the web server's configuration as follows:

- **Apache**:
  Add the following directive to the configuration file:

  ```
  Header set Cache-Control "no-cache, no-store"
  ```

- **IIS**:
  In IIS Manager, navigate to the desired level. Go to Features View -> HTTP Response Headers -> Actions pane. Click Add. In the Add Custom HTTP Response Header dialog box, enter **`Cache-Control`** in the *Name* box, and **`no-cache, no-store`** in the *Value* box.

It is also a good idea to set an **`Expires`** header along with the **`Cache-Control`** header for browsers and proxies which don't yet support HTTP/1.1. **`Expires`** should be set to a date in the past or an invalid date to prevent caching. For example, **`Sat, 31 May 2014 08:00:00 GMT`**.

Where can I read more about this?

For more information, see the OWASP Application Security FAQ and Mark Nottingham's Caching Tutorial.

Technical Details

Service: pcsync-https
Sent:
GET / HTTP/1.0
Host: dojo1.sainttest.local:8443
User-Agent: Mozilla/5.0
Received:
(no Cache-Control header)

CGI Gives Information about System (phpinfo.php)

Severity: Potential Problem

Impact

If a malicious user is able to exploit this vulnerability, he or she may be able acquire information about the web server and system settings of the exploited system. In certain circumstances, an attacker may even be able to gather information about user accounts on the affected system. A malicious user may then be able to gain unauthorized access to the system using this information. (Remember, an attacker's best weapon is knowledge!) For example, if an attacker is able to learn information about the operating system of the target, he or she will then be able to gear certain attacks (such as buffer overflows) towards that specific operating system.

Resolutions

phpinfo.php, info.php:
Delete the script from the server.

For Mambo Site Server 4.0.11 and earlier, download and install all security patches for versions 3.0.7 through 4.0.11, or upgrade to the 4.0.12 stable release when it becomes available.

Where can I read more about this?

More information on **`phpinfo.php`** and other vulnerabilities in Mambo Site Server can be found in Bugtraq archive 303137.

Technical Details

Service: http
Sent:
GET /dvwa/phpinfo.php HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/4.0
Connection: Keep-alive
Cookie: PHPSESSID=rtkje65i3sa9b8um2m9trb5r71; _session_id=5aefc39da276462c483aaae286b72889; security=high
Received:
<title>phpinfo()</title><meta name="ROBOTS" content="NOINDEX,NOFOLLOW,NOARCHIVE" /></head>

web server allows MIME sniffing

Severity: Potential Problem

Impact

An attacker may be able to cause arbitrary script to run in a user's browser in the context of the vulnerable site.

Resolution

All HTTP responses should include an accurate **`Content-Type`** header, and an **`X-Content-Type-Options: nosniff`** header. The latter header instructs browsers always to use the specified content type instead of performing MIME sniffing, and is currently supported by Internet Explorer and Chrome.

The **`X-Content-Type-Options: nosniff`** header can be set in the web server's configuration as follows:

- **Apache**:
  Add the following directive to the configuration file:

```
Header set X-Content-Type-Options "nosniff"
```

- **IIS**:
  In IIS Manager, navigate to the desired level. Go to Features View -> HTTP Response Headers -> Actions pane. Click Add. In the Add Custom HTTP Response Header dialog box, enter **`X-Content-Type-Options`** in the *Name* box, and **`nosniff`** in the *Value* box.

Where can I read more about this?

For more information about MIME-sniffing risks and defenses, see Wikipedia and IE8 Security Part V. (Scroll down to the *MIME-Handling Changes* section.)

Technical Details

Service: hbci
Sent:
GET / HTTP/1.0
Host: dojo1.sainttest.local:3000
User-Agent: Mozilla/5.0
Received:
Missing Content-Type header or X-Content-Type-Options header not set to nosniff

web server allows MIME sniffing

Severity: Potential Problem

Impact

An attacker may be able to cause arbitrary script to run in a user's browser in the context of the vulnerable site.

Resolution

All HTTP responses should include an accurate **`Content-Type`** header, and an **`X-Content-Type-Options: nosniff`** header. The latter header instructs browsers always to use the specified content type instead of performing MIME sniffing, and is currently supported by Internet Explorer and Chrome.

The **`X-Content-Type-Options: nosniff`** header can be set in the web server's configuration as follows:

- **Apache**:
  Add the following directive to the configuration file:

```
Header set X-Content-Type-Options "nosniff"
```

- **IIS**:
  In IIS Manager, navigate to the desired level. Go to Features View -> HTTP Response Headers -> Actions pane. Click Add. In the Add Custom HTTP Response Header dialog box, enter

`X-Content-Type-Options` in the *Name* box, and **`nosniff`** in the *Value* box.

Where can I read more about this?

For more information about MIME-sniffing risks and defenses, see Wikipedia and IE8 Security Part V. (Scroll down to the *MIME-Handling Changes* section.)

Technical Details

Service: 8080:TCP
Sent:
GET /index.html HTTP/1.0
Host: dojo1.sainttest.local:8080
User-Agent: Mozilla/5.0
Received:
Missing Content-Type header or X-Content-Type-Options header not set to nosniff

web server allows MIME sniffing

Severity: Potential Problem

Impact

An attacker may be able to cause arbitrary script to run in a user's browser in the context of the vulnerable site.

Resolution

All HTTP responses should include an accurate **`Content-Type`** header, and an **`X-Content-Type-Options: nosniff`** header. The latter header instructs browsers always to use the specified content type instead of performing MIME sniffing, and is currently supported by Internet Explorer and Chrome.

The **`X-Content-Type-Options: nosniff`** header can be set in the web server's configuration as follows:

- **Apache**:
  Add the following directive to the configuration file:

  ```
  Header set X-Content-Type-Options "nosniff"
  ```

- **IIS**:
  In IIS Manager, navigate to the desired level. Go to Features View -> HTTP Response Headers -> Actions pane. Click Add. In the Add Custom HTTP Response Header dialog box, enter **`X-Content-Type-Options`** in the *Name* box, and **`nosniff`** in the *Value* box.

Where can I read more about this?

For more information about MIME-sniffing risks and defenses, see Wikipedia and IE8 Security Part V. (Scroll down to the *MIME-Handling Changes* section.)

Technical Details

Service: http
Sent:

GET /dvwa/dvwa/js/dvwaPage.js HTTP/1.0
Host: dojo1.sainttest.local
User-Agent: Mozilla/5.0
Received:
Missing Content-Type header or X-Content-Type-Options header not set to nosniff

web server allows MIME sniffing

Severity: Potential Problem

Impact

An attacker may be able to cause arbitrary script to run in a user's browser in the context of the vulnerable site.

Resolution

All HTTP responses should include an accurate **`Content-Type`** header, and an **`X-Content-Type-Options: nosniff`** header. The latter header instructs browsers always to use the specified content type instead of performing MIME sniffing, and is currently supported by Internet Explorer and Chrome.

The **`X-Content-Type-Options: nosniff`** header can be set in the web server's configuration as follows:

- **Apache**:
  Add the following directive to the configuration file:

  ```
  Header set X-Content-Type-Options "nosniff"
  ```

- **IIS**:
  In IIS Manager, navigate to the desired level. Go to Features View -> HTTP Response Headers -> Actions pane. Click Add. In the Add Custom HTTP Response Header dialog box, enter **`X-Content-Type-Options`** in the *Name* box, and **`nosniff`** in the *Value* box.

Where can I read more about this?

For more information about MIME-sniffing risks and defenses, see Wikipedia and IE8 Security Part V. (Scroll down to the *MIME-Handling Changes* section.)

Technical Details

Service: pcsync-https
Sent:
GET /index.html HTTP/1.0
Host: dojo1.sainttest.local:8443
User-Agent: Mozilla/5.0
Received:
Missing Content-Type header or X-Content-Type-Options header not set to nosniff

web server autoindex enabled

Severity: Potential Problem **CVE:** CVE-1999-0569

Impact

A remote attacker could view the directory structure on the web server.

Resolutions

Ensure that autoindexing is not enabled on the web server. On Apache web servers, this can be done with the following directive in the configuration file:

```
Options -Indexes
```

Where can I read more about this?

For more information, see the Apache mod_autoindex documentation.

Technical Details

Service: pcsync-https
Received:
<title>Directory Listing For /Read-Me/</title>
Received:
<title>Directory Listing For /Read-Me/source/</title>

web server autoindex enabled

Severity: Potential Problem **CVE:** CVE-1999-0569

Impact

A remote attacker could view the directory structure on the web server.

Resolutions

Ensure that autoindexing is not enabled on the web server. On Apache web servers, this can be done with the following directive in the configuration file:

```
Options -Indexes
```

Where can I read more about this?

For more information, see the Apache mod_autoindex documentation.

Technical Details

Service: http
Received:
<title>Index of /dvwa/dvwa/css</title>
Received:
<title>Index of /dvwa/dvwa/js</title>
Received:
<title>Index of /dvwa/dvwa</title>
Received:
<title>Index of /dvwa/dvwa/images</title>
Received:
<title>Index of /dvwa/dvwa/includes</title>

web server autoindex enabled

Severity: Potential Problem **CVE:** CVE-1999-0569

Impact

A remote attacker could view the directory structure on the web server.

Resolutions

Ensure that autoindexing is not enabled on the web server. On Apache web servers, this can be done with the following directive in the configuration file:

```
Options -Indexes
```

Where can I read more about this?

For more information, see the Apache mod_autoindex documentation.

Technical Details

Service: 8080:TCP
Received:
<title>Directory Listing For /Read-Me/</title>
Received:
<title>Directory Listing For /Read-Me/source/</title>

Cookie without HTTPOnly attribute can be accessed by scripts: PHPSESSID

Severity: Potential Problem

Impact

A cookie without the HTTPOnly attribute could be susceptible to theft by cross-site scripting attacks.

Resolution

Modify web applications to set the HTTPOnly attribute for all cookies, or apply a patch or upgrade from your vendor.

Where can I read more about this?

For more information on the HTTPOnly attribute, see OWASP.

For more information about the session hijacking vulnerability in McAfee IntruShield Network Security Manager, see McAfee Security Bulletin SB10005.

Technical Details

Service: http
Path: /dvwa/index.php
Received:
Set-Cookie: PHPSESSID=tcngks6nln0h9n8q0gtna1dmb3; path=/?

Cookie without HTTPOnly attribute can be accessed by scripts: _session_id

Severity: Potential Problem

Impact

A cookie without the HTTPOnly attribute could be susceptible to theft by cross-site scripting attacks.

Resolution

Modify web applications to set the HTTPOnly attribute for all cookies, or apply a patch or upgrade from your vendor.

Where can I read more about this?

For more information on the HTTPOnly attribute, see OWASP.

For more information about the session hijacking vulnerability in McAfee IntruShield Network Security Manager, see McAfee Security Bulletin SB10005.

Technical Details

Service: hbci
Path: /
Received:
Set-Cookie: _session_id=2d584f03148814f2f1751391b162b6b5; path=/?

Autocomplete enabled for password input (/)

Severity: Potential Problem

Impact

Poor authentication practices may leave the web application vulnerable to authentication attacks.

Resolution

To use HTML form-based authentication more securely in web applications, do the following:

- Remove the **`value`** attribute from the **`INPUT`** tag corresponding to the password field.
- Submit all forms to an SSL-enabled (*https*) service using the form's **`action`** attribute.
- Place all protected web directories on an SSL-enabled (*https*) service.
- Use the **`autocomplete="off"`** attribute in the **`INPUT`** tag corresponding to the password field.
- Use the **`POST`** method to submit forms containing passwords.

Where can I read more about this?

Additional information on the INPUT element is in the HTML 4.01 Specification, Section 17.4.

For more information on HTTPS, see whatis.com.

For more information on the autocomplete feature in HTML, see HTML Code Tutorial.

Technical Details

Service: hbci
Received:
<input type="password" name="user_password" id="user_password" size="30"/>

HTML page uses cleartext form-based authentication (/)

Severity: Potential Problem

Impact

Poor authentication practices may leave the web application vulnerable to authentication attacks.

Resolution

To use HTML form-based authentication more securely in web applications, do the following:

- Remove the **`value`** attribute from the **`INPUT`** tag corresponding to the password field.
- Submit all forms to an SSL-enabled (*https*) service using the form's **`action`** attribute.
- Place all protected web directories on an SSL-enabled (*https*) service.
- Use the **`autocomplete="off"`** attribute in the **`INPUT`** tag corresponding to the password field.
- Use the **`POST`** method to submit forms containing passwords.

Where can I read more about this?

Additional information on the INPUT element is in the HTML 4.01 Specification, Section 17.4.

For more information on HTTPS, see whatis.com.

For more information on the autocomplete feature in HTML, see HTML Code Tutorial.

Technical Details

Service: hbci
Received:
<input type="password" name="user_password" id="user_password" size="30"/>

HTML page uses cleartext form-based authentication (/dvwa/login.php)

Severity: Potential Problem

Impact

Poor authentication practices may leave the web application vulnerable to authentication attacks.

Resolution

To use HTML form-based authentication more securely in web applications, do the following:

- Remove the **`value`** attribute from the **`INPUT`** tag corresponding to the password field.
- Submit all forms to an SSL-enabled (*https*) service using the form's **`action`** attribute.
- Place all protected web directories on an SSL-enabled (*https*) service.
- Use the **`autocomplete="off"`** attribute in the **`INPUT`** tag corresponding to the password field.
- Use the **`POST`** method to submit forms containing passwords.

Where can I read more about this?

Additional information on the INPUT element is in the HTML 4.01 Specification, Section 17.4.

For more information on HTTPS, see whatis.com.

For more information on the autocomplete feature in HTML, see HTML Code Tutorial.

Technical Details

Service: http
Received:
<input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

HTML page uses cleartext form-based authentication (/dvwa/vulnerabilities/brute/../../login.php)

Severity: Potential Problem

Impact

Poor authentication practices may leave the web application vulnerable to authentication attacks.

Resolution

To use HTML form-based authentication more securely in web applications, do the following:

- Remove the **`value`** attribute from the **`INPUT`** tag corresponding to the password field.
- Submit all forms to an SSL-enabled (*https*) service using the form's **`action`** attribute.
- Place all protected web directories on an SSL-enabled (*https*) service.
- Use the **`autocomplete="off"`** attribute in the **`INPUT`** tag corresponding to the password field.
- Use the **`POST`** method to submit forms containing passwords.

Where can I read more about this?

Additional information on the INPUT element is in the HTML 4.01 Specification, Section 17.4.

For more information on HTTPS, see whatis.com.

For more information on the autocomplete feature in HTML, see HTML Code Tutorial.

Technical Details

Service: http
Received:
<input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

HTML page uses cleartext form-based authentication (/dvwa/vulnerabilities/csrf/../../login.php)

Severity: Potential Problem

Impact

Poor authentication practices may leave the web application vulnerable to authentication attacks.

Resolution

To use HTML form-based authentication more securely in web applications, do the following:

- Remove the **`value`** attribute from the **`INPUT`** tag corresponding to the password field.
- Submit all forms to an SSL-enabled (*https*) service using the form's **`action`** attribute.
- Place all protected web directories on an SSL-enabled (*https*) service.
- Use the **`autocomplete="off"`** attribute in the **`INPUT`** tag corresponding to the password field.
- Use the **`POST`** method to submit forms containing passwords.

Where can I read more about this?

Additional information on the INPUT element is in the HTML 4.01 Specification, Section 17.4.

For more information on HTTPS, see whatis.com.

For more information on the autocomplete feature in HTML, see HTML Code Tutorial.

Technical Details

Service: http
Received:
<input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

HTML page uses cleartext form-based authentication (/dvwa/vulnerabilities/exec/../../login.php)

Severity: Potential Problem

Impact

Poor authentication practices may leave the web application vulnerable to authentication attacks.

Resolution

To use HTML form-based authentication more securely in web applications, do the following:

- Remove the **`value`** attribute from the **`INPUT`** tag corresponding to the password field.
- Submit all forms to an SSL-enabled (*https*) service using the form's **`action`** attribute.
- Place all protected web directories on an SSL-enabled (*https*) service.
- Use the **`autocomplete="off"`** attribute in the **`INPUT`** tag corresponding to the password field.
- Use the **`POST`** method to submit forms containing passwords.

Where can I read more about this?

Additional information on the INPUT element is in the HTML 4.01 Specification, Section 17.4.

For more information on HTTPS, see whatis.com.

For more information on the autocomplete feature in HTML, see HTML Code Tutorial.

Technical Details

Service: http

Received:
<input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

HTML page uses cleartext form-based authentication (/dvwa/vulnerabilities/fi/../../login.php)

Severity: Potential Problem

Impact

Poor authentication practices may leave the web application vulnerable to authentication attacks.

Resolution

To use HTML form-based authentication more securely in web applications, do the following:

- Remove the **`value`** attribute from the **`INPUT`** tag corresponding to the password field.
- Submit all forms to an SSL-enabled (*https*) service using the form's **`action`** attribute.
- Place all protected web directories on an SSL-enabled (*https*) service.
- Use the **`autocomplete="off"`** attribute in the **`INPUT`** tag corresponding to the password field.
- Use the **`POST`** method to submit forms containing passwords.

Where can I read more about this?

Additional information on the INPUT element is in the HTML 4.01 Specification, Section 17.4.

For more information on HTTPS, see whatis.com.

For more information on the autocomplete feature in HTML, see HTML Code Tutorial.

Technical Details

Service: http
Received:
<input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

HTML page uses cleartext form-based authentication (/dvwa/vulnerabilities/sqli/../../login.php)

Severity: Potential Problem

Impact

Poor authentication practices may leave the web application vulnerable to authentication attacks.

Resolution

To use HTML form-based authentication more securely in web applications, do the following:

- Remove the **`value`** attribute from the **`INPUT`** tag corresponding to the password field.
- Submit all forms to an SSL-enabled (*https*) service using the form's **`action`** attribute.
- Place all protected web directories on an SSL-enabled (*https*) service.
- Use the **`autocomplete="off"`** attribute in the **`INPUT`** tag corresponding to the password field.
- Use the **`POST`** method to submit forms containing passwords.

Where can I read more about this?

Additional information on the INPUT element is in the HTML 4.01 Specification, Section 17.4.

For more information on HTTPS, see whatis.com.

For more information on the autocomplete feature in HTML, see HTML Code Tutorial.

Technical Details

Service: http
Received:
<input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

HTML page uses cleartext form-based authentication (/dvwa/vulnerabilities/sqli_blind/../../login.php)

Severity: Potential Problem

Impact

Poor authentication practices may leave the web application vulnerable to authentication attacks.

Resolution

To use HTML form-based authentication more securely in web applications, do the following:

- Remove the **`value`** attribute from the **`INPUT`** tag corresponding to the password field.
- Submit all forms to an SSL-enabled (*https*) service using the form's **`action`** attribute.
- Place all protected web directories on an SSL-enabled (*https*) service.
- Use the **`autocomplete="off"`** attribute in the **`INPUT`** tag corresponding to the password field.
- Use the **`POST`** method to submit forms containing passwords.

Where can I read more about this?

Additional information on the INPUT element is in the HTML 4.01 Specification, Section 17.4.

For more information on HTTPS, see whatis.com.

For more information on the autocomplete feature in HTML, see HTML Code Tutorial.

Technical Details

Service: http
Received:
<input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

HTML page uses cleartext form-based authentication (/dvwa/vulnerabilities/upload/../../login.php)

Severity: Potential Problem

Impact

Poor authentication practices may leave the web application vulnerable to authentication attacks.

Resolution

To use HTML form-based authentication more securely in web applications, do the following:

- Remove the **`value`** attribute from the **`INPUT`** tag corresponding to the password field.
- Submit all forms to an SSL-enabled (*https*) service using the form's **`action`** attribute.
- Place all protected web directories on an SSL-enabled (*https*) service.
- Use the **`autocomplete="off"`** attribute in the **`INPUT`** tag corresponding to the password field.
- Use the **`POST`** method to submit forms containing passwords.

Where can I read more about this?

Additional information on the INPUT element is in the HTML 4.01 Specification, Section 17.4.

For more information on HTTPS, see whatis.com.

For more information on the autocomplete feature in HTML, see HTML Code Tutorial.

Technical Details

Service: http
Received:
<input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

HTML page uses cleartext form-based authentication (/dvwa/vulnerabilities/xss_r/../../login.php)

Severity: Potential Problem

Impact

Poor authentication practices may leave the web application vulnerable to authentication attacks.

Resolution

To use HTML form-based authentication more securely in web applications, do the following:

- Remove the **`value`** attribute from the **`INPUT`** tag corresponding to the password field.
- Submit all forms to an SSL-enabled (*https*) service using the form's **`action`** attribute.
- Place all protected web directories on an SSL-enabled (*https*) service.
- Use the **`autocomplete="off"`** attribute in the **`INPUT`** tag corresponding to the password field.
- Use the **`POST`** method to submit forms containing passwords.

Where can I read more about this?

Additional information on the INPUT element is in the HTML 4.01 Specification, Section 17.4.

For more information on HTTPS, see whatis.com.

For more information on the autocomplete feature in HTML, see HTML Code Tutorial.

Technical Details

Service: http
Received:
<input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

HTML page uses cleartext form-based authentication (/dvwa/vulnerabilities/xss_s/../../login.php)

Severity: Potential Problem

Impact

Poor authentication practices may leave the web application vulnerable to authentication attacks.

Resolution

To use HTML form-based authentication more securely in web applications, do the following:

- Remove the **`value`** attribute from the **`INPUT`** tag corresponding to the password field.
- Submit all forms to an SSL-enabled (*https*) service using the form's **`action`** attribute.
- Place all protected web directories on an SSL-enabled (*https*) service.
- Use the **`autocomplete="off"`** attribute in the **`INPUT`** tag corresponding to the password field.
- Use the **`POST`** method to submit forms containing passwords.

Where can I read more about this?

Additional information on the INPUT element is in the HTML 4.01 Specification, Section 17.4.

For more information on HTTPS, see whatis.com.

For more information on the autocomplete feature in HTML, see HTML Code Tutorial.

Technical Details

Service: http
Received:
<input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">

weak RSA public key

Severity: Potential Problem

Impact

A remote attacker with the ability to sniff network traffic could decrypt an encrypted session.

Resolution

Re-generate the RSA key pair with a minimum length of 2048 bits.

With OpenSSL, this can be done using the following commands:

```
openssl genrsa -out filename.pem 2048
openssl rsa -in filename.pem -pubout
```

Where can I read more about this?

For more information on RSA key length requirements, see Netcraft.

Technical Details

Service: pcsync-https
key length = 1024

SSL certificate is expired

Severity: Potential Problem

Impact

When a server's SSL certificate is invalid, clients cannot properly verify that the server is authentic, resulting in a lack of trust.

Resolution

For expired certificates, contact the issuer of your SSL certificate to renew your certificate.

For certificates where the subject does not match the target, change the registered DNS name of the site to match the certificate, or contact the issuer of your SSL certificate to get a corrected certificate.

Replace self-signed certificates with certificates issued by a trusted certificate authority.

For wildcard certificates, replace the wildcard certificates with certificates whose Common Names match the host they are intended to be used with.

Where can I read more about this?

For more information on certificates see the HOWTO.

Technical Details

Service: pcsync-https
certificate valid to = 060620214036Z

SSL certificate is self signed

Severity: Potential Problem

Impact

When a server's SSL certificate is invalid, clients cannot properly verify that the server is authentic, resulting in a lack of trust.

Resolution

For expired certificates, contact the issuer of your SSL certificate to renew your certificate.

For certificates where the subject does not match the target, change the registered DNS name of the site to match the certificate, or contact the issuer of your SSL certificate to get a corrected certificate.

Replace self-signed certificates with certificates issued by a trusted certificate authority.

For wildcard certificates, replace the wildcard certificates with certificates whose Common Names match the host they are intended to be used with.

Where can I read more about this?

For more information on certificates see the HOWTO.

Technical Details

Service: pcsync-https
Issued To localhost
Issued By localhost

SSL certificate subject does not match target

Severity: Potential Problem

Impact

When a server's SSL certificate is invalid, clients cannot properly verify that the server is authentic, resulting in a lack of trust.

Resolution

For expired certificates, contact the issuer of your SSL certificate to renew your certificate.

For certificates where the subject does not match the target, change the registered DNS name of the site to match the certificate, or contact the issuer of your SSL certificate to get a corrected certificate.

Replace self-signed certificates with certificates issued by a trusted certificate authority.

For wildcard certificates, replace the wildcard certificates with certificates whose Common Names match the host they are intended to be used with.

Where can I read more about this?

For more information on certificates see the HOWTO.

Technical Details

Service: pcsync-https
Certificate Issued To: localhost

SSL server accepts weak ciphers

Severity: Potential Problem

Impact

A remote attacker with the ability to sniff network traffic could decrypt an encrypted session.

Resolution

For Apache mod_ssl web servers, use the **`SSLCipherSuite`** directive in the configuration file to specify strong ciphers only and disable SSLv2 and export ciphers.

For Microsoft IIS web servers, disable SSLv2 and any weak ciphers as described in Microsoft knowledge

base articles 187498 and 245030.

For other types of web servers, consult the web server documentation.

Where can I read more about this?

For more information, see VNU Net: Weak Security Found in Many Web Servers.

Technical Details

Service: pcsync-https
Supported ciphers: EXP-RC4-MD5:TLSv1/SSLv3:40-bit EXP-DES-CBC-SHA:TLSv1/SSLv3:40-bit EXP-EDH-RSA-DES-CBC-SHA:TLSv1/SSLv3:40-bit DES-CBC-SHA:TLSv1/SSLv3:56-bit EDH-RSA-DES-CBC-SHA:TLSv1/SSLv3:56-bit RC4-MD5:TLSv1/SSLv3:128-bit RC4-SHA:TLSv1/SSLv3:128-bit AES128-SHA:TLSv1/SSLv3:128-bit DHE-RSA-AES128-SHA:TLSv1/SSLv3:128-bit DES-CBC3-SHA:TLSv1/SSLv3:168-bit EDH-RSA-DES-CBC3-SHA:TLSv1/SSLv3:168-bit EXP-RC4-MD5:TLSv1/SSLv3:40-bit EXP-DES-CBC-SHA:TLSv1/SSLv3:40-bit EXP-EDH-RSA-DES-CBC-SHA:TLSv1/SSLv3:40-bit DES-CBC-SHA:TLSv1/SSLv3:56-bit EDH-RSA-DES-CBC-SHA:TLSv1/SSLv3:56-bit RC4-MD5:TLSv1/SSLv3:128-bit RC4-SHA:TLSv1/SSLv3:128-bit AES128-SHA:TLSv1/SSLv3:128-bit DHE-RSA-AES128-SHA:TLSv1/SSLv3:128-bit DES-CBC3-SHA:TLSv1/SSLv3:168-bit EDH-RSA-DES-CBC3-SHA:TLSv1/SSLv3:168-bit EXP-RC4-MD5:TLSv1/SSLv3:40-bit EXP-DES-CBC-SHA:TLSv1/SSLv3:40-bit EXP-EDH-RSA-DES-CBC-SHA:TLSv1/SSLv3:40-bit DES-CBC-SHA:TLSv1/SSLv3:56-bit EDH-RSA-DES-CBC-SHA:TLSv1/SSLv3:56-bit RC4-MD5:TLSv1/SSLv3:128-bit RC4-SHA:TLSv1/SSLv3:128-bit AES128-SHA:TLSv1/SSLv3:128-bit DHE-RSA-AES128-SHA:TLSv1/SSLv3:128-bit DES-CBC3-SHA:TLSv1/SSLv3:168-bit EDH-RSA-DES-CBC3-SHA:TLSv1/SSLv3:168-bit

TLS/SSL server accepts export ciphers (FREAK/Logjam attack)

Severity: Potential Problem **CVE:** CVE-2015-0204 CVE-2015-4000

Impact

A remote attacker with the ability to sniff network traffic could decrypt an encrypted session.

Resolution

For Apache mod_ssl web servers, use the **`SSLCipherSuite`** directive in the configuration file to specify strong ciphers only and disable SSLv2 and export ciphers.

For Microsoft IIS web servers, disable SSLv2 and any weak ciphers as described in Microsoft knowledge base articles 187498 and 245030.

For other types of web servers, consult the web server documentation.

Where can I read more about this?

For more information, see VNU Net: Weak Security Found in Many Web Servers.

More information about the FREAK attack is available from SMACK. More information about the Logjam attack is available from weakdh.org.

Technical Details

Service: pcsync-https
Supported ciphers: EXP-RC4-MD5:TLSv1/SSLv3:40-bit EXP-DES-CBC-SHA:TLSv1/SSLv3:40-bit EXP-EDH-RSA-DES-CBC-SHA:TLSv1/SSLv3:40-bit DES-CBC-SHA:TLSv1/SSLv3:56-bit EDH-RSA-DES-CBC-SHA:TLSv1/SSLv3:56-bit RC4-MD5:TLSv1/SSLv3:128-bit RC4-SHA:TLSv1/SSLv3:128-bit AES128-SHA:TLSv1/SSLv3:128-bit DHE-RSA-AES128-SHA:TLSv1/SSLv3:128-bit DES-CBC3-SHA:TLSv1/SSLv3:168-bit EDH-RSA-DES-CBC3-SHA:TLSv1/SSLv3:168-bit EXP-RC4-MD5:TLSv1/SSLv3:40-bit EXP-DES-CBC-SHA:TLSv1/SSLv3:40-bit EXP-EDH-RSA-DES-CBC-SHA:TLSv1/SSLv3:40-bit DES-CBC-SHA:TLSv1/SSLv3:56-bit EDH-RSA-DES-CBC-SHA:TLSv1/SSLv3:56-bit RC4-MD5:TLSv1/SSLv3:128-bit RC4-SHA:TLSv1/SSLv3:128-bit AES128-SHA:TLSv1/SSLv3:128-bit DHE-RSA-AES128-SHA:TLSv1/SSLv3:128-bit DES-CBC3-SHA:TLSv1/SSLv3:168-bit EDH-RSA-DES-CBC3-SHA:TLSv1/SSLv3:168-bit EXP-RC4-MD5:TLSv1/SSLv3:40-bit EXP-DES-CBC-SHA:TLSv1/SSLv3:40-bit EXP-EDH-RSA-DES-CBC-SHA:TLSv1/SSLv3:40-bit DES-CBC-SHA:TLSv1/SSLv3:56-bit EDH-RSA-DES-CBC-SHA:TLSv1/SSLv3:56-bit RC4-MD5:TLSv1/SSLv3:128-bit RC4-SHA:TLSv1/SSLv3:128-bit AES128-SHA:TLSv1/SSLv3:128-bit DHE-RSA-AES128-SHA:TLSv1/SSLv3:128-bit DES-CBC3-SHA:TLSv1/SSLv3:168-bit EDH-RSA-DES-CBC3-SHA:TLSv1/SSLv3:168-bit

SSL certificate is signed with weak hash function: MD5

Severity: Potential Problem **CVE:** CVE-2004-2761

Impact

The SSL/TLS certificate is signed with a weak hash function. An attacker may be able to forge a SSL/TLS certificate that would appear to be valid for the website. This may allow an attacker to perform a man-in-the-middle attack against the SSL-secured website.

Resolution

Sites using certificates signed using a vulnerable hash function should request replacement certificates signed with a more secure hash function. The offending certificates should be revoked if they have not yet expired.

Currently, the SHA-256 and SHA-512 hash functions have proven to be resistant against both collision and preimage attacks. It is advisable to use one of these hash functions at this time.

Because some legacy applications and users with outdated systems may not be able to support SHA-2, most CAs still default to using SHA-1 in an attempt to avoid user experience issues. If your CA of choice does not offer an option to use SHA-2, you may try generating a Certificate Signing Request (CSR) that specifics SHA-2 by using OpenSSL or Microsoft IIS Certificate Services.

Instructions on how to generate a SHA256 CSR can be found here.

Where can I read more about this?

Information regarding Cryptographic Hash Functions, including a summary of attack complexity against various hash functions, can be found on Wikipedia.

Details regarding the creation of a rogue Certificate Authority by exploiting vulnerabilities in the MD5 hash are provided by the Eindhoven University of Technology.

Technical Details

Service: pcsync-https

OID: 0x2a864886f70d010104

5900/TCP

Severity: Service

Technical Details

RFB 003.007

SSH

Severity: Service

Technical Details

SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu7

WWW

Severity: Service

Technical Details

HTTP/1.1 403 Forbidden
Date: Mon, 14 Dec 2015 21:12:56 GMT
Server: Apache/2.2.14 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 277
Connection: close
Content-Type:

WWW (non-standard port 3000)

Severity: Service

Technical Details

HTTP/1.1 200 OK
Connection: close
Date: Mon, 14 Dec 2015 21:12:56 GMT
Set-Cookie: _session_id=3ff6a3d17c20958d7bed61281c81c6df; path=/
Cache-Control: no-cache
Status: 200

WWW (non-standard port 8080)

Severity: Service

Technical Details

HTTP/1.1 302 Moved Temporarily
Location: http://dojo1.local:8080/index.html
Content-Length: 0
Date: Mon, 14 Dec 2015 21:12:57 GMT
Server: Apache-Coyote/1.1
Connection:

WWW (non-standard port 8443)

Severity: Service

Technical Details

HTTP/1.1 302 Moved Temporarily??Location: https://dojo1.sainttest.local:8443/index.html??Content-Length: 0??Date: Mon, 14 Dec 2015 21:13:02 GMT??Server: Apache-Coyote/1.1??Connection:

commplex-link (5001/TCP)

Severity: Service

Technical Details

\172\237\000\005

pcsync-https (8443/TCP)

Severity: Service

Technical Details

\021\003\001\000\002\002