ASV Scan Report Vulnerability Details

Report Generated: February 17, 2021

1 Scan Information

Scan Customer Company: ABC Coffee Shop

Date scan was completed: February 17, 2021

ASV Company: SAINT Corporation

Scan expiration date: May 18, 2021

The following PCI vulnerability severity levels are also used to categorize the vulnerabilities in compliance with the PCI DSS:

CVSS Score	Security Level	Scan Results	Guidance
7.0 through 10.0	High Severity	Fail	To achieve a passing scan these vulnerabilities must be corrected and the environment must be re-scanned after the corrections (with a report that shows a passing scan). Organizations should take a risk-based approach to correct these types of vulnerabilities, starting with the most critical ones (rated 10.0), then those rated 9, followed by those rated 8, 7, etc. until all vulnerabilities rated 4.0 through 10.0 are corrected.
4.0 through 6.9	Medium Severity	Fail	
0.0 through 3.9	Low Severity	Pass	While passing scan results can be achieved with vulnerabilities rated 0.0 through 3.9, organizations are encouraged, but not required, to correct these vulnerabilities

1.1 Vulnerability List

Host Name:Port	Vulnerability / Service	CVE	CVSSv2 Base Score	PCI Compliant?	PCI Severity	Details
192.0.2.15:80	Autocomplete enabled for password input		2.6	PASS	low	There are potential vulnerabilities associated with HTML form-based authentication: Autocomplete Enabled. The form allows the browser's autocomplete feature to automatically fill the password field with previously submitted values when a user begins entering a password. This feature could reveal one user's password to another user on the same computer.
192.0.2.15:443	Load Balancer detected		2.6	PASS	low	A load balancer was detected. Therefore, the vulnerability scan may have been distributed among multiple physical machines, leading to inconsistent results if the machines differ in any way. A vulnerability on one machine may have been missed if it was not present on another machine in the cluster.

192.0.2.15:25	Remote OS available	2.6	PASS	low	This machine reveals its operating system type in the information which is returned when connecting to certain TCP ports. An attacker could use this information to choose attacks which specifically target the machine's operating system version, increasing the likelihood of success.
192.0.2.15:23	Remote OS available	2.6	PASS	low	This machine reveals its operating system type in the information which is returned when connecting to certain TCP ports. An attacker could use this information to choose attacks which specifically target the machine's operating system version, increasing the likelihood of success.
192.0.2.15:80	Remote OS available	2.6	PASS	low	This machine reveals its operating system type in the information which is returned when connecting to certain TCP ports. An attacker could use this information to choose attacks which specifically target the machine's operating system version, increasing the likelihood of success.
192.0.2.15:22	Remote OS available	2.6	PASS	low	This machine reveals its operating system type in the information which is returned when connecting to certain TCP ports. An attacker could use this information to choose attacks which specifically target the machine's operating system version, increasing the likelihood of success.
192.0.2.15:443	Remote OS available	2.6	PASS	low	This machine reveals its operating system type in the information which is returned when connecting to certain TCP ports. An attacker could use this information to choose attacks which specifically target the machine's operating system version, increasing the likelihood of success.
192.0.2.15:22	SSH supports weak MAC algorithms	2.6	PASS	low	The SSH server supports insecure MAC algorithms, which could allow an attacker who is able to modify network traffic to alter the data in the encrypted session.
192.0.2.15:25	SSL/TLS server allows anonymous key exchange	2.6	PASS	low	Some servers support anonymous key exchange algorithms, such as Anonymous Diffie-Hellman. Such algorithms allow the TLS/SSL session to proceed without any exchange of certificates, effectively taking away the client's ability to verify that it is communicating with the correct server and not an imposter. This makes it easier for attackers to launch man-in-the-middle attacks.
192.0.2.15:25	Sendmail command VRFY is enabled	2.6	PASS	low	While sendmail and the SMTP protocol have proven very useful in everyday life, they have presented us with a security problem. A malicious user able to connect to a machine running sendmail may be able to acquire information about user accounts on that system. As discussed earlier in this briefing, after getting this information, the hacker may be able to do some dreadful things indeed. You might be asking how a malicious user could get this account information. The answer is through the use of special SMTP commands.
192.0.2.15:443	TCP timestamp requests enabled	2.6	PASS	low	TCP timestamps are enabled on the remote host. This could allow a remote attacker to estimate the amount of time since the remote host was last booted.
192.0.2.15:443	Web server default page detected	2.6	PASS	low	A web server containing a default web page is running. This indicates that the web server has not been configured, possibly because the owner of the target is not aware that web server software has been installed. Unconfigured web servers create an unnecessary exposure and could contain potential security vulnerabilities.
192.0.2.15:80	Web server default page detected	2.6	PASS	low	A web server containing a default web page is running. This indicates that the web server has not been configured, possibly because the owner of the target is not aware that web server software has been installed. Unconfigured web servers create an unnecessary exposure and could contain potential security vulnerabilities.

192.0.2.15:80	Web site may be vulnerable to clickjacking attacks		2.6	PASS	low	The web site may be vulnerable to a clickjacking attack, where an attacker's page lays specially crafted content over a hidden iframe containing a page from the real web site. Clicking on what appears to be a harmless page element in the attacker's content would actually cause a mouse click on the real web site which performs some malicious activity. For example, the attacker may place a link labeled "Click Here to Win" over a "Delete Account" button from the real web site hidden in the iframe. When the user clicks on the link, the real web site would handle the mouse click event on the button and delete the user's account.
192.0.2.15:443	weak https cache policy		2.6	PASS	low	The web site does not set a strong caching policy for https sessions. Pages from secure https sessions should never be cached, since that would compromise the confidentiality provided by SSL if an attacker is able to access the stored copy of the page. It is expected that caches will refrain from caching pages from https sessions, but it is insufficient to rely on that behavior. An explicit caching policy should be set by the web server.
192.0.2.15	ICMP timestamp requests enabled	CVE-1999-0524	0.0	PASS	low	ICMP information such as (1) netmask and (2) timestamp is allowed from arbitrary hosts.
192.0.2.15:25	SMTP					
192.0.2.15:22	SSH					
192.0.2.15:23	Telnet					
192.0.2.15:80	WWW					
192.0.2.15:443	WWW (Secure)					

2 Details

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

2.1 192.0.2.15

IP Address: 192.0.2.15 **Host type:** Ubuntu 16.04
Scan time: Feb 17 12:54:43 2021

Autocomplete enabled for password input

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 80/tcp
PCI Compliant: PASS

Impact

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

Background

Some web applications perform authentication by requiring a user to enter a login and password into an HTML form. This type of authentication is achieved using the HTML `INPUT` element with the `type` attribute set to `password`.

The Problem

There are potential vulnerabilities associated with HTML form-based authentication:

- **Autocomplete Enabled**. The form allows the browser's autocomplete feature to automatically fill the password field with previously submitted values when a user begins entering a password. This feature could reveal one user's password to another user on the same computer.

Resolution

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

- Use the **`autocomplete="off"`** attribute in the **`INPUT`** tag corresponding to the password field.

References

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
Location: /authtest.php
Received: <input name="password" type="password"><br/>

Load Balancer detected

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 443/tcp
PCI Compliant: PASS

Impact

The scan results may be inconclusive.

Background

Load Balancing is the act of distributing requests for the same resource evenly among multiple physical servers, known as a cluster. This allows more requests to be processed in a shorter amount of time. Load balancing can be implemented either by creating multiple address records in the DNS table, or by using a separate server or device which distributes the requests to the actual servers.

The Problem

A load balancer was detected. Therefore, the vulnerability scan may have been distributed among multiple physical machines, leading to inconsistent results if the machines differ in any way. A vulnerability on one machine may have been missed if it was not present on another machine in the cluster.

Resolution

Verify that all of the machines in the cluster are configured identically, or position the scanner such that it can scan the individual servers without being affected by the load balancer.

References

See page 21 of the PCI DSS ASV Program Guide for more information on handling load balancers during compliance scanning.

Technical Details

Service: https

Response headers differ:

```
HTTP/1.1 400 Bad Request
Server: Apache/2.4.18 (Ubuntu)
Connection: close
Content-Type: text/html; charset=iso-8859-1
HTTP/1.1 200 OK
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Tue, 05 Nov 2019 21:32:39 GMT
ETag: "2c39-596a02be83305"
Accept-Ranges: bytes
Content-Length: 11321
Vary: Accept-Encoding
Connection: close
Content-Type: text/html
```

Remote OS available

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 22/tcp
PCI Compliant: PASS

Impact

The ability to detect which operating system is running on a machine enables attackers to be more accurate in attacks.

Background

Many systems include specific operating system information in the data which is returned when connecting to certain TCP ports. This data is known as the *banner* for a service.

The Problem

Remote OS available

05/27/08
This machine reveals its operating system type in the information which is returned when connecting to certain TCP ports. An attacker could use this information to choose attacks which specifically target the machine's operating system version, increasing the likelihood of success.

Resolution

Including the operating system in service banners is usually unnecessary. Therefore, change the banners of the services which are running on accessible ports. This can be done by disabling unneeded services, modifying the banner in a service's source code or configuration file if possible, or using TCP wrappers to modify the banner as described in the Red Hat Knowledgebase.

References

An example of ways to remove the Remote OS and other information is at my digital life.

Technical Details

Service: ssh
Received:

```
SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8
```

Remote OS available

PCI Severity: low

CVSSv2 Base Score: 2.6 **Port:** 23/tcp
PCI Compliant: PASS

Impact

The ability to detect which operating system is running on a machine enables attackers to be more accurate in attacks.

Background

Many systems include specific operating system information in the data which is returned when connecting to certain TCP ports. This data is known as the *banner* for a service.

The Problem

Remote OS available

05/27/08
This machine reveals its operating system type in the information which is returned when connecting to certain TCP ports. An attacker could use this information to choose attacks which specifically target the machine's operating system version, increasing the likelihood of success.

Resolution

Including the operating system in service banners is usually unnecessary. Therefore, change the banners of the services which are running on accessible ports. This can be done by disabling unneeded services, modifying the banner in a service's source code or configuration file if possible, or using TCP wrappers to modify the banner as described in the Red Hat Knowledgebase.

References

An example of ways to remove the Remote OS and other information is at my digital life.

Technical Details

Service: telnet
Received:
Ubuntu 16.04.6 LTS

Remote OS available

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 80/tcp
PCI Compliant: PASS

Impact

The ability to detect which operating system is running on a machine enables attackers to be more accurate in attacks.

Background

Many systems include specific operating system information in the data which is returned when connecting to certain TCP ports. This data is known as the *banner* for a service.

The Problem

Remote OS available

05/27/08
This machine reveals its operating system type in the information which is returned when connecting to certain TCP

ports. An attacker could use this information to choose attacks which specifically target the machine's operating system version, increasing the likelihood of success.

Resolution

Including the operating system in service banners is usually unnecessary. Therefore, change the banners of the services which are running on accessible ports. This can be done by disabling unneeded services, modifying the banner in a service's source code or configuration file if possible, or using TCP wrappers to modify the banner as described in the Red Hat Knowledgebase.

References

An example of ways to remove the Remote OS and other information is at my digital life.

Technical Details

Service: http
Received:
Server: Apache/2.4.18 (Ubuntu)

Remote OS available

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 25/tcp
PCI Compliant: PASS

Impact

The ability to detect which operating system is running on a machine enables attackers to be more accurate in attacks.

Background

Many systems include specific operating system information in the data which is returned when connecting to certain TCP ports. This data is known as the *banner* for a service.

The Problem

Remote OS available

05/27/08
This machine reveals its operating system type in the information which is returned when connecting to certain TCP ports. An attacker could use this information to choose attacks which specifically target the machine's operating system version, increasing the likelihood of success.

Resolution

Including the operating system in service banners is usually unnecessary. Therefore, change the banners of the services which are running on accessible ports. This can be done by disabling unneeded services, modifying the banner in a service's source code or configuration file if possible, or using TCP wrappers to modify the banner as described in the Red Hat Knowledgebase.

References

An example of ways to remove the Remote OS and other information is at my digital life.

Technical Details

Service: smtp
Received:

220 192.0.2.15 ESMTP Postfix (Ubuntu)

Remote OS available

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 443/tcp
PCI Compliant: PASS

Impact

The ability to detect which operating system is running on a machine enables attackers to be more accurate in attacks.

Background

Many systems include specific operating system information in the data which is returned when connecting to certain TCP ports. This data is known as the *banner* for a service.

The Problem

Remote OS available

05/27/08
This machine reveals its operating system type in the information which is returned when connecting to certain TCP ports. An attacker could use this information to choose attacks which specifically target the machine's operating system version, increasing the likelihood of success.

Resolution

Including the operating system in service banners is usually unnecessary. Therefore, change the banners of the services which are running on accessible ports. This can be done by disabling unneeded services, modifying the banner in a service's source code or configuration file if possible, or using TCP wrappers to modify the banner as described in the Red Hat Knowledgebase.

References

An example of ways to remove the Remote OS and other information is at my digital life.

Technical Details

Service: https
Received:
Server: Apache/2.4.18 (Ubuntu)

SSH supports weak MAC algorithms

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 22/tcp
PCI Compliant: PASS

Impact

A remote attacker with the ability to modify network traffic could alter the data in an encrypted session.

Background

Secure Shell, or **ssh**, is a program used to log into another computer over a network, execute commands on a remote machine and move files from one machine to another. It provides strong authentication and secure communications over unsecure communication channels. **ssh** is intended as a replacement for **`rlogin`**, **`rsh`** and **`rcp`**. Additionally, **ssh** provides secure **`X`** connections and secure forwarding of arbitrary **`TCP`** connections.

At the beginning of an SSH session, the client and server negotiate the Message Authentication (MAC) algorithm, which is used to ensure integrity of the session data.

The Problem

The SSH server supports insecure MAC algorithms, which could allow an attacker who is able to modify network traffic to alter the data in the encrypted session.

Resolution

Configure the SSH server not to support the MD5 and SHA1 algorithms.

To do this on OpenSSH servers, edit the **`sshd_config`** file and add a **`Macs`** line (or modify this line if it already exists) as follows:

```
    Macs hmac-sha2-256,hmac-sha2-512,hmac-ripemd160,hmac-ripemd160@openssh.com
```

Then restart the SSH service.

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

References

For more information on configuring SSH, see OpenSSH Security and Hardening.

Technical Details

Service: ssh
SSH2 mac_algorithms_client_to_server =
umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1

SSL/TLS server allows anonymous key exchange

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 25/tcp
PCI Compliant: PASS

Impact

Services which allow anonymous key exchange could facilitate man-in-the-middle attacks.

Background

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide communication security over the Internet. At the beginning of each TLS/SSL session, the client and server negotiate a *cipher suite*, which determines the key exchange algorithm, encryption algorithm, and hash algorithm which will be used in the session. Most key exchange algorithms include the exchange of certificates, which allows the client to verify that it is communicating with the correct server.

The Problem

02/24/17
Some servers support anonymous key exchange algorithms, such as Anonymous Diffie-Hellman. Such algorithms allow the TLS/SSL session to proceed without any exchange of certificates, effectively taking away the client's ability to verify that it is communicating with the correct server and not an imposter. This makes it easier for attackers to launch man-in-the-middle attacks.

Resolution

Disable anonymous key exchange algorithms in the affected service.

For Apache, this is done by adding the string **`!aNULL`** to the **`SSLCipherSuite`** directive.

For other web servers, consult the web server documentation.

References

More information about cipher suites can be found in RFC 5246 Appendix A.5.

Technical Details

Service: smtp
Server accepted TLS 1.0 anonymous cipher: TLS_DH_anon_WITH_RC4_128_MD5

Sendmail command VRFY is enabled

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 25/tcp
PCI Compliant: PASS

Impact

By exploiting the **`sendmail`** vulnerability, a malicious user may be able to gather information, such as user names, about user accounts located on the system on which **`sendmail`** resides. Using this information, it would then be a relatively simple task for the malicious user to gain access to the system. If the user is able to gain access to the system through an administrative (or ***root***) account, the results could be catastrophic indeed. The malicious user, or hacker, could decide to overwrite important system files, delete file systems altogether or use the compromised system as a base from which to compromise other systems on the network. A secondary issue is that the hacker may also be able to access any mailing lists used by **`sendmail`**. This means, of course, that the hacker would have the email addresses of any person found on these mailing lists.

Background

`Sendmail`, first released circa 1983, is a mail router program, and was designed to route email between peers on a network and also to route mail between networks. Note that **`sendmail`** is a *routing* program, and not an application that an ordinary user would use to format and send messages. Instead, **`sendmail`** accepts formatted messages from an email program (such as Outlook Express, Eudora or Pegasus), and then sends them to the appropriate recipients. The message is sent using the Simple Mail Transfer Protocol (SMTP), which was designed to be a reliable and effective transport for mail messages.

The Problem

While **`sendmail`** and the **`SMTP`** protocol have proven very useful in everyday life, they have presented us with a security problem. A malicious user able to connect to a machine running **`sendmail`** may be able to acquire information about user accounts on that system. As discussed earlier in this briefing, after getting this information, the hacker may be able to do some dreadful things indeed.

You might be asking how a malicious user could get this account information. The answer is through the use of special **`SMTP`** commands. These **`SMTP`** commands allow for the distribution of certain user account data to anyone who knows how to request it. Basically, the hacker has to do nothing more than connect to a remote system and simply ask for the account data. The example below shows the steps a hacker would take to get this account data:

```
telnet <hostname> 25
220 <hostname> ESMTP Sendmail 8.8.7/8.8.7; Tues 27 Apr 1999 11:11:20 -0400
EXPN root
```

```
250 root <root@hostname>
EXPN guest
250 guest <guest@hostname>
EXPN lpr
250 lpr <lpr@hostname>
QUIT
```

Before we continue our discussion, let us examine exactly what is happening in the example above. First, the hacker connects to port 25 on the remote system running **sendmail**. Port 25 is the default port on which **SMTP** runs (remember that **sendmail** and **SMTP** work in conjunction to process email messages). After connection, the hacker will get back a line of text. Contained in this text will be the version number of the **sendmail** program running on the remote system, as well as the current date and time. At this point, the malicious user may start requesting user account information using special **SMTP** commands. In the above example, the hacker uses the **EXPN** command, followed by an account name common to most systems (at this point, the hacker is engaging in educated guesswork). If that account is indeed on the remote system, information will be returned about that account. Another command that could be used for this purpose is the **VRFY** command. In the above example, the hacker guessed, correctly as it turns out, that the **root**, **guest**, **lpr** would exist on the system.

The malicious user now knows that a **guest** account exists on the system (this is a default account included, and left, on most systems.) Also, there seems to be a printer account, an account used by the printer to talk to the server, which could be used to access the system (this is the **lpr** account in the above example). Armed with this information, the malicious user can now begin his or her break-in attempts in earnest (using such tools as telnet, ssh or FTP). As such, it is always a good idea to disable the **EXPN** and **VRFY** commands (another good reason is that version 8.6.10, and earlier versions, built with **sendmail** version 5.x as their base are susceptible to buffer overflow attacks).

Resolution

To eliminate the vulnerability discussed above, we will want to disable the **EXPN** and **VRFY** commands (as discussed above). To do so, you will need to modify the **sendmail** configuration file (sendmail.cf). The example below shows how to do this:

```
#privacy flags
O PrivacyOptions=authwarnings
O PrivacyOptions=noexpn
O PrivacyOptions=novrfy
```

the "**noexp**" text in the above example disables the **EXPN** command, while the "**novrfy**" text will disable the **VRFY** command.

References

Email Protocols gives a look at all the different protocols including sendmail. *Connected: An Internet Encyclopedia* also has some information on the EXPN & VRFY commands.

Technical Details

Service: smtp

TCP timestamp requests enabled

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 443/tcp
PCI Compliant: PASS

Impact

A remote attacker could possibly determine the amount of time since the computer was last booted.

Background

The *Transmission Control Protocol* (TCP) is the protocol used by services such as **telnet**, **ftp**, and **smtp** to establish a connection between a client and a server. The **TCP** packet header includes an *option* field, which can hold zero or more options. One of those options is the *TCP timestamp*, which is used for round-trip time measurement. The value of the timestamp is obtained from a virtual clock which is proportional to real time.

The Problem

TCP timestamps are enabled on the remote host. This could allow a remote attacker to estimate the amount of time since the remote host was last booted.

Resolution

TCP timestamps are generally only useful for testing, and support for them should be disabled if not needed.

References

More information on TCP timestamps and round-trip time measurement is available in RFC1323 and Microsoft Article 224829.

Technical Details

Service: https
timestamp=1383124537; uptime guess=64d 0h 48m 18s

Web server default page detected

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 80/tcp
PCI Compliant: PASS

Impact

An unconfigured web server creates an unnecessary security exposure on the network.

Background

Many operating systems, such as Microsoft Windows and Linux, include web server software as either a default or optional package. The web server software usually includes a default web page. The default web page is the page that is served if a web client sends the web server a request before the web server has been configured.

The Problem

02/05/10 A web server containing a default web page is running. This indicates that the web server has not been configured, possibly because the owner of the target is not aware that web server software has been installed.

Unconfigured web servers create an unnecessary exposure and could contain potential security vulnerabilities.

Resolution

Disable unconfigured web servers. If the web server is needed, replace the default page with some appropriate site-specific content.

References

For more information about default web pages, see about.com.

Technical Details

Service: http
Received:
<title>Apache2 Ubuntu Default Page: It works</title>

Web server default page detected

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 443/tcp
PCI Compliant: PASS

Impact

An unconfigured web server creates an unnecessary security exposure on the network.

Background

Many operating systems, such as Microsoft Windows and Linux, include web server software as either a default or optional package. The web server software usually includes a default web page. The default web page is the page that is served if a web client sends the web server a request before the web server has been configured.

The Problem

02/05/10 A web server containing a default web page is running. This indicates that the web server has not been configured, possibly because the owner of the target is not aware that web server software has been installed.

Unconfigured web servers create an unnecessary exposure and could contain potential security vulnerabilities.

Resolution

Disable unconfigured web servers. If the web server is needed, replace the default page with some appropriate site-specific content.

References

For more information about default web pages, see about.com.

Technical Details

Service: https
Received:
<title>Apache2 Ubuntu Default Page: It works</title>

Web site may be vulnerable to clickjacking attacks

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 80/tcp
PCI Compliant: PASS

Impact

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

Background

An iframe is an HTML element used to display one web page inside of another.

The Problem

The web site may be vulnerable to a clickjacking attack, where an attacker's page lays specially crafted content over a hidden iframe containing a page from the real web site. Clicking on what appears to be a harmless page element in the attacker's content would actually cause a mouse click on the real web site which performs some malicious activity. For example, the attacker may place a link labeled "Click Here to Win" over a "Delete Account" button from the real web site hidden in the iframe. When the user clicks on the link, the real web site would handle the mouse click event on the button and delete the user's account.

Resolution

Prevent unauthorized sites from using your web pages in iframes by configuring the web server to send the X-Frame-Options response header and set it to SAMEORIGIN, DENY, or ALLOW-FROM. 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.

References

More information about clickjacking is available from OWASP.

Technical Details

Service: http
Site allows authentication and has no X-Frame-Options or Content-Security-Policy header
Received: <input name="password" type="password"><br/>

weak https cache policy

PCI Severity: low
CVSSv2 Base Score: 2.6 **Port:** 443/tcp
PCI Compliant: PASS

Impact

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

Background

To improve response times and reduce network bandwidth usage, retrieved web pages are often saved to a local repository called a *cache*. The next time the page is requested, the file can be reloaded from the cache, usually must faster than it would take to reload the page from the original web server.

Caches exist both in web browsers, where they are accessible only to the local machine, and on the network, where they are shared among clients.

The Problem

The web site does not set a strong caching policy for **`https`** sessions. Pages from secure **`https`** sessions should never be cached, since that would compromise the confidentiality provided by SSL if an attacker is able to access the stored copy of the page. It is expected that caches will refrain from caching pages from **`https`** sessions, but it is insufficient to rely on that behavior. An explicit caching policy should be set by the web server.

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"
  ```

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`**.

References

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

Technical Details

Service: https
Sent:
GET /manual HTTP/1.0
Host: 192.0.2.15
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Received:
(no Cache-Control header)

ICMP timestamp requests enabled

PCI Severity: low	**CVE:** CVE-1999-0524
CVSSv2 Base Score: 0.0	**Port:**
PCI Compliant: PASS	

Impact

A remote attacker could obtain sensitive information about the network.

Background

The Internet Control Message Protocol (ICMP) is a protocol used primarily for sending diagnostic messages and error messages between computers. The protocol defines a number of different message types, including echo requests and replies (used by the *ping* utility) and destination unreachable messages.

The Problem

CVE 1999-0524
ICMP defines a number of message types which disclose information about a computer. These message types were designed to help synchronize computers on a network, but in practice are rarely needed and should be disabled to prevent attackers from using them. Such message types include:

- *Timestamp requests*. These messages could be used by an attacker to determine the system's clock state, which could be used to defeat authentication mechanisms which rely on certain pseudo-random number

generators.
- *Netmask requests*. These messages could be used by an attacker to gather information about a network's subnet structure.

Resolution

Configure the system or firewall not to allow ICMP timestamp requests (message type 13) or ICMP netmask requests (message type 17). Instructions for doing this on specific platforms are as follows:

Linux:
Use ipchains or iptables to filter ICMP netmask requests using the command:

```
ipchains -A input -p icmp --icmp-type address-mask-request -j DROP
```

Use ipchains or iptables to filter ICMP timestamp requests using the commands:

```
ipchains -A input -p icmp --icmp-type timestamp-request -j DROP
ipchains -A output -p icmp --icmp-type timestamp-reply -j DROP
```

To ensure that this change persists after the system reboots, put the above command into the system's boot-up script (typically **`/etc/rc.local`**).

References

For more information about ICMP, see RFC792.

Technical Details

Service: icmp
timestamp=03d393ef

SMTP

Severity: Service
Port: 25/tcp

Technical Details

220 192.0.2.15 ESMTP Postfix (Ubuntu)

SSH

Severity: Service
Port: 22/tcp

Technical Details

SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8

Telnet

Severity: Service
Port: 23/tcp

Technical Details

Ubuntu 16.04.6 LTS

WWW

Severity: Service
Port: 80/tcp

Technical Details

```
HTTP/1.1 200 OK
Date: Wed, 17 Feb 2021 17:48:51 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Tue, 05 Nov 2019 21:32:39 GMT
ETag: "2c39-596a02be83305"
Accept-Ranges:
```

WWW (Secure)

Severity: Service
Port: 443/tcp

Technical Details

```
HTTP/1.1 400 Bad Request
Date: Wed, 17 Feb 2021 17:48:51 GMT
Server: Apache/2.4.18 (Ubuntu)
Content-Length: 454
Connection: close
Content-Type: text/html;
```

Scan Session: sample_scan; Scan Policy: PCI External; Scan Data Set: 17 February 2021 12:54