ABC Coffee ASV Vulnerability Details

Report Generated: April 18, 2014

Part 1. Scan Information

Scan Customer Company: ABC Coffee Shop
Date scan was completed: April 17, 2014
ASV Company: SAINT Corporation
Scan expiration date: July 16, 2014

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

IP Address:Port	Vulnerability /Service	CVE	CVSSv2 Base Score	PCI Compliant?	PCI Severity	Details
10.8.0.2	TCP reset using approximate sequence number	CVE-2004-0230	5.0	PASS	medium	TCP, when using a large Window Size, makes it easier for remote attackers to guess sequence numbers and cause a denial of service (connection loss) to persistent TCP connections by repeatedly injecting a TCP RST packet, especially in protocols that use long-lived connections, such as BGP.

10.8.0.2:80	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.
10.8.0.2:23	telnet receives cleartext passwords	2.6	PASS	low	Telnet is a cleartext protocol. It does not require encryption between the client and server. Therefore, telnet passwords and other sensitive information from the user's session could be captured by an attacker, if the attacker is able to place a network sniffer somewhere between the client and the server.
10.8.0.2:80	web server uses cleartext HTTP Basic authentication (/)	2.6	PASS	low	There are several potential vulnerabilities associated with HTML form-based authentication: Authentication Credentials Prefilled.
10.8.0.2:23	Telnet				
10.8.0.2:80	WWW				

Details

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

2.1 Input Authentication vulnerabilities

Updated 10/05/09

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 Problems

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

- **Authentication Credentials Prefilled**. The password field is prefilled with a default value, possibly allowing universal access to the application being authenticated.
- **Clear-text Form-based Authentication**. The password is sent over the network unencrypted when a user submits the login form, thereby allowing an attacker who is capable of sniffing the network to view the password.
- **Clear-text HTTP Basic Authentication** The password is sent over the network unencrypted when a user authenticates to a protected web directory, thereby allowing an attacker who is capable of sniffing the network to view the password.

- **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:

- 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.

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.

2.2 Remote OS available vulnerabilities

Created 05/27/08

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 Problems

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.

Where can I read more about this?

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

2.3 TCP reset

Created 04/29/04
CVE 2004-0230

Impact

A remote attacker could cause a denial of service on systems which rely upon persistent TCP connections.

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. Every **`TCP`** packet includes a *sequence number* in the header to ensure that all packets are received at the destination and re-assembled in the correct order. The sequence numbering begins with an *initial sequence number* which is chosen by the server and sent to the client when the connection is established. Thus, sequence numbers also help to verify the identity of the client, since only the intended client has knowledge of the initial sequence number.

The Border Gateway Protocol (BGP) is a TCP protocol used by routers to exchange routing information. It is primarily used by Internet service providers.

The Problem

Clients and servers using the TCP protocol negotiate the size of a window of allowed sequence numbers. That is, TCP packets with sequence numbers within a certain range will be accepted. Thus, an attacker need only know an approximate sequence number, and not necessarily the exact sequence number, in order to inject packets into a session. Even if the attacker does not know the initial sequence number, he or she could still determine an acceptable sequence number by trial and error, incrementing each trial by the window size. This could be done in a matter of seconds if the attacker has access to a T-1 Internet connection and the window size is large.

The end result of this attack is that an attacker could terminate existing TCP connections by injecting a reset message into the stream. Services such as BGP which rely on persistent TCP connections are at the most risk from this vulnerability. Sustained attacks could result in a denial of service.

Resolution

To correct this problem on Cisco devices, apply one of the fixes referenced in the Cisco security advisories for IOS and non-IOS operating systems. Refer to US-CERT Vulnerability Note VU#415294 and NISSC vulnerability advisory 236929 for other vendor fixes.

If a fix is not available, this problem can be worked around by using a secure protocol such as IPsec, or by filtering incoming connections to services such as BGP which rely on persistent TCP connections at the firewall, such that only allowed addresses may reach them.

Where can I read more about this?

This vulnerability was reported in US-CERT alert 04-111A.

For more information on TCP, see RFC793.

2.4 telnet server

Created 01/29/13

Impact

Passwords could be stolen if an attacker is able to capture network traffic to and from the telnet server.

Background

Telnet is a TCP protocol enabling interactive command sessions on remote computers. A typical telnet session begins with the user sending a login name and password to the remote computer.

The Problem

Telnet is a cleartext protocol. It does not require encryption between the client and server. Therefore, telnet passwords and other sensitive information from the user's session could be captured by an attacker, if the attacker is able to place a network sniffer somewhere between the client and the server.

Resolution

Disable the telnet service and use a more secure protocol such as SSH to access the computer remotely. If telnet cannot be disabled, restrict access using iptables or TCP Wrappers such that only addresses on a local, trusted network can connect.

Where can I read more about this?

For more information, see Protocols - The Problem With Cleartext.
