



My FreeScan Vulnerabilities Report

For 193.174.103.20 on Dec 07, 2006

[Print](#)[Help](#)

Thank you for trying FreeScan. Below you'll find the complete results of your scan, including whether or not the IP you provided is exposed to any vulnerabilities. For detected vulnerabilities, a complete description of the issue, possible consequences if exploited, and an assigned severity level are provided. Follow links to verified remedies to fix these issues before they can be exploited.

FreeScan is just one component of QualysGuard®. To experience all of QualysGuard's vulnerability management capabilities (both perimeter and internal) sign up for a free 7-day trial of QualysGuard. With your trial, you will receive customized network mapping with access to an unlimited number of scans and get comprehensive reports that include vulnerability trending, business risk assessment, risk matrixes, policy & compliance reporting and much more.

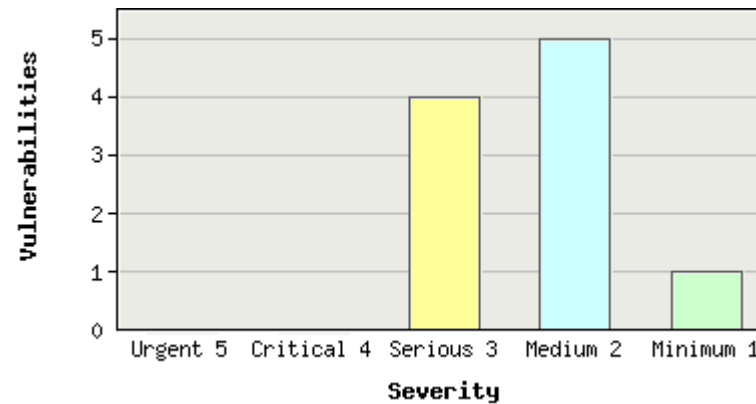
Sign up now for your [Free 7-day Trial](#)

Email this [Free Network Security Scan](#) to a colleague.

Summary for 193.174.103.20

Vulnerabilities

	0	Severity 5 (Urgent)
	0	Severity 4 (Critical)
	4	Severity 3 (Serious)
	5	Severity 2 (Medium)
	1	Severity 1 (Minimum)
10 Total		



List of Vulnerabilities for 193.174.103.20				
Severity	Analysis			
	3	Web Server Brute Force Discovery of Unix Account Names Vulnerability		
	3	SSL Server Supports Weak Encryption Vulnerability		
	3	SSL Server May Be Forced to Use Weak Encryption Vulnerability		
	3	SSL Server Has SSLv2 Enabled Vulnerability		
	2	Web Server HTTP Trace/Track Method Support Cross-Site Tracing Vulnerability		
	2	Web Server HTTP Trace/Track Method Support Cross-Site Tracing Vulnerability		
	2	Netscape/OpenSSL Cipher Forcing Bug		
	2	Global User List		
	2	SSL Certificate - Signature Verification Failed Vulnerability		
	1	Apache Web Server ETag Header Information Disclosure Weakness		

Detailed Vulnerabilities for 193.174.103.20	
Severity	Analysis
	Vulnerability: Web Server Brute Force Discovery of Unix Account Names Vulnerability Qualys ID : 5001 CVE ID : CVE-2001-1013 Port : 80 Diagnosis: When a request for a user is made (http://your.host/~user), certain servers (such as

 3	Apache Versions 1.3.12 and 1.3.9) return a different reply depending on whether the account user exists on the host or not. If a request is made for an account that exists on the host, a 403 error is returned. If a request is made for a non-existent account, then a 404 error is returned. Consequences: Unauthorized remote users can implement brute force attacks on the Web server to guess a valid account name on the server. Even though they may be successful in obtaining a valid account, they will still have to guess the password. However, if user passwords are weak, some services may also be brute forced. Solution: Disable the default-enabled "UserDir" directive. To do so, add the following line to the httpd.conf file: UserDir Disabled Apache Versions 1.3.9 and 1.3.12 are vulnerable. Other Web servers may also be vulnerable. There are currently no patches available. We strongly advise you to upgrade to a later version of Apache. Result: Detailed result listings are provided in the Free 7-day Trial ^ back to top
	Vulnerability: SSL Server Supports Weak Encryption Vulnerability Qualys ID : 38140 Port : 443 Diagnosis: The Secure Socket Layer (SSL) protocol allows for secure communication between a client and a server. SSL encryption ciphers are classified based on encryption key length as follows: ■ HIGH - key length larger than 128 bits ■ MEDIUM - key length equal to 128 bits ■ LOW - key length smaller than 128 bits Messages encrypted with LOW encryption ciphers are easy to decrypt. Commercial SSL servers should only support MEDIUM or HIGH strength ciphers to guarantee transaction security. The following link provide more information about this vulnerability:

■ SSL 3.0 Specification

Please note that this detection only checks for weak cipher support at the SSL layer. Some servers may implement additional protection at the data layer. For example, some SSL servers and SSL proxies (such as SSL accelerators) allow cipher negotiation to complete but send back an error message and abort further communication on the secure channel. This vulnerability may not be exploitable for such configurations.

Consequences: An attacker can exploit this vulnerability to decrypt secure communications without authorization.

Solution:

Disable support for LOW encryption ciphers.
Apache

Typically, for Apache/mod_ssl, httpd.conf or ssl.conf should have the following lines:

```
SSLProtocol -ALL +SSLv3 +TLSv1
```

```
SSLCipherSuite ALL:!aNULL:!ADH:!eNULL:!LOW:!EXP:RC4+RSA:+HIGH:+MEDIUM
```

For Apache/apache_ssl include the following line in the configuration file (httpsd.conf):

```
SSLRequireCipher ALL:!aNULL:!ADH:!eNULL:!LOW:!EXP:RC4+RSA:+HIGH:+MEDIUM
```

Tomcat

```
sslProtocol="SSLv3"
```

```
ciphers="SSL_RSA_WITH_RC4_128_MD5,SSL_RSA_WITH_RC4_128_SHA,SSL_DHE_RSA_W  
ITH_3DES_EDE_CBC_SHA"
```

IIS

How to Control the Ciphers for SSL and TLS on IIS (IIS restart required)

How to Restrict the Use of Certain Cryptographic Algorithms and Protocols in Schannel.dll
(Windows restart required)

	How to disable PCT 1.0, SSL 2.0, SSL 3.0, or TLS 1.0 in Internet Information Services (Windows restart required) For Novell Network 6.5 please refer to the following document SSL Allows the use of Weak Ciphers. -TID10100633 Result: Detailed result listings are provided in the Free 7-day Trial ^ back to top
 3	Vulnerability: SSL Server May Be Forced to Use Weak Encryption Vulnerability Qualys ID : 38141 Port : 443 Diagnosis: The Secure Socket Layer (SSL) protocol allows for secure communication between a client and a server. SSL encryption ciphers are classified based on the encryption key length as follows: ■ HIGH - key length larger than 128 bits ■ MEDIUM - key length equal to 128 bits ■ LOW - key length smaller than 128 bits During the SSL handshake, the SSL client and the SSL server negotiate which cipher to use for the session. The SSL server chooses a cipher from a list proposed by the SSL client. The list is sorted by preference with the first cipher in the list being the most preferred. This vulnerability is reported when the list of ciphers submitted by the client has a mixture of LOW, MEDIUM and HIGH ciphers with a LOW grade cipher listed first, and the SSL server chooses to use the LOW grade cipher even though it supports at least one MEDIUM or HIGH grade cipher in the list. Messages encrypted with LOW encryption ciphers are easy to decrypt. Commercial SSL servers should only support MEDIUM or HIGH strength ciphers to guarantee transaction security. SSL servers support a LOW grade cipher even though the client supports stronger ciphers. Consequences: An attacker can exploit this vulnerability to decrypt secure communications without authorization. Solution: Disable support for LOW encryption ciphers. Typically, for Apache/mod_ssl, httpd.conf or ssl.conf should have the following lines:

	SSLProtocol -ALL +SSLv3 +TLSv1 SSLCipherSuite ALL:!aNULL:!ADH:!eNULL:!LOW:!EXP:RC4+RSA:+HIGH:+MEDIUM If for some reason LOW grade cipher are needed, then using the SSLHonorCipherOrder directive will enforce the server's preference on cipher selection and will guarantee that weak ciphers will be used only if nothing else is available. SSLHonorCipherOrder Directive How to Control the Ciphers for SSL and TLS on IIS For Novell Netware 6.5 please refer to the following document SSL Allows the use of Weak Ciphers. -TID10100633 Result: Detailed result listings are provided in the Free 7-day Trial ^ back to top
	Vulnerability: SSL Server Has SSLv2 Enabled Vulnerability Qualys ID : 38139 Port : 443 Diagnosis: The Secure Socket Layer (SSL) protocol allows for secure communication between a client and a server. There are known flaws in the SSLv2 protocol. A man-in-the-middle attacker can force the communication to a less secure level and then attempt to break the weak encryption. The attacker can also truncate encrypted messages. These flaws have been fixed in SSLv3 (or TLSv1). Most servers (including all popular web-servers, mail-servers, etc.) and clients (including Web-clients like IE, Netscape Navigator and Mozilla and mail clients) support both SSLv2 and SSLv3. However, SSLv2 is enabled by default for backward compatibility. The following links provide more information about this vulnerability: ■ SSL Server Security Survey ■ SSL 3.0 Specification Consequences: An attacker can exploit this vulnerability to read secure communications or maliciously modify messages. Solution: Disable SSLv2.

	Typically, for Apache/mod_ssl, httpd.conf or ssl.conf should have the following lines: SSLProtocol -ALL +SSLv3 +TLSv1 SSLCipherSuite ALL:!aNULL:!ADH:!eNULL:!LOW:!EXP:RC4+RSA:+HIGH:+MEDIUM For Apache/apache_ssl, httpd.conf or ssl.conf should have the following line: SSLNoV2 How to disable SSLv2 on IIS : Microsoft Knowledge Base Article - 187498 How to Restrict the Use of Certain Cryptographic Algorithms and Protocols in Schannel.dll : Microsoft Knowledge Base Article - 245030 Result: Detailed result listings are provided in the Free 7-day Trial ^ back to top
	Vulnerability: Web Server HTTP Trace/Track Method Support Cross-Site Tracing Vulnerability Qualys ID : 86473 Port : 443 Diagnosis: A Web server was detected that supports the HTTP TRACE method. This method allows debugging and connection trace analysis for connections from the client to the Web server. Per the HTTP specification, when this method is used, the Web server echoes back the information sent to it by the client unmodified and unfiltered. Microsoft IIS web server uses an alias TRACK for this method, and is functionally the same. A vulnerability related to this method was discovered. A malicious, active component in a Web page can send Trace requests to a Web server that supports this Trace method. Usually, browser security disallows access to Web sites outside of the present site's domain. Although unlikely and difficult to achieve, it's possible, in the presence of other browser vulnerabilities, for the active HTML content to make external requests to arbitrary Web servers beyond the hosting Web server. Since the chosen Web server then echoes back the client request unfiltered, the response also includes cookie-based or Web-based (if logged on) authentication credentials that the browser automatically sent to the specified Web application on the specified Web server. The significance of the Trace capability in this vulnerability is that the active component in the page visited by the victim user has no direct access to this authentication information, but gets it after the target Web server echoes it back as its Trace response. Since this vulnerability exists as a support for a method required by the HTTP protocol specification, most common Web servers are vulnerable.

The exact method(s) supported, Trace and/or Track, and their responses are in the Results section below.

Consequences: If this vulnerability is successfully exploited, users of the Web server may lose their authentication credentials for the server and/or for the Web applications hosted by the server to an attacker. This may be the case even if the Web applications are not vulnerable to cross site scripting attacks due to input validation errors.

Solution:

Solutions for some of the common Web servers are supplied below. For other Web servers, please check your vendor's documentation.

Apache: Recent Apache versions have a Rewrite module that allows HTTP requests to be rewritten or handled in a specific way. Compile the Apache server with the mod_rewrite module. You might need to uncomment the 'AddModule' and 'LoadModule' directives in the httpd.conf configuration file. Add the following lines for each virtualhost in your configuration file (Please note that, by default, Rewrite configurations are not inherited. This means that you need to have Rewrite directives for each virtual host in which you wish to use it):

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
</IfModule>
```

With this configuration, Apache catches all TRACE requests, and replies with a page reporting the request as forbidden. None of the original request's contents are echoed back.

A slightly tighter fix is to use:

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteCond %{REQUEST_METHOD} !^(GET|POST|HEAD)$
RewriteRule .* - [F]
</IfModule>
```

Please note that RewriteEngine can be processor intensive and may impact the web server performance. The trace method can also be controlled by use of the TraceEnable directive, and track can be controlled through either the Limit or LimitExcept.

In the httpd.conf add or modify:

```
<Directory>
```

```
<limit TRACK>
```

deny from all

</limit>

...

</Directory>

TraceEnable Off

Microsoft IIS: Microsoft released **URLScan**, which can be used to screen all incoming requests based on customized rulesets. URLScan can be used to sanitize or disable the TRACE requests from the clients. Note that IIS aliases 'TRACK' to 'TRACE'. Therefore, if URLScan is used to specifically block the TRACE method, the TRACK method should also be added to the filter.

URLScan uses the 'urlscan.ini' configuration file, usually in \System32\InetSrv\URLScan directory. In that, we have two sections - AllowVerbs and DenyVerbs. The former is used if the UseAllowVerbs variable is set to 1, else (if its set to 0), the DenyVerbs are used. Clearly, either can be used, depending on whether we want a Default-Deny-Explicit-Allow or a Default-Allow-Explicit-Deny policy. To disallow TRACE and TRACK methods through URLScan, first remove 'TRACK', 'TRACE' methods from the 'AllowVerbs' section and add them to the 'DenyVerbs' section. With this, URLScan will disallow all 'TRACE' and 'TRACK' methods, and generate an error page for all requests using that method. To enable the changes, restart the 'World Wide Web Publishing Service' from the 'Services' Control Panel item.

Sun ONE/iPlanet Web Server: Here are the sun recommendations to **disable the trace method**.

For more details about other web servers : **Cert Advisory**.

Result:

Detailed result listings are provided in the [Free 7-day Trial](#)

[^ back to top](#)

Vulnerability:

Web Server HTTP Trace/Track Method Support Cross-Site Tracing Vulnerability
Qualys ID : 86473
Port : 80

Diagnosis:

A Web server was detected that supports the HTTP TRACE method. This method allows debugging and connection trace analysis for connections from the client to the Web server. Per the HTTP specification, when this method is used, the Web server echoes back the information sent to it by the client unmodified and unfiltered. Microsoft IIS web server uses an alias TRACK for this method, and is functionally the same.



2

A vulnerability related to this method was discovered. A malicious, active component in a Web page can send Trace requests to a Web server that supports this Trace method. Usually, browser security disallows access to Web sites outside of the present site's domain. Although unlikely and difficult to achieve, it's possible, in the presence of other browser vulnerabilities, for the active HTML content to make external requests to arbitrary Web servers beyond the hosting Web server. Since the chosen Web server then echoes back the client request unfiltered, the response also includes cookie-based or Web-based (if logged on) authentication credentials that the browser automatically sent to the specified Web application on the specified Web server.

The significance of the Trace capability in this vulnerability is that the active component in the page visited by the victim user has no direct access to this authentication information, but gets it after the target Web server echoes it back as its Trace response.

Since this vulnerability exists as a support for a method required by the HTTP protocol specification, most common Web servers are vulnerable.

The exact method(s) supported, Trace and/or Track, and their responses are in the Results section below.

Consequences: If this vulnerability is successfully exploited, users of the Web server may lose their authentication credentials for the server and/or for the Web applications hosted by the server to an attacker. This may be the case even if the Web applications are not vulnerable to cross site scripting attacks due to input validation errors.

Solution:

Solutions for some of the common Web servers are supplied below. For other Web servers, please check your vendor's documentation.

Apache: Recent Apache versions have a Rewrite module that allows HTTP requests to be rewritten or handled in a specific way. Compile the Apache server with the mod_rewrite module. You might need to uncomment the 'AddModule' and 'LoadModule' directives in the httpd.conf configuration file. Add the following lines for each virtualhost in your configuration file (Please note that, by default, Rewrite configurations are not inherited. This means that you need to have Rewrite directives for each virtual host in which you wish to use it):

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
</IfModule>
```

With this configuration, Apache catches all TRACE requests, and replies with a page reporting the request as forbidden. None of the original request's contents are echoed back.

A slightly tighter fix is to use:

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteCond %{REQUEST_METHOD} !^(GET|POST|HEAD)$
RewriteRule .* - [F]
</IfModule>
```

Please note that RewriteEngine can be processor intensive and may impact the web server performance. The trace method can also be controlled by use of the TraceEnable directive, and track can be controlled through either the Limit or LimitExcept.

In the httpd.conf add or modify:

```
<Directory>

<limit TRACK>

deny from all

</limit>

...

</Directory>
```

TraceEnable Off

Microsoft IIS: Microsoft released **URLScan**, which can be used to screen all incoming requests based on customized rulesets. URLScan can be used to sanitize or disable the TRACE requests from the clients. Note that IIS aliases 'TRACK' to 'TRACE'. Therefore, if URLScan is used to specifically block the TRACE method, the TRACK method should also be added to the filter.

URLScan uses the 'urlscan.ini' configuration file, usually in \System32\InetSrv\URLScan directory. In that, we have two sections - AllowVerbs and DenyVerbs. The former is used if the UseAllowVerbs variable is set to 1, else (if its set to 0), the DenyVerbs are used. Clearly, either can be used, depending on whether we want a Default-Deny-Explicit-Allow or a Default-Allow-Explicit-Deny policy. To disallow TRACE and TRACK methods through URLScan, first remove 'TRACK', 'TRACE' methods from the 'AllowVerbs' section and add them to the 'DenyVerbs' section. With this, URLScan will disallow all 'TRACE' and 'TRACK' methods, and generate an error page for all requests using that method. To enable the changes, restart the 'World Wide Web Publishing Service' from the 'Services' Control Panel

	item. Sun ONE/iPlanet Web Server: Here are the sun recommendations to disable the trace method. For more details about other web servers : Cert Advisory. Result: Detailed result listings are provided in the Free 7-day Trial ^ back to top
 2	Vulnerability: Netscape/OpenSSL Cipher Forcing Bug Qualys ID : 38284 Port : 443 Diagnosis: Netscape's SSLv3 implementation had a bug where if a SSLv3 connection is initially established, the first available cipher is used. If a session is resumed, a different cipher may be chosen if it appears in the passed cipher list before the session's current cipher. This bug can be used to change ciphers on the server. OpenSSL contains this bug if the SSL_OP_NETSCAPE_REUSE_CIPHER_CHANGE_BUG option is enabled during runtime. This option was introduced for compatibility reasons. The problem arises when different applications using OpenSSL's libssl library enable all compatibility options including SSL_OP_NETSCAPE_REUSE_CIPHER_CHANGE_BUG, thus enabling the bug. Consequences: A malicious legitimate client can enforce a ciphersuite not supported by the server to be used for a session between the client and the server. This can result in disclosure of sensitive information. Solution: This problem can be fixed by disabling the SSL_OP_NETSCAPE_REUSE_CIPHER_CHANGE_BUG option from the options list of OpenSSL's libssl library. This can be done by replacing the SSL_OP_ALL definition in the openssl/ssl.h file with the following line: <pre>#define SSL_OP_ALL (0x00000FFFL ^ SSL_OP_NETSCAPE_REUSE_CIPHER_CHANGE_BUG)</pre> The library and all programs using this library need to be recompiled to ensure that the correct OpenSSL library is used during linking. Result: Detailed result listings are provided in the Free 7-day Trial ^ back to top
	Vulnerability: Global User List Qualys ID : 45002 Port : N/A Diagnosis:

 2	This is the global system user list, which was retrieved during the scan by exploiting one or more vulnerabilities. The Qualys IDs for the vulnerabilities leading to the disclosure of these users are also given in the Result section. Each user will be displayed only once, even though it may be obtained by using different methods. Consequences: These common account(s) can be used by a malicious user to break-in the system via password bruteforcing. Solution: To prevent your host from being attacked, do one or more of the following: ■ Remove (or rename) unnecessary accounts ■ Shutdown unnecessary network services ■ Ensure the passwords to these accounts are kept secret ■ Use a firewall to restrict access to your hosts from unauthorized domains Result: Detailed result listings are provided in the Free 7-day Trial ^ back to top
 2	Vulnerability: SSL Certificate - Signature Verification Failed Vulnerability Qualys ID : 38173 Port : 443 Diagnosis: An SSL Certificate associates an entity (person, organization, host, etc.) with a Public Key. In an SSL connection, the client authenticates the remote server using the server's Certificate and extracts the Public Key in the Certificate to establish the secure connection. The authentication is done by verifying that the public key in the certificate is signed by a trusted third-party Certificate Authority. If a client is unable to verify the certificate, it can abort communication or prompt the user to continue the communication without authentication. Consequences: By exploiting this vulnerability, man-in-the-middle attacks in tandem with DNS cache poisoning can occur. Exception: If the server communicates only with a restricted set of clients who have the server

	certificate or the trusted CA certificate, then the server or CA certificate may not be available publicly, and the scan will be unable to verify the signature. Solution: Please install a server certificate signed by a trusted third-party Certificate Authority. Result: Detailed result listings are provided in the Free 7-day Trial ^ back to top
1	Vulnerability: Apache Web Server ETag Header Information Disclosure Weakness Qualys ID : 86477 Port : 80 Diagnosis: The Apache HTTP Server is a popular, open-source HTTP server for multiple platforms, including Windows, Unix, and Linux. A cache management feature for Apache makes use of an entity tag (ETag) header. When this option is enabled and a request is made for a document relating to a file, an ETag response header is returned containing various file attributes for caching purposes. ETag information allows subsequent file requests to contain specific information, such as the file's inode number. A weakness has been found in the generation of ETag headers under certain configurations implementing the FileETag directive. Among the file attributes included in the header is the file inode number that is returned to a client. In Apache Versions 1.3.22 and earlier, it's not possible to disable inodes in ETag headers. In later versions, the default behavior is to release this sensitive information. Consequences: This vulnerability poses a security risk, as the disclosure of inode information may aid in launching attacks against other network-based services. For instance, NFS uses inode numbers to generate file handles. Solution: OpenBSD has released a patch that fixes this vulnerability. After installing the patch, inode numbers returned from the server are encoded using a private hash to avoid the release of sensitive information. Customers are advised to upgrade to the latest version of Apache. In Apache Version 1.3.23 and later, it's possible to configure the FileETag directive to generate ETag headers without inode information. To do so, include "FileETag -INode" in the Apache server configuration file for a specific subdirectory. In order to fix this vulnerability globally, for the Web server, use the option "FileETag None". Use the option "FileETag MTime Size" if you just want to remove the Inode information. Result: Detailed result listings are provided in the Free 7-day Trial

[^ back to top](#)

7-Day Trial

With a Free 7-day Trial of QualysGuard, you will have in-depth information on your vulnerabilities and possible threats found on your host. Such information includes services identified, operating systems, open ports, and server banners. Request your [Free 7-day Trial](#) now.