Disable older protocols and cipher suites

Who is this article for?

✔ Code42 for Enterprise
See product plans and features (https://support.code42.com/Terms_and_conditions/Code42_customer_support_resources/Code42_product_plans)

✔ CrashPlan for Small Business

CrashPlan for Small Business, no.

Code42 for Enterprise, yes.

Link: Product plans and features.

This article applies to versions 6 and 7.

Other available versions:

Version 5 (https://support.code42.com/Administrator/5/Planning_and_installing/Security_hardening_best_practices/Disable_older_protocols_and_cipher_suites)

Overview

This article describes how to disable older Secure Socket Layer (SSL) and Transport Layer Security (TLS) security protocols (HTTPS://en.wikipedia.org/wiki/Transport_Layer_Security#History_and_development) and cipher suites (https://en.wikipedia.org/wiki/Cipher_suite) that are known to possess security vulnerabilities. Blacklisting specific protocols and cipher suites makes your Code42 environment more secure against attacks designed to exploit these vulnerabilities.
Server security requires a CA-signed certificate and the TLS protocol

Reliable security of any production web server requires an SSL certificate signed by a trusted certificate authority (CA) and enforced use of the TLS protocol (that is, HTTPS, not HTTP).


• By default, your authority server uses a self-signed certificate and TLS. That provides for encrypting client-server traffic.
• Adding a CA-signed certificate (https://support.code42.com/Administrator/6/Configuring/Install_a_CA-signed_SSL_certificate_for_HTTPS_con... provides further security by confirming your server’s identity to clients. It prevents attackers from acquiring client data through counterfeit servers and encryption keys.
• Never reconfigure a production server to use HTTP, rather than TLS and HTTPS.
• Configuring Code42 servers and apps to use strict TLS validation (https://support.code42.com/Administrator/6/Planning_and_installing/Security_hardening_best_practices/Configure_TLS_strict_certificate_validation) further ensures the security of client-server connections.
• Configuring Code42 servers to use an HTTPS Strict Transport Security (HSTS) (https://support.code42.com/Administrator/6/Planning_and_installing/Security_hardening_best_practices/Enforce_HTTPS_access) response header further prevents unencrypted browser access to administration consoles.

Background

SSL and its successor, TLS, are protocols used to secure data transfer between clients and servers by means of encryption and authentication. These protocols use cipher suites to provide encryption for secure connection and data transport.

However, even if you are using TLS, you still must be careful to use only secure cipher suites. Older cipher suites may allow attacks of data in transit. You can issue commands in the Code42 command line interface (CLI) to disable not only specific protocols but also specific cipher suites.
Considerations

To utilize the approved protocols and cipher suites in your Code42 environment, we recommend you stay up-to-date (https://support.code42.com/Administrator/Upgrade) on our Code42 software versions. The default exclusions of protocols and cipher suites in Code42 software provide you adequate security. However, you can disable additional older protocols and cipher suites to strengthen security as needed. Be advised that Code42 may add additional exclusions in future versions that may differ from what you set.

Protocols

By default, protocols SSL 3.0 and SSL 2.0 are disabled, depending on your specific version of Code42 software. Use the prop.show c42.https.exclude.protocols command to see the disabled protocols.

To make your Code42 environment more secure, consider disabling the following protocols:

- SSL 2.0
- SSL 3.0
- TLS 1.0
- TLS 1.1

Cipher suites

By default, a large set of cipher suites are disabled in Code42 software. Use the prop.show c42.https.exclude.ciphers command to see the disabled cipher suites.

To make your Code42 environment more secure, consider disabling the following cipher suites:

- MD5
- SHA-1
- Null, export grade, or otherwise weak ciphers (weaker than AES-128)

Before you begin

We recommend you run an analysis against a web site you deem to be secure to find out the protocols and cipher suites that it uses. This may help you determine the protocols and cipher suites to allow or to exclude in your Code42 environment.

There are numerous tools you can use to list the SSL and TLS cipher suites a particular web site offers such as SSL Labs (https://www.ssllabs.com/ssltest/index.html). After you perform steps in the following sections to disable specific protocols and cipher suites in your Code42 environment, you can use this same kind of analysis to verify that your Code42 environment uses only those protocols and cipher suites that you specified.
Step 1: Disable protocols

1. **Sign in to the administration console**
   ![Sign in to the administration console](https://support.code42.com/Administrator/6/Administration_console_reference/01_Administration_Console_Overview#Access_the_administration_console).

2. Double-click the logo in the upper-left corner of the administration console.
   *The command-line interface appears in the administration console.*

3. To verify the current protocols exclusion setting in your Code42 environment, enter the following `prop.show` command:
   ![prop.show](https://support.code42.com/Administrator/6/Administration_console_reference/Administration_console_command-line_interface_reference#Prop.show):

   ```
   prop.show c42.https.exclude.protocols
   ```

   The default list of excluded ciphers are shown, for example:

   ```
   c42.https.exclude.protocols SSLv2,SSLv3
   ```

4. To set the list of protocols to exclude, enter the following `prop.set` command:
   ![prop.set](https://support.code42.com/Administrator/6/Administration_console_reference/Administration_console_command-line_interface_reference#Prop.set):

   ```
   prop.set c42.https.exclude.protocols "<protocols>" save all
   ```

   Replace `<protocols>` with a comma-separated list of protocols that you no longer want to allow for communication with the Code42 environment, for example:

   ```
   prop.set c42.https.exclude.protocols "SSLv2,SSLv3,TLSv1,TLSv1.1" save all
   ```

5. Restart the server using the `node.restart` command:
   ![node.restart](https://support.code42.com/Administrator/6/Administration_console_reference/Administration_console_command-line_interface_reference#Node.restart)

6. To verify the new protocols settings in your Code42 environment, enter the `prop.show` command:
   ![prop.show](https://support.code42.com/Administrator/6/Administration_console_reference/Administration_console_command-line_interface_reference#Prop.show)

7. (Optional) To verify that the protocol exclusion works as expected, run an analysis on your Code42 environment of the protocols and cipher suites in use.

Step 2: Disable cipher suites

1. In the administration console, double-click the logo in the upper-left corner.
   *The command-line interface appears in the administration console.*

2. To verify the current cipher suites exclusion setting in your Code42 environment, enter the following `prop.show` command:
   ![prop.show](https://support.code42.com/Administrator/6/Administration_console_reference/Administration_console_command-line_interface_reference#Prop.show)
prop.show c42.https.exclude.ciphers

The default list of excluded ciphers are shown, for example:

```
c42.https.exclude.ciphers .*NULL.*,.*RC4.*,.*MD5.*,.*DES.*,.*DSS.*,TLS_EMPTY_RENEGOTIATION_INFO_SCSV
```

We recommend you copy this default list of excluded ciphers so you can add to it when you create your new exclusion list.

3. To set the list of ciphers to exclude, enter the following prop.set (https://support.code42.com/Administrator/6/Administration_console_reference/Administration_console_command-line_interface_reference#Prop.set) command:

```
prop.set c42.https.exclude.ciphers "<cipher suites>" save all
```

Replace `<cipher suites>` with a comma-separated list of cipher suites that you no longer want to allow for communication encryption within the Code42 environment. We recommend you start with the default set of ciphers obtained in the previous set and then add to additional ciphers to it.

4. Restart the server using the node.restart (https://support.code42.com/Administrator/6/Administration_console_reference/Administration_console_command-line_interface_reference#Node.restart) command:

```
node.restart
```

5. To verify the new cipher settings in your Code42 environment, enter the prop.show c42.https.exclude.ciphers command.

6. Verify that the cipher exclusion works as expected by running an analysis on your Code42 server of the protocols and cipher suites in use.

**Example cipher exclusion list**

Following is the default list of cipher suites that are disabled in a Code42 environment:

```
.*NULL.*,.*RC4.*,.*MD5.*,.*DES.*,.*DSS.*,TLS_EMPTY_RENEGOTIATION_INFO_SCSV
```

**Enable cipher suites**

If needed to provide compatibility with older systems, you can enable older cipher suites by either:

- Removing the cipher suite from the list of excluded ciphers
- Including the cipher suite
Remove the cipher suite from the list of excluded ciphers

1. Remove the cipher suite from the exclusion list:
   prop.set c42.https.exclude.ciphers "<cipher suites>" save all
2. Restart the server using the node.restart (https://support.code42.com/Administrator/6/Administration_console_reference/Administration_console_command-line_interface_reference#Node.restart) command:
   node.restart
3. To verify the new cipher exclusion setting in your Code42 environment, run the following command:
   prop.show c42.https.exclude.ciphers
4. Verify that the cipher exclusion works as expected by running an analysis on your Code42 server of the protocols and cipher suites in use.

Include the cipher suite

1. Include the cipher suite with the following command:
   prop.set c42.https.include.ciphers "<cipher suites>" save all
2. Restart the server using the node.restart (https://support.code42.com/Administrator/6/Administration_console_reference/Administration_console_command-line_interface_reference#Node.restart) command:
   node.restart
3. To verify the new cipher inclusion setting in your Code42 environment, run the following command:
   prop.show c42.https.include.ciphers
4. Verify that the cipher inclusion works as expected by running an analysis on your Code42 server of the protocols and cipher suites in use.

Exclusion takes precedence
Values set by the c42.https.exclude.ciphers property override values set by the c42.https.include.ciphers property. So if a cipher suite is listed in both, the exclusion takes precedence.

External resources

- acunetix: Recommendations for TLS/SSL Cipher Hardening (https://www.acunetix.com/blog/articles/tls-ssl-cipher-hardening/)
- Java cipher suites for JDK 8 (https://docs.oracle.com/javase/8/docs/technotes/guides/security/SunProviders.html#SunJSSEProvider)
Related topics

- Configure TLS strict certificate validation (https://support.code42.com/Administrator/6/Planning_and_installing/Security_hardening_best_practices/Configure_TLS_strict_certificate_validation)
- TLS messaging (https://support.code42.com/Administrator/6/Configuring/TLS_messaging)