SSL v3 and TLS v1 Protocol Weak CBC Mode Vulnerability

Contents

Introduction

This document describes how to disable Cipher Block Chaining (CBC) Mode Ciphers on the Cisco Email Security Appliance (ESA). A security audit/scan might report that an ESA has a Secure Sockets Layer (SSL) v3/Transport Layer Security (TLS) v1 Protocol Weak CBC Mode Vulnerability.

> **Attention**: If you are running older code of AsyncOS for Email Security, it is recommended to upgrade to version 11.0.3 or newer. Please review the Cisco Email Security Release Notes for our latest versions and information. If you need further assistance with upgrades or disabling ciphers, please open a support case.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on AsyncOS for Email Security (any revision), a Cisco ESA, and a virtual ESA.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Background Information

- Payment Card Industry Data Security Standard (PCI DSS) compliance requires CBC Ciphers to be disabled.
- A security audit/scan has identified a potential vulnerability with SSL v3/TLS v1 protocols that use CBC Mode Ciphers.

  Tip: SSL Version 3.0 (RFC-6101) is an obsolete and insecure protocol. There is a vulnerability in SSLv3 CVE-2014-3566 known as Padding Oracle On Downgraded Legacy Encryption (POODLE) attack, Cisco bug ID CSCur27131. The recommendation is to disable SSL v3 while you change the ciphers and use TLS only, and select option 3 (TLS v1). Review the provided Cisco bug ID CSCur27131 for complete details.

SSL v3 and TLS v1 protocols are used in order to provide integrity, authenticity, and privacy to other protocols such as HTTP and Lightweight Directory Access Protocol (LDAP). They provide these services with the use of encryption for privacy, x509 certificates for authenticity, and one-way encryption functionality for integrity. In order to encrypt data, SSL and TLS can use block ciphers which are encryption algorithms that can encrypt only a fixed block of original data to an encrypted block of the same size. Note that these ciphers will always obtain the same resulting block for the same original block of data. In order to achieve a difference in the output, the output of the encryption is XORed with yet another block of the same size referred to as initialization vectors (IV). CBC uses one IV for the initial block and the result of the previous block for each subsequent block in order to obtain the difference in the output of block cipher encryption.

In SSL v3 and TLS v1 implementation, the choice CBC mode usage was poor because the entire traffic shares one CBC session with a single set of initial IVs. The rest of the IVs are, as mentioned previously, results of the encryption of the previous blocks. The subsequent IVs are available to the eavesdroppers. This allows an attacker with the capability to inject arbitrary traffic into the plain-text stream (to be encrypted by the client) in order to verify their guess of the plain-text that precedes the injected block. If the attackers' guess is correct, then the output of the encryption is the same for two blocks.

For low entropy data, it is possible to guess the plain-text block with a relatively low number of attempts. For example, for data that has 1000 possibilities, the number of attempts can be 500.

Requirements

There are several requirements that must be met in order for the exploit to work:

1. The SSL/TLS connection must use one of the block encryption ciphers that use CBC modes, such as DES or AES. Channels that use stream ciphers such as RC4 are not subject to the flaw. A large proportion of SSL/TLS connections use RC4.
2. The vulnerability can only be exploited by someone that intercepts data on the SSL/TLS connection, and also actively sends new data on that connection. The exploitation of the flaw causes the SSL/TLS connection to be terminated. The attacker must continue to monitor and use new connections until enough data is gathered to decrypt the message.
3. Since the connection is terminated each time, the SSL/TLS client must be able to continue to reestablish the SSL/TLS channel long enough for the message to be decrypted.
4. The application must resend the same data on each SSL/TLS connection that it creates and the listener must be able to locate it in the data stream. Protocols like IMAP/SSL that have a fixed set of messages to log in meet this requirement. General web browsing does not.

Threat

The CBC vulnerability is a vulnerability with TLS v1. This vulnerability has been in existence since early 2004 and was resolved in later versions of TLS v1.1 and TLS v1.2.

Prior to AsyncOS 9.6 for Email Security, the ESA utilizes TLS v1.0 and CBC mode ciphers. With the release of AsyncOS 9.6, the ESA introduces TLS v1.2. Still, CBC mode ciphers can be disabled, and only RC4 ciphers can be used which are not subject to the flaw.

In addition, if SSLv2 is enabled this can trigger a false positive for this vulnerability. It is very important that SSL v2 be disabled.

Solution

> **Attention**: If you are running older code of AsyncOS for Email Security, it is recommended to upgrade to version 11.0.3 or newer. Please review the Cisco Email Security Release Notes for our latest versions and information. If you need further assistance with upgrades or disabling ciphers, please open a support case.

Disable CBC mode ciphers in order to leave only RC4 ciphers enabled. Set the device to only use TLS v1, or TLS v1/TLS v1.2:

1. Log in to the CLI.
2. Enter the command **sslconfig**.
3. Enter the command **GUI**.
4. Choose option number 3 for "TLS v1", or as listed in AsyncOS 9.6 "TLS v1/TLS v1.2".
5. Enter this cipher:
   `MEDIUM:HIGH:-SSLv2:-aNULL:@STRENGTH:-EDH-RSA-DES-CBC3-SHA:-EDH-DSS-DES-CBC3-SHA:-DES-CBC3-SHA`
6. Enter the command: **INBOUND**.
7. Choose option number 3 for "TLS v1", or as listed in AsyncOS 9.6 "TLS v1/TLS v1.2".
8. Enter this cipher:
   `MEDIUM:HIGH:-SSLv2:-aNULL:@STRENGTH:-EDH-RSA-DES-CBC3-SHA:-EDH-DSS-DES-CBC3-SHA:-DES-CBC3-SHA`
9. Enter the command **OUTBOUND**.
10. Choose option number 3 for "TLS v1", or as listed in AsyncOS 9.6 "TLS v1/TLS v1.2".
11. Enter this cipher:
    `MEDIUM:HIGH:-SSLv2:-aNULL:@STRENGTH:-EDH-RSA-DES-CBC3-SHA:-EDH-DSS-DES-CBC3-SHA:-DES-CBC3-SHA`
12. Press **Enter** until you return to the hostname prompt.
13. Enter the command **commit**.
14. Finalize committing your changes.

The ESA is now configured to only support TLS v1, or TLSv1/TLS v1.2, with RC4 ciphers while it disallows any CBC filters.

Here is the list of ciphers used when you set RC4:-SSLv2. Note that there are no CBC mode ciphers in the list.

```
ECDHE-RSA-RC4-SHA SSLv3 Kx=ECDH Au=RSA Enc=RC4(128) Mac=SHA1
```

```
ECDHE-ECDSA-RC4-SHA SSLv3 Kx=ECDH Au=ECDSA Enc=RC4(128) Mac=SHA1
ADH-RC4-MD5 SSLv3 Kx=DH Au=None Enc=RC4(128) Mac=MD5
RC4-SHA SSLv3 Kx=RSA Au=RSA Enc=RC4(128) Mac=SHA1
RC4-MD5 SSLv3 Kx=RSA Au=RSA Enc=RC4(128) Mac=MD5
PSK-RC4-SHA SSLv3 Kx=PSK Au=PSK Enc=RC4(128) Mac=SHA1
EXP-ADH-RC4-MD5 SSLv3 Kx=DH(512) Au=None Enc=RC4(40) Mac=MD5 export
EXP-RC4-MD5 SSLv3 Kx=RSA(512) Au=RSA Enc=RC4(40) Mac=MD5 export
```

While this exploit is of very low concern due to its complexity and requirements to exploit, the performance of these steps is a great safeguard for the prevention of possible exploits, as well as to pass strict security scans.

Related Information

- **Cisco Email Security Appliance - End-User Guides**
- **Technical Support & Documentation - Cisco Systems**