Early CCS Attack Analysis

June 6, 2014

1 Background

The OpenSSL project released a security advisory on June 5th, 2014[1] for several newly patched vulnerabilities. Among these is CVE-2014-0224, an attack affecting a two susceptible OpenSSL endpoints in the presence of a network attacker.

This vulnerability was discovered by Masashi Kikuchi of Lepidum.[2] It is not a flaw in the SSL or TLS protocol itself, but rather an implementation error in the OpenSSL library.

In the client configuration, essentially all versions are vulnerable:

- OpenSSL versions at or below 0.9.8y (inclusive)
- OpenSSL versions at or below 1.0.0l (inclusive)
- OpenSSL versions at or below 1.0.1g (inclusive)

In the server configuration, the following versions are vulnerable:

- OpenSSL versions at or below 1.0.1g (inclusive)

Note that the 0.9.8 and 1.0.0 branches of OpenSSL, when used in the server configuration, are not vulnerable; and that the 1.0.2-beta1 branch *is* vulnerable in all configurations.

2 Attack Vectors & Impact

To exploit this bug, both the client and server must be running vulnerable versions of OpenSSL. If either the client or server are patched, an attack will fail when the patched end of the connection fails after the attacker forces the vulnerable state transition. The attacker must also be able to perform an active attack on the network connection.

A successful exploit allows an attacker to intercept, decrypt, and modify the underlying plaintext traffic between vulnerable clients and servers. In standard configurations, this impersonation is undetectable to the victim parties attempting to communicate. IDS or network analysis tools with specialized rules may be able to detect attempted exploitation.

3 Recommendations

This vulnerability is patched in the following versions:

- OpenSSL 0.9.8 is patched in 0.9.8za
- OpenSSL 1.0.0 is patched in 1.0.0m

[1] https://www.openssl.org/news/secadv_20140605.txt

[2] http://ccsinjection.lepidum.co.jp/index.html

- OpenSSL 1.0.1 is patched in 1.0.1h

Users of these branches should upgrade to the latest version immediately. The 1.0.2 branch (still a beta release) is as yet unpatched. Because the client configuration is also vulnerable, it is also important to upgrade machines that do not run listening services, as client programs that connect to vulnerable machines (such as partners or third parties for updates or business purposes) can still be attacked.

3.1 Operating System Updates

- Windows - If OpenSSL is in use by applications, it will need to be updated through whatever update mechanism is available. OpenSSL can be built for Windows,[3] but the commonly provided installers are not up to date as of this writing.
- Ubuntu - See USN-2232-1[4]
- Debian - See DSA-2950-1[5]
- Red Hat Enterprise Linux - See RHSA-2014-0625[6] and RHSA-2014-0626[7]
- CentOS 5 - See CESA-2014:0626[8]
- CentOS 6 - See CESA-2014:0625[9] and CESA-2014:0626[10]
- Gentoo Linux - the OpenSSL 1.0.1h package is available
- FreeBSD - See FreeBSD-SA-14:14.openssl [11]
- AWS Infrastructure - Refer to https://aws.amazon.com/security/security-bulletins/openssl-security-advisory/

After OpenSSL has been updated, applications or servers that are running will need to be restarted. The following command will identify processes using the OpenSSL library:

```
$ lsof | grep libssl
```

3.2 Common Applications

Many popular applications use SSL/TLS, but not all of them use the OpenSSL library.

- OpenVPN - OpenVPN *is* affected. If on Windows, upgrade to the 1002 release[12] which bundles the patched 1.0.1h library. An update for the popular Tunnelblick client on Mac is still in development[13]
- Browsers - On the desktop, neither Internet Explorer, Chrome, Firefox, nor Safari use OpenSSL and are not affected.
- Mobile Applications - Certain more complex mobile applications bundle their own version of OpenSSL, which may be vulnerable. Applying mobile application updates that are available is recommended.

[3] http://git.openssl.org/gitweb/?p=openssl.git;a=blob;f=INSTALL.W32;h=80e538273e996b61ae79662719bb4972f232df42;hb=HEAD
[4] http://www.ubuntu.com/usn/usn-2232-1/
[5] https://www.debian.org/security/2014/dsa-2950
[6] https://rhn.redhat.com/errata/RHSA-2014-0625.html
[7] https://rhn.redhat.com/errata/RHSA-2014-0626.html
[8] http://lists.centos.org/pipermail/centos-announce/2014-June/020346.html
[9] http://lists.centos.org/pipermail/centos-announce/2014-June/020344.html
[10] http://lists.centos.org/pipermail/centos-announce/2014-June/020345.html
[11] http://www.freebsd.org/security/advisories/FreeBSD-SA-14%3A14.openssl.asc
[12] https://forums.openvpn.net/topic16039.html
[13] https://code.google.com/p/tunnelblick/source/list

- Third Party Applications on Windows - It is common for applications running on Windows, especially applications that are available for other platforms as well, to use a version of OpenSSL compiled for Windows. Contact the support channels of the vendor for more information.
- Daemons & Services on Linux - It is most common for applications to use the system-provided OpenSSL library. For specialty or proprietary applications, contact the support channels of the vendor.
- Third Party Appliances - Most appliances are built on a Linux platform, which usually includes a version of the OpenSSL library. Contact the support channels of the vendor for more information.

For third party applications installed on self-managed systems, it is often possible to examine the running processes from a root or Administrator-level context to determine if they use any open source libraries such as OpenSSL, and if so, what versions.

4 Technical Details

The Transport Layer Security (TLS) protocol is a widely deployed mechanism allowing two parties to establish a secure channel for communication. It attempts to provide integrity, confidentiality, and authenticity; secure messages cannot be read or modified without detection. (In certain configurations, the use of Anonymous of NULL ciphersuites may opt out of authentication or confidentiality.)

To establish a TLS session, two parties acting respectively as client and server initiate a “handshake”. This allows them to agree on shared secret material used to secure future communication. As specified, the handshake is resilient to malicious third parties; an attacker cannot recover the secret or influence its derivation.

This particular bug is in the State Machine of TLS. A State Machine is model that defines specific states a system or program can be, and what valid transitions can be made between them. For example, a traffic light in the United States moves from Green to Yellow to Red, and then to Green. A state where both the Green and Red lights are lit is invalid, and the transition from Yellow to Green is also invalid.

Implementation mistakes in the OpenSSL library allow an attacker to cause an invalid state transition in TLS. In particular, an attacker can inject messages into the handshake to fix the session secret to a known value. This allows them to decipher and modify messages at their leisure, breaking the guarantees of TLS. The client and server will be unable to detect this attack except by careful examination of the handshake transcript.

For more technical information, see Masashi Kikuchi[14] and Adam Langley’s[15] posts on the subject.

5 Testing External & Internal Infrastructure

A tool to test internal infrastructure is in development, and an early version is available at https://github.com/iSECPartners/ccs-testing-tool. SSLLabs (https://ssllabs.com) is in development of a tool to test external infrastructure, this document will be updated as these develop.

6 About

NCC Group provides high-end application and network security services including penetration testing, code and architecture review, threat modeling, Secure Development Lifecycle creation, and cryptographic design and implementation services.

[14] http://ccsinjection.lepidum.co.jp/blog/2014-06-05/CCS-Injection-en/index.html

[15] https://www.imperialviolet.org/2014/06/05/earlyccs.html