Heartbleed: A Vulnerability of OpenSSL

Sean Conway
Graham Burek
Brad Harris

9 December 2016

1 Introduction and Overview

Breaches in software systems are nothing new in a world that places a heavy and ever growing importance on computers and their processes. One such breach is Heartbleed, a vulnerability that exploits an improper implementation of code in an extension to a popular TLS/SSL library, OpenSSL.

While it is unclear how prevalent Heartbleed's use was before it was discovered, there are a few recorded cases of its use, discussed in depth later. However, any applications that use OpenSSL for cryptographic services such as SSL/TLS for their own applications and services are at risk.

Background

Transport Layer Security (TLS) and Secure Sockets Layer (SSL) are transport-layer protocols that provide a secure link between a client and server. They secure the communication channel between two hosts using encryption. TLS, a newer version of SSL, operates by using two 'sub-protocols': The TLS Record Protocol, as well as the TLS Handshake Protocol. [1] The Record sub-protocol provides a secure way for the Handshake sub-protocol to create a secure connection. Once this connection is created, the two hosts can communicate securely.

However, sometimes the secure TLS connection must be renegotiated. Renegotiation is the process by which a new TLS session is created while being protected by the previous one. [2] For example, renegotiation might occur to keep the connection alive if no data is being sent over the connection.

In fact, this is the reason the heartbeat extension, which Heartbleed exploits, exists. Because TLS provides no way to keep an underlying connection alive, the heartbeat extension sends out a continuous message (termed a heartbeat) to ensure the other host is still there. [3]

The Exploit

The Heartbleed exploit uses a bug in versions 1.01a through 1.01f of the heartbeat extension code found in OpenSSL. [4] There are some canonical examples of the original exploit, such as the Canadian Revenue Agency (CRA) breach mentioned below, but considering a successful attack would leave little to no trace, it is likely that many attacks have gone unreported. The attack's general workings are as follows:

1. Heartbeat packets have a length and a payload field.
2. The heartbeat works by asking the other host to send back the same payload, with the given length.
3. Because of a missing bounds check in the heartbeat code, an attacker can request a length longer than the payload they provide. [5]
4. Because of this, the code allocates more memory than is used for the payload, exposing and sending random nearby data from memory to the other host along with the short payload. This is an example of a buffer over-read vulnerability.

While each attack can only retrieve 64KB of information from memory, an attacker can send an unlimited number of malformed heartbeat packets, allowing them to eventually gain access to sensitive information. In addition, the attack can come from **any** host (client or server) that is able to establish a TLS connection with another host running the exploitable version of OpenSSL.

In addition, the attacks cannot be traced or monitored. To the attacked host, the attacks look like normal heartbeat requests.

Extent of Damage

The exploit was active from March 2012 to April 2014 before a fix was released (1.01g). This fix for OpenSSL is not an example of continuous improvement, but rather is an emergency fix deployed in response to the vulnerability.

Although the effects of Heartbleed are not easily measurable, it has the potential to be catastrophic. Approximately 64 percent of all running internet servers had the potential to be affected by the bug, as well as 17 percent of secured websites. [5] In the worst case scenario, if a server's private key is corrupted, the site's current and past traffic would be vulnerable to complete decryption until the key is replaced and the vulnerability patched. Even if the private key is not obtained, getting access to password protected devices using Heartbleed can set the stage for different attacks. For example, if private information is revealed, phishing and blackmail attacks can be conducted. In addition, it might be possible to gain access to a organization's security certificate to perform man-in-the-middle attacks, or to tamper with its cookies to spoof a user's identity.

2 Attack Scenarios

Goals

The goals of a Heartbleed attack can be narrowed down to two main objectives: it creates a vulnerability that allows for a simple leak of data (usernames, passwords, etc), and it allows for the ability to take over a private key and gain access to web traffic using it. [5]

The resources needed to exploit a Heartbleed attack are a working knowledge of routing, as well as packet crafting software (or another way to send malformed heartbeat packets). As stated before, a certificate is not even entirely necessary to exploit the attack.

Attack 1: Canadian Revenue Agency (CRA)

Description

An instance of the Heartbleed vulnerability wreaked havoc among CRA Representatives as 900 social insurance numbers (equivalent of US Social Security Numbers) of Canadian taxpayers were stolen as part of a cyber attack. [6] These insurance numbers were completely removed from the system, and the CRA was forced to manually filter through their databases in search of any other significant damage. Affected individuals were told that they would receive a formal letter in the mail, containing free credit protection as well as a dedicated contact hotline. They were instructed to ignore any phone calls or emails claiming to be from the CRA as they would most likely be scams. [6]

The CRA breach was very expensive. Because of the breach, the agency had to extend tax deadlines for 5 days, potentially costing millions of dollars in tax revenue. [7]

Resources

Due to the nature of Heartbleed and its design, no extra tools besides an internet connection and an understanding of programming (as well as packet crafting) are necessary to exploit this vulnerability.

Goals

The goal of the attack on the CRA was to demonstrate the CRA's vulnerability to the Heartbleed exploit. It was carried out by a 19-year old computer science student. [8] The student said "he didn't mean any harm" when he undertook the exploit, but because he broke into a government agency's

database, he caused at least moderate distress to the administrators and the system's users. [8] Although a Heartbleed attack is much more simplistic than a full on breakthrough of a system, the fact that the student was able to take advantage of the vulnerability demonstrated that he had strong technical knowledge and had scoped out the CRA's system.

Collateral Results

Some collateral results of this attack were the updates of the CRA's software systems and security implementations, and the eventual house arrest of Stephen Solis-Reyes. [8]

Recommended Mitigation

The attack could have been prevented by not updating to the most recent version of OpenSSL. Unfortunately, most users will update to newer versions as fast as possible, since they often patch security vulnerabilities. Post attack, an update to OpenSSL (version 1.01g or higher), as well as the revocation and reissuing of private keys, would be the best course of action.

Attack 2: Cupid

Description

Designed in a way similar to Heartbleed, Cupid exploits a vulnerability in OpenSSL. It primarily affects wireless networks and mobile users in a two-part attack. Cupid gives wireless networks the ability to deploy 'evil networks' and discreetly send connected devices malicious information. [9]

Unlike the original Heartbleed bug, Cupid exploits a vulnerability in the TLS mechanism for the Extensible Authentication Protocol (EAP-TLS). Until recently, EAP-TLS implementations required a client-side X.509 certificate, causing its use to be mostly limited to enterprises, as setting up certificates for home and public-access networks is unwieldy. [10] Therefore, attacks with Cupid were most common against enterprise systems.

The exploit infects client devices as they attempt to connect to corporate networks. Cupid is also problematic for Linux users, since these networks can be deployed with a Linux program (hostapd) for setting up an access point. [11]

Resources

In order for Cupid to work, a threat agent needs access to an EAP-TLS protected network. As mentioned earlier, this is generally only found in a corporate setting. In addition, the attacker needs the cupid software extensions, which send the malformed heartbeat requests.

Goals

This attack is extremely similar to the original Heartbleed exploit, and, therefore, many of the goals of the two exploits resonate with each other. Both of the attacks are done using the OpenSSL framework, and both generally use Wi-Fi networks to exploit their targets. However, the Cupid exploit is generally only executed in a corporate setting, because home and public network users do not use EAP-TLS authentication mechanisms. Attackers using Cupid would have a high level of motivation, most likely looking to infiltrate a corporation to destroy software infrastructure or steal private information.

Collateral Results

The attacker might get more information then they were looking for, or the attacker might use credentials stolen using Cupid to infiltrate more systems.

Recommended Mitigation

The best mitigation would be to update any devices using OpenSSL to a fixed newer version (version 1.01g or higher), and to change private keys.

3 Summary

Although Heartbleed was a simple programming error, it did significant damage. Precious information was able to be collected through the vulnerability. Because of this, many valuable documents

with a vast amount of information were passed into the wrong hands.

Before knowledge of Heartbleed became public, defending against the exploit was simply just a matter of chance. If a system was using the affected version of OpenSSL, they were at risk for being attacked by the exploit.

Nevertheless, at a high level, systems could have minimized the damage from Heartbleed attacks by following the principles of defense in depth, segmentation, and least privilege. For example, corporations could make sure that one breach into a router or endpoint on their network does not expose the whole system. Practically, this means having multiple factor authentication on all corporate devices (defense in depth), as well as keeping the smallest amount of information on the most external devices (least privileges). This way, a single attack vector (Heartbleed) will not let the attacker gain access to the most important information in the system.

On the other hand, following some basic security tenets were detrimental to securing systems against the exploit, specifically continuous improvement and open design. Users that are quick to apply security updates (which is generally a good security practice) were punished for their trust in the developers of OpenSSL. While OpenSSL (and other open source projects) are generally secure, the bug that caused Heartbleed was small enough to go unnoticed for years. [5]

References

[1] "RFC 5246." https://tools.ietf.org/html/rfc5246. Posted: August 2008 Accessed: December 4, 2016.

[2] "SSL profiles part 6: SSL renegotiation." https://devcentral.f5.com/articles/ssl-profiles-part-6-ssl-renegotiation. Posted: June 11, 2013 Accessed: December 4, 2016.

[3] "RFC 6520." https://tools.ietf.org/html/rfc6520. Posted: February 2012 Accessed: December 4, 2016.

[4] Codenomicon, "Heartbleed bug." http://heartbleed.com. Posted: April 2014 Accessed: December 4, 2016.

[5] T. Hunt, "Everything you need to need to know about the heartbleed SSL bug." https://www.troyhunt.com/everything-you-need-to-know-about3. Posted: April 8, 2014 Accessed: December 4, 2016.

[6] D. Gilbert, "Heartbleed bug claims first confirmed victims in canada." http://www.ibtimes.co.uk/heartbleed-bug-claims-first-confirmed-victims-canada-1444751/. Posted: April 14, 2014 Accessed: December 4, 2016.

[7] T. Cestnick, "CRA gaffes: Your taxpayer dollars at work." http://www.theglobeandmail.com/globe-investor/personal-finance/taxes/cra-gaffes-your-taxpayer-dollars-at-work/article24170833/. Posted: April 29, 2015 Accessed: December 4, 2016.

[8] J. Sims. http://news.nationalpost.com/news/canada/i-never-meant-harm-says-student-who-hacked-into-canada-revenue-thereby-extending-tax-deadline/. Posted: May 7, 2016 Accessed: December 4, 2016.

[9] D. Goodin, "Meet "cupid," the heartbleed attack that spawns "evil" wi-fi networks." http://arstechnica.com/security/2014/06/meet-cupid-the-heartbleed-attack-spawns-evil-wi-fi-networks/. Posted: June 2, 2014 Accessed: December 4, 2016.

[10] C. Byrd, "Open secure wireless." https://sites.google.com/a/riosec.com/home/articles/Open-Secure-Wireless/Open-Secure-Wireless.pdf?attredirects=1. Posted: May 5, 2010 Accessed: December 4, 2016.

[11] L. Grangeia, "Heartbleed, cupid and wireless." http://www.sysvalue.com/en/heartbleed-cupid-wireless/. Posted: May 30, 2014 Accessed: December 4, 2016.