Heartbleed
Moving Forward

White Paper

Abstract
The 7th April was when most people came to hear about the notorious Heartbleed bug. What are the lessons learnt and what steps Senior Management and IT Security Managers can take to manage the business risks associated with software flaws, attacks and malware.

Chris Russell
Contents

Introduction 3

Do you want The Good News, The Bad News or The Other Bad News? 5

Heartbleed – The Good, The Bad And The Just Not Sure 6

Lessons Learnt 7

Conclusions 8
Introduction

What was or is Heartbleed? heartbleed.com. It is a bug, a default, a mistake. At the end of the day it was human error. For reasons that I will come to later I will take some time explaining some technical details. For the non-technical amongst you please persevere. If all else fails refer to xkcd.com/1354/ for a cartoon version.

The Heartbleed bug affected OpenSSL a library that is used to allow websites to support secure web traffic, ie https.

Part of this functionality is a heartbeat command, something that allows one computer to know what the other computer is still alive and running. In this case the heartbeat is a request for the website to echo back a given “word or phrase”. Basically along the lines of

“To prove you are still alive and working reply with the word ‘Fred’”. It is up to the sending computer to decide what the word or phrase should be. For reasons that I will spare you the actual message is

“To prove you are still alive and working reply with the word ‘Fred’ and I am expecting 4 characters in the answer”.

And so to the actual bug, if you send a message to a website that uses broken version of OpenSSL a message such as

“To prove you are still alive and working reply with the word ‘Fred’ and I am expecting 400 characters in the answer”.

The server replies with Fred followed by 396 random characters from memory that will pad the message out to the required 400 characters. Although unpredictable may be a better word than random. The area of memory that the padding will come from will be memory that was being used by another part or process of the OpenSSL library. When a process has finished with an area of memory that memory is freed up, but the contents are not deleted. So the area of memory adjacent to the area being used by the heartbeat function is likely to contain data relating to recent activities on the website.

This data maybe relating to a web form submission, including possibly a user’s password, it may also be data relating to key management on the server, or temporary data used in data manipulation that is of no interest at all. The key is this data is not encrypted.

For more detail read blog.cloudflare.com/answering-the-critical-question-can-you-get-private-ssl-keys-using-heartbleed

And an attacker can actually ask, “To prove you are still alive and working reply with the word ‘Fred’ and I am expecting 65000 characters in the answer”.

What Heartbleed actually allows an attacker to do is take peaks at what is going on in the memory of a website. In essence casting a net into the website’s memory to see what it can catch.
Do you want The Good News, The Bad News or The Other Bad News?

The good news is that on busy websites the contents of the memory will be impossible to predict and chances are a single Heartbleed request will uncover any useful information is probably very small.

However, the bad news is that this is an attack that is very easy to automate. An attacker can just keep getting random bits of the server memory and then sift through what they find for important information such as secret keys, passwords etc.

The other bad news is that such an attack leaves no trace, so I can leave my attack merrily running looking for nuggets of information amongst the general noise of the server’s memory.

And perhaps most worryingly of all is that it has been shown that under certain circumstances a server's private key can be compromised which opens up the possibility of man-in-the-middle and Phishing attacks.

One common theme here is the unknown. The attacker does not know what they will get back in a Heartbleed Response. The victim does not know whether they have been attacked or not or if they have been attacked what has been leaked.

It is not clear under what circumstances really sensitive information will be leaked. It has been. This makes life uncomfortable.
Heartbleed - The Good, The Bad And The Just Not Sure

The response to the Heartbleed issue shows the best and the worst of the IT security industry. Let’s start with the bad. Starting with the bad - Advice to reset all passwords was not thought through and unfortunately it was this advice that the mainstream media picked up on. Why wouldn’t they, it was a simple piece of headline grabbing response.

As pointed our earlier an attacker never knows what they are going to get in their Heartbleed responses. However in order to increase their chances of passwords being included in their harvest what an attacker really wants is lots of traffic coming to the site relating to passwords, perhaps based on a global instruction to change all passwords.

The other possibility of attack is phishing attacks, either just plain old Phishing or a site that uses a stolen key obtained via Heartbleed. What does this attack need to increase its chances of success? People expecting emails with hyperlinks in them telling them to change their password.

In fact if you were the customer of a site vulnerable to Heartbleed, probably the best thing to do was just to stay away and wait for advice from the owner of the website.

The challenge is how the industry reacts to issues like this and it is a challenge that I do not pretend to have an answer to. When details of the defect were released it started a race between the bad guys wanting to exploit the defect and the good guys trying to prevent damage caused by the defect.

Tools were released to test whether a site was vulnerable or not (http://tif.mcafee.com/heartbleedtest) and instructions on how to detect an attack. (http://blog.logrhythm.com/security/the-internets-bleeding-heart/)

What these and many others responses did show is a genuine ambition amongst the IT security community to fix the issue and share knowledge and insights into the possible impact of the defect and how to mitigate it.

1 But may be detectable http://blog.logrhythm.com/security/the-internets-bleeding-heart/
2 Unfortunately exactly what these circumstances are is not clear, possibly requires a server reboot whilst server is under attack
3 http://staff.tumblr.com/post/82113034874/urgent-security-update
Lessons Learnt

Be prepared

How well did you understand the description of the bug? Did you read the article about how memory is allocated within OpenSSL? How well do you understand the potential impact of your webserver primary key being compromised? Well it does not matter if you do. What matters is that someone in your organisation does understand and understands very well and that this person can inform and influence (or be part of) the leadership of your organisation so that the best decisions possible can be made based on imperfect information.

In the wake of the Heartbleed announcement there was a lot of bad advice, when what as required was a cool head, an assessment of the risk and a considered action plan. Risk is something that organisations deal with all the time, yet many organisations want their IT to be risk free. It never will be. IT has risks that need ongoing management and you need the expertise to be able to manage this risk.

It will happen again

The defect in OpenSSL is a classic defect. Any programmer who has worked in languages such a C that allows chunks of memory to be randomly accessed has no doubt created similar defects in their time. For most of us these bugs were either picked up in testing or no doubt had a much lower impact than this one.

But defects, malware and attacks will continue to compromise servers and once again we will hear the advice change all your passwords or have a different password for every site or both.

If your website data is valuable, do not rely on static passwords to protect it.

---

4 http://www.computing.co.uk/ctg/news/2339208/confusion-over-conflicting-advice-over-heartbleed-openssl-security-flaw#
5 More to this than just stealing the key, stealing the key would be the first stage of a sophisticated attack.
Conclusions

The world is still turning; modern civilisation has not come to a sticky end. Was it a close shave or an issue that was over-hyped? Who knows what would have happened had the good guys not found the defect? Perhaps nobody would have. Maybe the bad guys would have? Maybe the bad guys have known about it for years?

This is the unknown that is so uncomfortable and this is the risk that Security IT professionals need to help their businesses deal with and this is where businesses need to listen to their IT Security Professionals.

My own personal view is I think the good guys found this first and therefore there was a very limited window of opportunity for hackers to exploit Heartbleed.

I stress that I am not basing this on any evidence just a hunch that such a simple but effective attack would have come to light in other ways had it been used widely whereas in fact it was discovered during routine testing.

I am sure that the mumsnet hack so soon after Heartbleed came to light was not a coincidence and was a hacker saying “I’ll try that”.

But if I was in charge of the IT of an enterprise that could have been vulnerable to Heartbleed, I would be having fewer sleepless nights if I had implemented security solutions such as strong authentication on my enterprise’s web presence and that the use of one-time code based authentication (especially two-factor) reduces the value of the data that an attacker can obtain via means such as Heartbleed.

7 http://readwrite.com/2014/04/13/heartbleed-security-codenomicon-discovery#awesm=-oBwCHh7x9WZs5u