Heartbleed -The Art of shutting stable doors

Fittingly enough what was to become the story of potentially the most damaging, expensive and controversial episodes in the history of the Internet begins at the stroke of midnight on New Years Eve 2011.

What motivated Robin Seggelmann, a programmer based in Germany to make the submission at that particular time has been the subject of much speculation, most likely we can assume that in common with most programmers he wasn't blessed with friends or female company and so it came to pass that he wasn't overwhelmed with invites to join in the celebrations, whatever the reasons one of the most infamous updates ever released was launched like a firework into cyberspace and amongst all the hullabaloo that accompanied the coming of a new year Seggelman's contribution probably went unnoticed; the explosion and the resulting fallout wouldn't happen for another two years. Robin Seggelmann submitted the code in an update submitted at 11:59pm on New Year's Eve, 2011. Its purpose was to enable a function called "Heartbeat" in Open SSL, a software package that is used by nearly half of all web servers to enable secure connections.

Inadvertently seggelmann as a result of a programming error in the code he submitted that night, his code had introduced what has turned out to be a catastrophic flaw, which has blown a hole in one of the processes that was designed to make the transit of data across the Internet secure. Christened Heartbleed and given its own natty logo to underline the importance of the find, this flaw was discovered in April 2014 by two separate teams of security researchers from Google and Codenomicon. Finally the pyrotechnics that had been launched at midnight on the last day of 2011 were underway.

OpenSSL

The Open SSL Project

The Open SSL Project that Seggelmann was a volunteer developer for is part of the wider open source movement, which means anyone can take the code behind it and use it to make their own versions of the software. Open SSL has become so successful in that respect that it exists everywhere from servers to network appliances even Android phones and tablets; it has been adopted by Corporations and Governments and is a key component in the security of the web.

Seggelmann to his credit has stated that "I am responsible for the error," "because I wrote the code and missed the necessary validation by an oversight. Unfortunately, this mistake also slipped through the review process and therefore made its way into the released version."

He continued less convincingly that the mistake has nothing to do with its festive date stamp. "The code was the work of several weeks. It's only a coincidence that it was submitted during the holiday season". He makes no reference to girlfriends and his mother was unavailable for comment at the time of going to print.

Of course we might wonder why such an important application has essentially been put into the hands of hobbyists, to get a better picture of how that came to be you have to remember that Open Source is very much at the origins and part of the whole ethos of the Internet. I'm personally not inclined or qualified to argue the pros and cons of this approach and greater men than me have been lynched for trying.

Of course it wasn't long before others got into the business of launching incendiaries and the brick bats are certainly flying. Critics notably from the group who are in the process of creating a rival form of SSL called Libre SSL have claimed that the Open SSL Foundation have been negligent in the way they have carried out their work.

Open SSL have argued that the problems it faces are down to lack of funds. The organisation's president, Steve Marquess, wrote that it has been surviving on less than $1m annual gross revenue since it was formed five years ago, and has just one full-time member.

Governments have stayed remarkably silent on the subject presumably due to a worry that they might be expected to put their collective hand into the treasury pocket and maybe Governments could and should have been contributing all along.

According to Marquess "There should be at least half-dozen full-time Open SSL team members, not just one, able to concentrate on the care and feeding of Open SSL without having to hustle commercial work," Belatedly the funding issues appear to be being taken seriously with Google, Microsoft and Face book each pledging £300,000 over the next 3 years.

Why is Heartbleed such a big issue?

Of course Seggelmann is only partly culpable in the fiasco that has become known as Heat bleed there is supposed to be a pre-release testing program where new code should be reviewed and the companies who have used Open SSL within their own offerings have their own development teams who also seem to have missed the problem.

To try to understand why Heartbleed is such an issue I will need to briefly explain why Open SSL is for some websites and systems a key component. The Secure Socket Layer (SSL) and Transport Layer Security (TLS) is the most widely deployed

security protocols used today. It is a means of providing a secure channel between two machines operating over the Internet or an internal network and thus obscuring the messages that flow across the network from eavesdroppers.

It is desirable for reasons of scale and usability that for widely used processes common solutions are adopted and as Open SSL is freely available it has been adopted and refined by manufacturers and software developers the world over.

So a key system that is there to protect our data could actually be exploited and used to eves drop on us to gain access to login credentials and encryption keys.

We have for a long time taken for granted that when we see the https:\\ at the front of an address that we have moved in to a more secure environment. Post Heartbleed that is not necessarily the case and in some cases should make us more wary.

Of course not all websites that use SSL are at risk we should point out here that only the ones using Open SSL 1.0.1 through 1.0.1f (inclusive) are vulnerable. Whilet as individuals how do we tell if the site we are currently using is affected?. To give you an idea of the scope of the problem it is estimated that it affects two thirds of the world's websites. As you would expect there has been an almighty scramble to identify and secure sites deemed to be at risk and with a reported 60% take-up progress would appear to have been made, the sobering fact is that of the remaining 40% we don't know how many will eventually be secured but we must accept that there will be a proportion that will be left over. Added to this another sobering fact if you weren't sober enough already is that it is alleged that a proportion of the fixes for Heartbleed are being badly or improperly implemented which could result in the problem being exacerbated.

How Heartbleed works

In the natural way of things I would normally steer clear of trying to explain how processes work within the networking environment. In this case we can make an exception as the flaw which actually leaks data in the Heartbleed bug is blindingly simple as long as we don't let the acronyms scare us.

The update that Seggelmann worked on was related to an upgrade called "Heartbeat"; this is a function within the Transport Layer Security" (TLS), a system that is used to protect confidential data when surfing the web. The function of Heartbeat is to keep connections open, even when no data is being passed.

In normal practice a user's computer regularly sends a Heartbeat packet to the server for the duration of the connection, so imagine you are visiting your banking web site. This packet simply contains a chunk of random data, and a statement

relating to the quantity of data sent; the server receives the packet, and then sends back exactly the same data, to confirm that is still listening.

The problem which can be exploited in a Heartbleed attack involves the attacker's computer lying about how much data it has sent: it sends over a single byte of information, but tells the server that it has sent 64KB instead. The server being gullible makes a note, and knows that it has to send 64KB back, but doesn't have a full 64KB of data in the packet it has received, and so instead of doing the sensible thing and rejecting the packet and ending the session it does something that compromises the security of the whole process.

Unfortunately the server's response to this problem of wanting to send back 64KB of data is to fill the rest of the packet with any other information which is resident in its memory at the time.

A computer's memory is where it stores temporary information about the tasks it's working on, and so the data which it pulls into the Heartbleed packet is related to the other queries it's responding to. At Yahoo, that included usernames and passwords of users logging in at the same time; at DuckDuckGo, it was the full text of search queries. The researchers who discovered the bug also say it can include SSL keys, which could let an attacker decrypt conversations captured months before.

DuckDuckGo

mumsnet
By parents for parents

YAHOO!

Who has been exploited so far?

The list of companies to have so far admitted to having been victims of Heartbleed in addition to Yahoo and DuckDuckGo we can add Mumsnet and the Canadian tax authorities but I'm sure the list will grow as the months go by. Ironically Heartbleed may have given the security companies an opportunity to monitor the message boards that are frequented by the perpetrators of Cybercrime.

The fallout from Heartbleed also affects SSL Certification

Another victim would appear to be the industry responsible for the issuing SSL Certification that commonly are used as a means of proving that the webpage we are looking at is actually who they purport to be.

The cost of revoking and re-issuing compromised certificates will amount to millions according to Cloudflare, which provides services to website hosts.

The certificates rely on SSL for the security of private keys used in encryption that must be kept hidden, but the Heartbleed flaw allows an attacker to steal them by pummelling a server with carefully crafted requests.

Cloudfare un-wisely as it turned out initially announced that such an attack was impossible on the type of web server they use, but after opening it up to the public to test they were forced into an abrupt about turn, as it wasn't long before the firm was proved wrong. As a result, it has decided to revoke and reissue all SSL certificates for its customers – well over 100,000 of them.

The Company put out a statement as a response to queries as to why it had not done so earlier, as a preventative measure. "The answer," cofounder Matthew Prince writes, "is that the revocation process for SSL certificates is far from perfect and imposes a significant cost on the internet's infrastructure." He made no mention to the cost it would undoubtedly have on his company's future profits.

What Conclusions can we make

While the Heartbleed fallout undoubtedly affects all of us as individuals, Heartbleed is a problem that must be sorted out by the various industries involved. As individuals all we can take extra care about our Internet Activities.

Of course companies should be looking at their own websites and network infrastructure to see where these issues might apply to them. It is likely that it will take years for the whole mess to be cleared up and it may be that it is the beginning of the end for SSL as we know it.

As an industry we need to take heed, learn our lesson and be vigilant as to where the next Open SSL is likely to arise.

All the material for this article was produced and sourced by Gary Johnston

IT Security Specialist