Google | Online Security Blog

The latest news and insights from Google on security and safety on the Internet

CVE-2015-7547: glibc getaddrinfo stack-based buffer overflow

Posted: Tuesday, February 16, 2016 G+1 144

Posted by Fermin J. Serna, Staff Security Engineer and Kevin Stadmeyer, Technical Program Manager

Have you ever been deep in the mines of debugging and suddenly realized that you were staring at something far more interesting than you were expecting? You are not alone! Recently a Google engineer noticed that their SSH client segfaulted every time they tried to connect to a specific host. That engineer filed a ticket to investigate the behavior and after an intense investigation we discovered the issue lay in glibc and not in SSH as we were expecting.

Thanks to this engineer's keen observation, we were able determine that the issue could result in remote code execution. We immediately began an in-depth analysis of the issue to determine whether it could be exploited, and possible fixes. We saw this as a challenge, and after some intense hacking sessions, we were able to craft a full working exploit!

In the course of our investigation, and to our surprise, we learned that the glibc maintainers had previously been alerted of the issue via their bug tracker in July, 2015. (bug). We couldn't immediately tell whether the bug fix was underway, so we worked hard to make sure we understood the issue and then reached out to the glibc maintainers. To our delight, Florian Weimer and Carlos O'Donell of Red Hat had also been studying the bug's impact, albeit completely independently! Due to the sensitive nature of the issue, the investigation, patch creation, and regression tests performed primarily by Florian and Carlos had continued "off-bug."

Google
google.com/+google
News and updates on Google's products, technology and more
G+ Follow +1
+ 10,865,223

Archives

Feed

Google
YouTube 3M
Follow @google
Follow

Give us feedback in our Product

This was an amazing coincidence, and thanks to their hard work and cooperation, we were able to translate both teams' knowledge into a comprehensive patch and regression test to protect glibc users.

That patch is available here.

Issue Summary:

Our initial investigations showed that the issue affected all the versions of glibc since 2.9. You should definitely update if you are on an older version though. If the vulnerability is detected, machine owners may wish to take steps to mitigate the risk of an attack.

The glibc DNS client side resolver is vulnerable to a stack-based buffer overflow when the getaddrinfo() library function is used. Software using this function may be exploited with attacker-controlled domain names, attacker-controlled DNS servers, or through a man-in-the-middle attack.

Google has found some mitigations that may help prevent exploitation if you are not able to immediately patch your instance of glibc. The vulnerability relies on an oversized (2048+ bytes) UDP or TCP response, which is followed by another response that will overwrite the stack. Our suggested mitigation is to limit the response (i.e., via DNSMasq or similar programs) sizes accepted by the DNS resolver locally as well as to ensure that DNS queries are sent only to DNS servers which limit the response size for UDP responses with the truncation bit set.

Technical information:

glibc reserves 2048 bytes in the stack through alloca() for the DNS answer at _nss_dns_gethostbyname4_r() for hosting responses to a DNS query.

Later on, at send_dg() and send_vc(), if the response is larger than 2048 bytes, a new buffer is allocated from the heap and all the information (buffer pointer, new buffer size and response size) is updated.

Under certain conditions a mismatch between the stack buffer and the new heap allocation will happen. The final effect is that the stack buffer will be used to store the DNS response, even though the response is larger than the stack buffer and a heap buffer was allocated. This behavior leads to the stack buffer overflow.

The vectors to trigger this buffer overflow are very common and can include ssh, sudo, and curl. We are confident that the

Forums.

exploitation vectors are diverse and widespread; we have not attempted to enumerate these vectors further.

Exploitation:

Remote code execution is possible, but not straightforward. It requires bypassing the security mitigations present on the system, such as ASLR. We will not release our exploit code, but a non-weaponized Proof of Concept has been made available simultaneously with this blog post. With this Proof of Concept, you can verify if you are affected by this issue, and verify any mitigations you may wish to enact.

As you can see in the below debugging session we are able to reliably control EIP/RIP.

```
(gdb) x/i $rip
=> 0x7fe156f0ccce <_nss_dns_gethostbyname4_r+398>: req
(gdb) x/a $rsp
0x7fff56fd8a48: 0x4242424242424242 0x4242424242420042
```

When code crashes unexpectedly, it can be a sign of something much more significant than it appears; ignore crashes at your peril!

Failed exploit indicators, due to ASLR, can range from:

- Crash on free(ptr) where ptr is controlled by the attacker.
- Crash on free(ptr) where ptr is semi-controlled by the attacker since ptr has to be a valid readable address.
- Crash reading from memory pointed by a local overwritten variable.
- Crash writing to memory on an attacker-controlled pointer.

We would like to thank Neel Mehta, Thomas Garnier, Gynvael Coldwind, Michael Schaller, Tom Payne, Michael Haro, Damian Menscher, Matt Brown, Yunhong Gu, Florian Weimer, Carlos O'Donell and the rest of the glibc team for their help figuring out all details about this bug, exploitation, and patch development.

No comments:

Post a Comment

Links to this post

Create a Link

Home Older Post

Company-wide

Official Google Blog

Public Policy Blog

Student Blog

Products

Android Blog

Chrome Blog

Lat Long Blog

Developers

Developers Blog

Ads Developer Blog

Android Developers Blog

Google · Privacy · Terms