Glibc vulnerability CVE-2015-7547 and Google!

DIEZ AÑOS FORTALECIENDO LA
CIBERSEGURIDAD NACIONAL

CCN
Centro Criptológico Nacional

ESQUEMA DE EVALUACIÓN Y CERTIFICACIÓN DE LA SEGURIDAD DE LAS TECNOLOGÍAS DE INFORMACIÓN

Bio

Staff Security Engineer @ Google, leading ISE-TPS (production security):

- Vulnerability research (2k+ CVEs in OSS)
- Exploit mitigations (Linux KASLR, LLVM secure allocator, …)
- Sandboxing technologies (seccomp-bpf, nsjail, …)
- API hardening: non security engineers to make **easy & secure** decisions

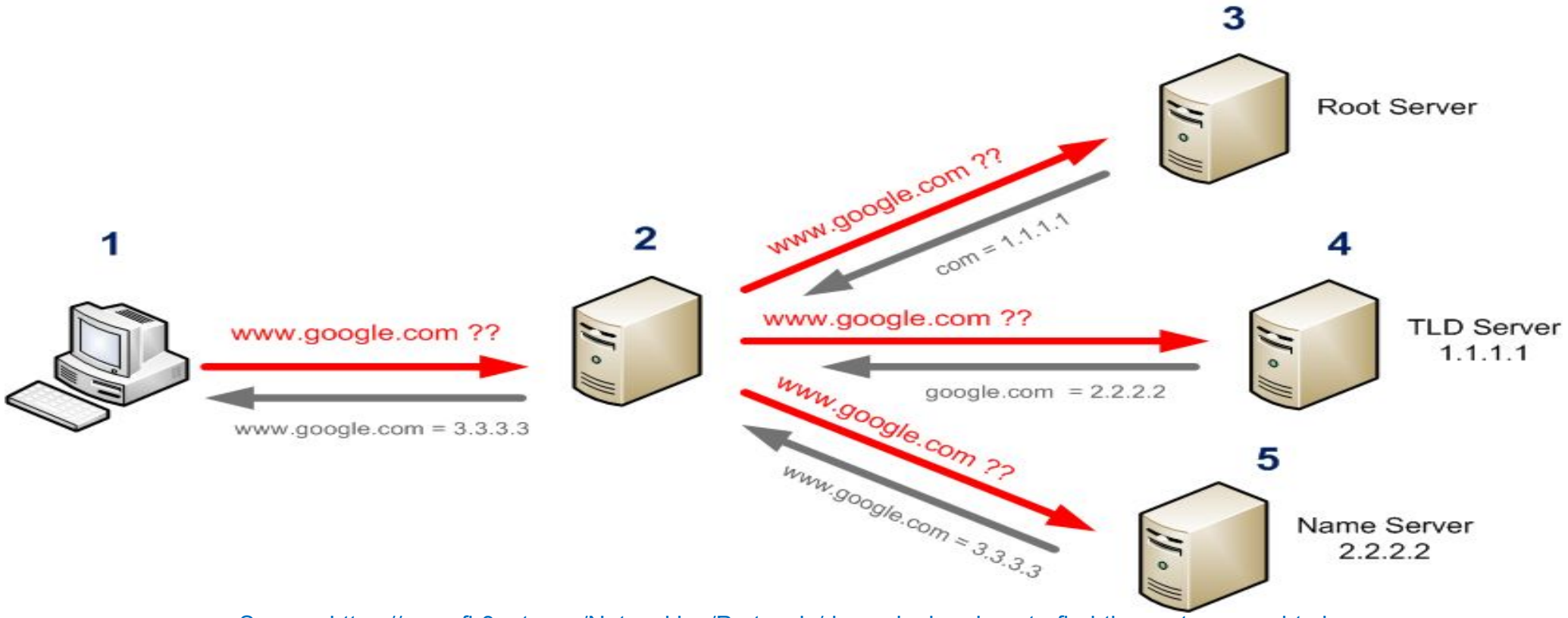
Previously:

- Microsoft (MSRC) and main developer/visionary of EMET
- Owner of NGSEC
- Founder member of S21SEC

Agenda

1. Background information
2. Technical details of the vulnerability
3. Exploitation
4. Vulnerability disclosure
5. Lessons learned

How DNS works



Source: https://www.fir3net.com/Networking/Protocols/dns-nslookup-how-to-find-the-root-servers.html

How DNS works with Linux/Glibc

API:

```
int getaddrinfo(const char *node,
                const char *service,
                const struct addrinfo *hints,
                struct addrinfo **res);
```

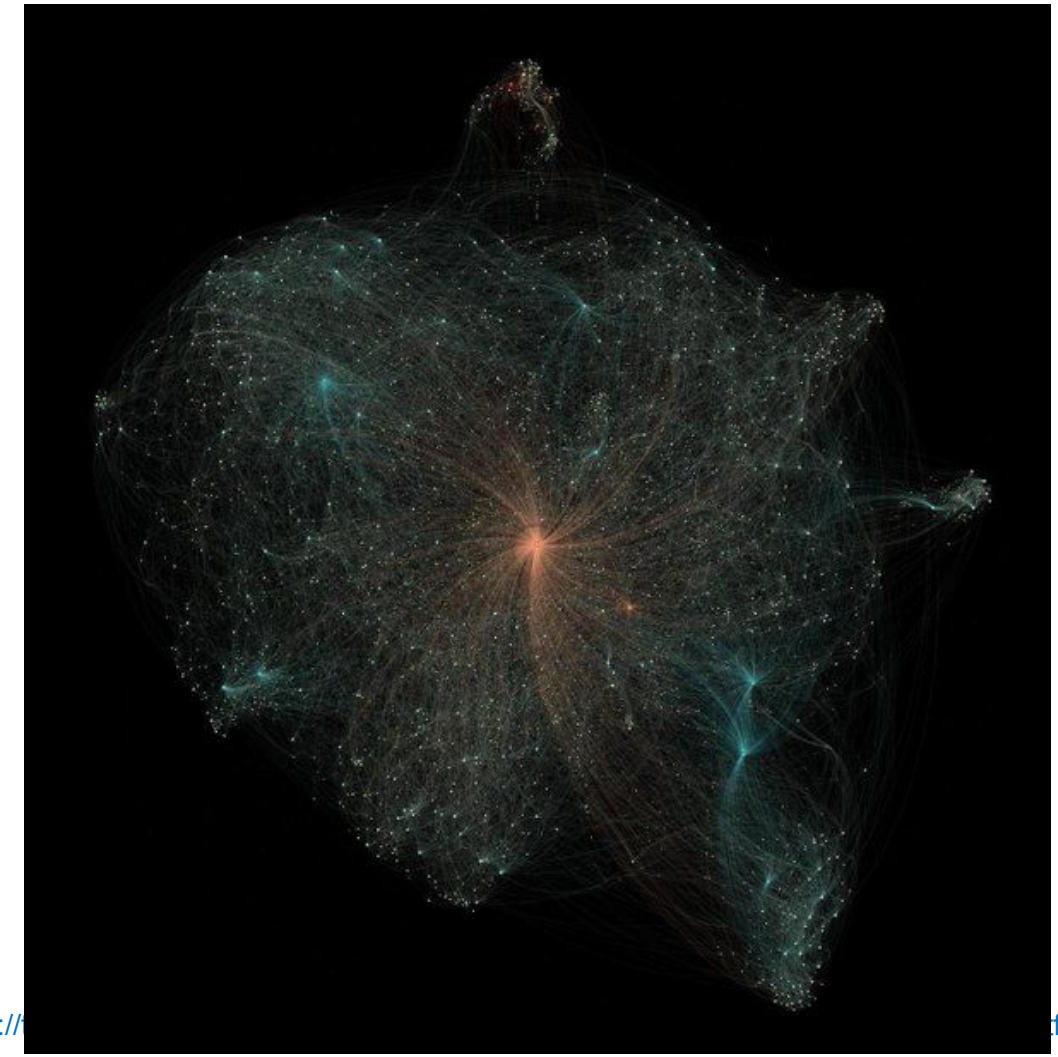
Description:

Given *node* and *service*, which identify an Internet host and a service, getaddrinfo() returns one or more *addrinfo* structures, each of which contains an Internet address that can be specified in a call to bind(2) or connect(2).

Glibc and Linux

Dependency graph of ubuntu packages

Guess who is the bright spot?



Source: Rui Vieira - https://t tfw

Imagine a vulnerability there...

```
[[root@sandbox-3]$ gcc -o client client.c
[[root@sandbox-3]$
[[root@sandbox-3]$ ./client
 Segmentation fault (core dumped)
[[root@sandbox-3]$
[[root@sandbox-3]$ wget https://google.com
 --2016-02-16 15:51:51--  https://google.com/
 Resolving google.com... Segmentation fault (core dumped)
[[root@sandbox-3]$
[[root@sandbox-3]$ curl https://google.com
 Segmentation fault (core dumped)
[[root@sandbox-3]$
 [root@sandbox-3]$
```

Anyone ever found a remote vulnerability in sudo?

```
root@staging:~# apt-get update
0% [Connecting to us.archive.ubuntu.com] [Connecting to archive.canonical.com] [C
0% [Connecting to us.archive.ubuntu.com] [Connecting to archive.canonical.com] [C
** Error in `/usr/lib/apt/methods/http': double free or corruption (out): 0x00007
*** Error in `/usr/lib/apt/methods/http': double free or corruption (out): 0x0000
E: Method http has died unexpectedly!
E: Sub-process http received a segmentation fault.
root@staging:~#
root@staging:~# exit
ubuntu@staging:~$ sudo bash
*** Error in `sudo': double free or corruption (out): 0x00007fff04941df0 ***
======= Backtrace: =========
/lib/x86_64-linux-gnu/libc.so.6(+0x80a46)[0x7f1701baba46]
/lib/x86_64-linux-gnu/libresolv.so.2(__libc_res_nsearch+0x2d2)[0x7f1700869882]
/lib/x86_64-linux-gnu/libnss_dns.so.2(_nss_dns_gethostbyname4_r+0xf8)[0x7f1700a7c
/lib/x86_64-linux-gnu/libc.so.6(+0xcbf98)[0x7f1701bf6f98]
/lib/x86_64-linux-gnu/libc.so.6(getaddrinfo+0xf4)[0x7f1701bfadb4]
/usr/lib/sudo/sudoers.so(+0x13289)[0x7f17010c0289]
/usr/lib/sudo/sudoers.so(+0x13f9d)[0x7f17010c0f9d]
sudo(+0x4594)[0x7f1702742594]
/lib/x86_64-linux-gnu/libc.so.6(__libc_start_main+0xf5)[0x7f1701b4cea5]
sudo(+0x58c5)[0x7f17027438c5]
======= Memory map: ========
7f170064a000-7f170065e000 r-xp 00000000 ca:01 394639                     /lib/x86
7f170065e000-7f170085e000 ---p 00014000 ca:01 394639                     /lib/x86
7f170085e000-7f170085f000 r--p 00014000 ca:01 394639                     /lib/x86
7f170085f000-7f1700860000 rw-p 00015000 ca:01 394639                     /lib/x86
7f1700860000-7f1700876000 r-xp 00000000 ca:01 394687                     /lib/x86
7f1700876000-7f1700a76000 ---p 00016000 ca:01 394687                     /lib/x86
```

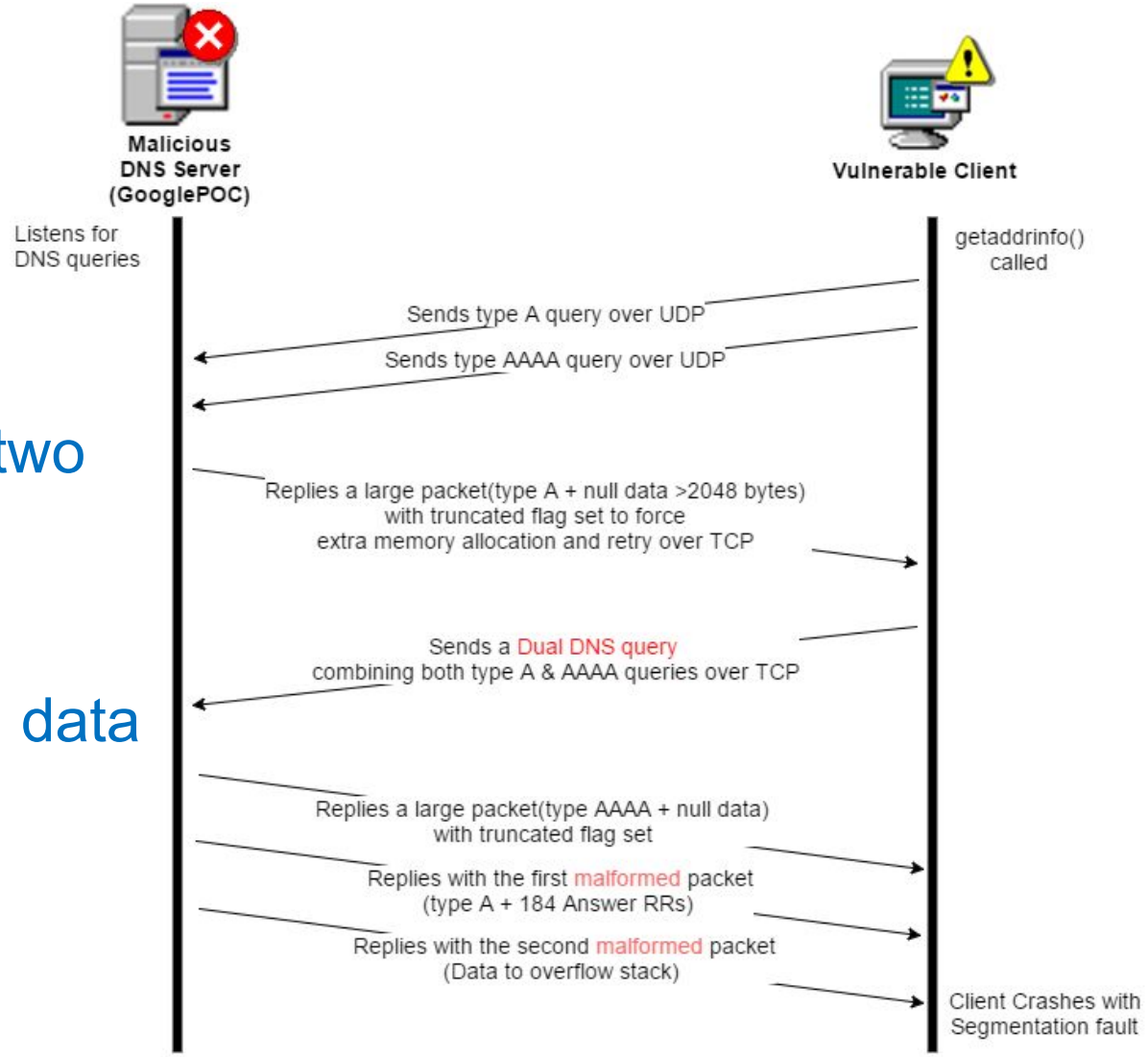
The vulnerability

Logical state bug

After some DNS interaction, usage of two pointers (pointer_1 and pointer_2) and sizes (size_1 and size_2) get mixed.

We end up copying attacker controlled data into pointer_1 (stack) with size_2.

Non trivial to exploit stack buffer overflow.



Source: https://labs.jumpsec.com/

When a remote vulnerability moves into an internet bug

Essentially boils down to answering this:

Does it just affect the Starbucks (local network sniffing/race) scenario?

or

Can I exploit someone over the internet by making them resolve an attacker controlled name?

Hint: second one :)

When a remote vulnerability moves into an internet bug

TODO: details on the internet scenario

Exploitation

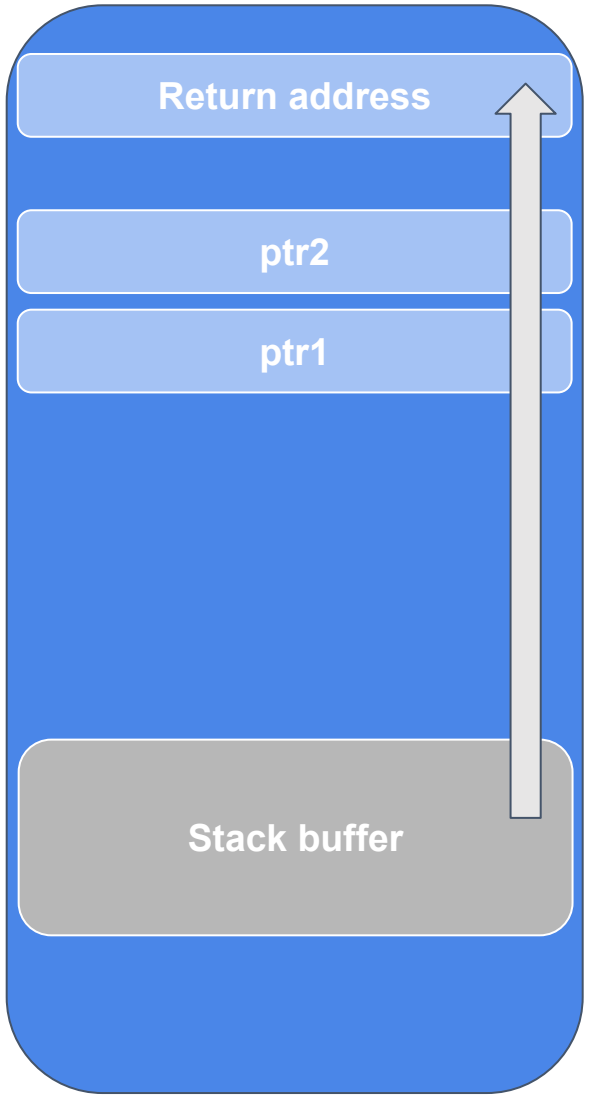
We got it exploited… even on an ASLR'd environment

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

Exploitation

Plain stack overflow, but if you use 2000 techniques you run out into:

- Local variable at the stack (ptr1) is overridden and later used.
  - Easy… override it with a readable address.
- Before the return where we control the saved EIP/RIP there is a free(ptr2)
  - ptr2 comes from the stack, we control it
  - Not easy to bypass… we enter heap exploitation to not crash

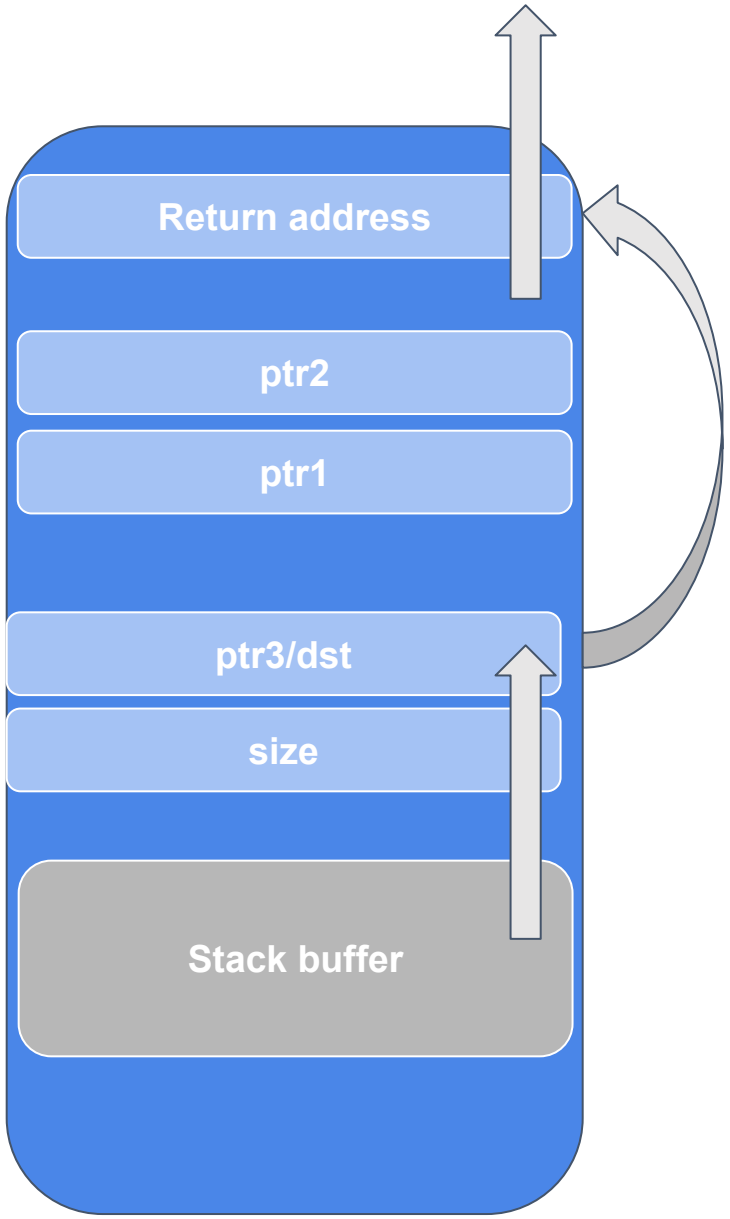


Exploitation

Solution: what else is happening before the free() with local variables that the attack controls?

Bingo: almost-memcpy with dst and size coming from stack

- Bypass ASLR with a **partial overflow** of dst
- dst points to the stack after ptr2
- free still succeeds with not overridden pointer



Vulnerability disclosure

TODO: timeline

Thanks and questions time!

How to contact me:

Twitter: @fjserna

Email: fjserna@google.com or fjserna@gmail.com

Now, don’t be shy.

- **E-Mails**
  - info@ccn-cert.cni.es
  - ccn@cni.es
  - sat-inet@ccn-cert.cni.es
  - sat-sara@ccn-cert.cni.es
  - organismo.certificacion@cni.es
- **Websites**
  - www.ccn.cni.es
  - www.ccn-cert.cni.es
  - www.oc.ccn.cni.es
- **Síguenos en**