Apache use after free bugs

While doing some fuzz testing on the apache httpd server with address sanitizer we regularly observed use after free bugs. We originally observed these issues in the http2 module, but we were also able to reproduce them without http2 enabled, so either we're facing multiple bugs or there's an underlying bug in the core apache code.

Originally we used fuzzing payloads to trigger this bug, but we later observed that sending random garbage in parallel is enough to trigger the bug.

We reported this behavior to the apache security team for the first time in June 2018. The apache developers did not seem to take the issue as seriously as we had expected.

It was pointed out to us that some fixes already in their code may fix the issue, however we are still able to reproduce these bugs in the latest version (2.4.37).

The apache developers indicated to us that they'd not consider these security issues unless we can show a practical exploit. Due to the complexity of the apache code base and our lack of specialization in binary memory exploitation we feel unable to do this. It is, however, our belief that use after free bugs should generally be seen as potential security bugs.

For this reason, we have chosen to share this information with the community and hope others will continue the analysis.

apr pool allocator

For memory allocations apache http uses the apr library's pool allocator that allows reserving a larger chunk of memory as a pool and do memory allocations within that pool. This can, and in our case does, hide memory safety issues.

apr has an option --enable-pool-debug=yes that will cause a single malloc call for each memory allocation, allowing the use of memory safety checkers like ASAN.

The apache developers suggested that our ASAN reports may stem from an incompatibility between the pool debugger and the http2 module. However we were later able to reproduce these issues without the http2 module.

We were also able to reproduce these issues with valgrind and without the pool allocator.

threading related error

In addition to the ASAN use after free reports, httpd logs threading related errors:

AH00052: child pid [pid] exit signal Aborted (6) apache2: tpp.c:84: __pthread_tpp_change_priority: Assertion `new_prio == -1 || (new_prio >= fifo_min_prio && new_prio <= fifo_max_prio)' failed.

We found a ten year old bug in the Apache bug tracker mentioning such errors: https://bz.apache.org/bugzilla/show_bug.cgi?id=46185

It was closed as "INVALID".

asan stack traces

We share asan stack traces from these bugs at https://github.com/hannob/apache-uaf/tree/master/asan

reproduction

To reproduce the issue:

1. Compile apr with the pool debugger and address sanitizer.
2. Compile apache with address sanitizer.
3. Run a command like this to send random garbage to the server: for x in $(seq 150); do for i in $(seq 1 1000); do head -n 10 /dev/urandom | nc 127.0.0.1 80 & done; sleep 5; done

The bugs appear very irregularly, you may need to "attack" it for a while.

Hanno Böck Craig Young (Tripwire VERT)

Thanks to Markus Vervier and Luis Merino of X41 D-SEC GmbH for double checking.