computer emergency response team
CERT-EU
for the EU institutions, bodies and agencies

CERT-EU Security Advisory 2017-013

Privileges Escalation Vulnerabilities in Unix Operating Systems

June 20, 2017 — v1.0

History:

- *20/06/2017 — v1.0: Initial publication*

Summary

On 19th of June 2017, Qualys Research Team published a blog post [1] and a security advisory [2] about vulnerabilities in the memory management of several UNIX operating systems. These vulnerabilities can lead to privilege escalation on these systems, by corrupting memory and executing arbitrary code. They named the bug **Stack Clash** as it exploits flaws in the way these operating system are handling the stack in memory.

Qualys Research Team validated the exploitation of the **Stack Clash** bug on the following systems: Linux, OpenBSD, NetBSD, FreeBSD, and Solaris, on i386 and amd64 architectures. They worked with vendors to have patches available.

Technical Details

The **Stack Clash** bug and other findings discovered by Qualys Research Team received the following CVEs:

- CVE-2017-1000364 - issue in the size of the stack guard page on Linux [3]
- CVE-2017-1000365 - bypass of size restriction limit for arguments and environmental strings in Linux [4]
- CVE-2017-1000366 - arbitrary code execution in `glibc` [5]
- CVE-2017-1000367 - *sudoer-to-root* exploit [6]
- CVE-2017-1000369 - arbitrary code execution in Exim [7]
- CVE-2017-1000370 - local-root exploit against `ld.so` and most SUID-root binaries on i386 Debian, Fedora, and CentOS [8]
- CVE-2017-1000371 - local-root exploit against `ld.so` and most SUID-root PIEs on i386 Debian, Fedora, and CentOS [9]
- CVE-2017-1000372 - arbitrary code execution in OpenBSD implementation of the stack guard page [10]
- CVE-2017-1000373 - stack memory manipulation in OpenBSD `qsort` function [11]
- CVE-2017-1000374 - arbitrary code execution in NetBSD implementation of the stack guard page [12]
- CVE-2017-1000375 - memory manipulation and arbitrary code execution in NetBSD [13]

- CVE-2017-1000376 - arbitrary code execution in `libffi 3.2.1` [14]
- CVE-2017-1000377 - bypass of stack guard page in PAX Linux kernel [15]
- CVE-2017-1000378 - memory manipulation and arbitrary code execution in NetBSD via `qsort` function [16]
- CVE-2017-1000379 - stack manipulation in AMD64 Linux kernel [17]

Qualys Research Team decided to answer a question from 2005 regarding the way operating systems manage large memory: *If the heap grows up, and the stack grows down, what happens when they clash? Is it exploitable? How?*. During their research, they exploited such *stack-clashes*, even with protection against such attacks (a *guard-page* mapped below the stack) implemented in Linux (2010).

The particularity of the stack is that it will automatically grow when more memory is needed. When it grows to much, part of it can get overwritten by another memory region, leading potentially to code execution and privilege escalation.

Products Affected

Impacted: Most distributions of Linux, OpenBSD, NetBSD, FreeBSD, and Solaris, on i386 oand amd64 architectures.

These vulnerabilities require at least a local account on the targeted machines and are not remotely exploitable.

Recommendations

Apply patches provided by the vendors on all affected systems.

As a workaround, local access to affected systems can be restrict, and a hard `RLIMIT_STACK` and `RLIMIT_AS` limits can be set for local users and remote services to some reasonably low values [18].

References

[1] Qualys Research Team blog post https://blog.qualys.com/securitylabs/2017/06/19/the-stack-clash

[2] Qualys Security Advisory https://www.qualys.com/2017/06/19/stack-clash/stack-clash.txt

[3] CVE-2017-1000364 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000364

[4] CVE-2017-1000365 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000365

[5] CVE-2017-1000366 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000366

[6] CVE-2017-1000367 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000367

[7] CVE-2017-1000369 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000369

[8] CVE-2017-1000370 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000370

[9] CVE-2017-1000371 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000371

[10] CVE-2017-1000372 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000372

[11] CVE-2017-1000373 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000373

[12] CVE-2017-1000374 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000374

[13] CVE-2017-1000375 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000375

[14] CVE-2017-1000376 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000376

[15] CVE-2017-1000377 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000377

[16] CVE-2017-1000378 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000378

[17] CVE-2017-1000379 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000379

[18] Setrlimit manual https://linux.die.net/man/2/setrlimit