NEWS

X41 D-SEC GmbH Security Advisory: X41-2020-006

Advisory X41-2020-006: Memory Corruption Vulnerability in bspatch

Severity Rating:
High

Confirmed Affected Versions:
Colin Percival's bsdiff 4.3

Confirmed Patched Versions:
FreeBSD's bsdiff (https://svnweb.freebsd.org/base/head/usr.bin/bsdiff/bspatch/bspatch.c)

Vendor:
Colin Percival

Vendor URL:
https://www.daemonology.net/bsdiff/

Vendor Reference:
None

Vector:
Patch file

Credit:
X41 D-SEC GmbH, Luis Merino

Status:
Public

CVE:
CVE-2020-14315

CWE:
119

CVSS Score:
N/A

CVSS Vector:
N/A

Advisory-URL:
https://www.x41-dsec.de/lab/advisories/x41-2020-006-bspatch/

Summary and Impact

A memory corruption vulnerability is present in bspatch as shipped in Colin Percival's bsdiff tools version 4.3. Insufficient checks when handling external inputs allows an attacker to bypass the sanity checks in place and write out of a dynamically allocated buffer boundaries.

Even though the patching procedure is usually combined with integrity and authenticity checks, an attacker that is able to deliver a malicious patch can cause heap corruption in the process running bspatch code, when the authenticity checks happen after applying the patches. Depending on their ability to control and shape the heap state before and during the processing of a malicious patch file, remote code execution may be achieved. This has already been demonstrated (https://gist.github.com/anonymous/e48209b03f1dd9625a992717e7b89c4f#file-freebsd-txt-L1192) as a proof-of-concept exploit in 2016 by an anonymous author against the FreeBSD bspatch implementation on 32bit architectures.

This issue was initially reported for bspatch in bsdiff "as used in Apple OS X before 10.11.6 and other products" with CVE-2014-9862 by an anonymous researcher and was partially addressed by several projects, including Android (https://android.googlesource.com/platform/external/bsdiff/+/4d054795b673855e3a7556c6f2f7ab99ca509998%5E%21/#F0), ChromiumOS (https://bugs.chromium.org/p/chromium/issues/detail?id=372525) and FreeBSD (https://www.freebsd.org/security/advisories/FreeBSD-SA-16:25.bspatch.asc) during 2016. This initial batch of fixes prevented the attack via negative control values.

Nevertheless, huge control values that would integer overflow the sanity checks and allow an attacker writing out of bounds were not fixed. A subsequent patch was released by FreeBSD (https://www.freebsd.org/security/advisories/FreeBSD-SA-16:29.bspatch.asc) addressing the remaining issues together with additional hardening. Unfortunately, most of bspatch copies didn't port this fix.

It is worth mentioning that bsdiff 4.3, as hosted at Colin Percival's bsdiff website https://www.daemonology.net/bsdiff/, still ships a copy of bspatch.c vulnerable to these issues via both negative and huge control values. All the Linux distributions we have checked shipping bsdiff are building from this sources, with some of them applying the partial fix initially released.

Product Description

bsdiff and bspatch are tools for building and applying patches to binary files. They provide an efficient way to apply binary patches for applications update mechanisms.

Analysis

Insufficient checks when calculating the buffer offset and size of write operations allows writing out of a heap allocated buffer boundaries.

```
    while(newpos<newsize) {
        /* Read control data */
        for(i=0;i<=2;i++) {
            lenread = BZ2_bzRead(&cbz2err, cpfbz2, buf, 8);
            if ((lenread < 8) || ((cbz2err != BZ_OK) &&
                 (cbz2err != BZ_STREAM_END)))
                errx(1, "Corrupt patch\n");
            ctrl[i]=offtin(buf);
        };

        /* Sanity-check */
        if(newpos+ctrl[0]>newsize)
            errx(1,"Corrupt patch\n");

        /* Read diff string */
        lenread = BZ2_bzRead(&dbz2err, dpfbz2, new + newpos, ctrl[0]);
```

When `ctrl[0]` takes either negative values or big enough values to overflow `newpos+ctrl[0]`, the sanity check in place will pass allowing operations that write out of buffer `new` boundaries via `BZ2_bzRead()`. It is worth mentioning that `BZ2_bzRead()` will truncate `ctrl[0]` from 64-bit `off_t` to 32-bit `int`.

It is expected that an attacker that is able to deliver an specially crafted patch file can gain remote code execution capabilities when certain conditions for exploitation are met.

Proof of Concept

A crashing reproducer can be downloaded from https://github.com/x41sec/advisories/blob/master/X41-2020-006/x41-2020-006-bspatch-poc.patch

Fix

Please, refer to the FreeBSD advisories
https://www.freebsd.org/security/advisories/FreeBSD-SA-16:25.bspatch.asc and
https://www.freebsd.org/security/advisories/FreeBSD-SA-16:29.bspatch.asc for fixes.

Workarounds

As a workaround, only patches passing integrity and authenticity checks should be applied.

Timeline

2016-07-21
CVE-2014-9862 published

2016-07-25
Partial fix for FreeBSD published at FreeBSD-SA-16:25.bspatch

2016-10-10
Complete fix for FreeBSD published at FreeBSD-SA-16:29.bspatch

2020-07-02
X41 Discovers the vulnerability was not or incorrectly fixed upstream and in prominent forks of the code

2020-07-06
Colin Percival and distros@ notified

2020-07-09
Public disclosure

About X41 D-SEC GmbH

X41 is an expert provider for application security services. Having extensive industry experience and expertise in the area of information security, a strong core security team of world class security experts enables X41 to perform premium security services.

Fields of expertise in the area of application security are security centered code reviews, binary reverse engineering and vulnerability discovery. Custom research and IT security consulting and support services are core competencies of X41.

Author: Luis Merino, Markus Vervier
Date: July 09, 2020

«
Advisory X41-2020-002: Multiple Vulnerabilities in Psyprax 3.1.2.2
»
bspatch strikes back

CONTACT

X41 D-SEC GmbH
Krefelder Str. 123
52070 Aachen

+49 (0) 241 9809418-0
+49 (0) 241 9809418-9
info@x41-dsec.de

CONNECT

–

FAQ

Partner

Terms of Use

Privacy

Imprint