Arch Linux Security Projects

Remi Gacogne
November 10, 2015
About me

- Remi Gacogne
- Software Engineer @ PowerDNS
- Teaching C / HA @ Epita, Web Security @ Ionis-STM
- Used to be a SysAdmin, I know uptime matters
- Linux user since 2001 (Arch, CentOS, Debian, Fedora, Mandrake, Slackware)
- rgacogne on IRC (OFTC, Freenode)
Do you know what the Arch Linux Security Team does?
Plan

Tracking Vulnerabilities
One year of ASAs
The easy way
The hard way
Fixing issues

Reproducible builds
What?
Why?
How?

Hardening Arch
Hardening binaries
Protecting pacman’s database
Tracking Vulnerabilities
Roughly one year ago:

- Levente and I: “hey, it’s great to have CVE Monitoring, and we would like to build on that to have security advisories, how can we help?”

- Allan: “it’s not going to happen”

- Allan: “if you want to have security advisories in Arch, do it yourself, because no one else is going to, as there is no glory in it”

Aaaaand there goes my free time..
Well

VULNERABILITIES

VULNERABILITIES EVERYWHERE
Arch Linux Security Advisory ASA-201510-9
========================================
Severity: Critical
Date : 2015-10-15
CVE-ID : CVE-2015-5291
Package : mbedtls
Type : arbitrary code execution
Remote : Yes
Link : https://wiki.archlinux.org/index.php/CVE

Summary
=======
The package mbedtls before version 2.1.2-1 is vulnerable to remote code execution.

Resolution
==========
Upgrade to 2.1.2-1.

# pacman -Syu `mbedtls>=2.1.2-1`

The problem has been fixed upstream in versions 2.1.2, 1.3.14 and 1.2.17.

Workaround
==========
To be protected against this vulnerability, you need to...

Description
============
When the client creates its ClientHello message, due to insufficient bounds checking it can overflow the heap-based buffer containing the message while writing some extensions...
One year later..

- What started as an unofficial project got endorsed by Arch
- No rage-quit (yet)
- Advisories are listed on LWN.net

The team:

- Christian Rebischke (shibumi)
- Levente Polyak (antraxx), also a TU
- Remi Gacogne (rgacogne)
- a lot of people in the shadow (thanks!)
Remote vulnerabilities
Browsers and Flashplugin


We are lazy, so let’s try using automated tools:

- Matching packages against vulnerability databases
- Mitre, OSVDB, Red Hat, NVD..
- https://github.com/jelly/ArchCVE

Unfortunately..
Well, looks like we still need some manual monitoring:

- Reading changelogs
- Following public ML (bugtraq, full-disclosure, oss-sec)
- Following private ML (distros, linux-distros)
A new vulnerability has been found in a package we ship, what now?

- Update the CVE page
- Fix the issue in Arch
### TRACKED CVE's

<table>
<thead>
<tr>
<th>CVE-ID</th>
<th>Package</th>
<th>Disclosure date</th>
<th>Affected versions</th>
<th>Fixed in Arch Linux package version</th>
<th>Arch Linux response time</th>
<th>Status (and related bug reports)</th>
<th>ASA-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2015-7184</td>
<td>firefox</td>
<td>2015-10-15</td>
<td>&lt;= 41.0.1-1</td>
<td>41.0.2-1</td>
<td>&lt;1d</td>
<td>Fixed</td>
<td>ASA-201510-10F</td>
</tr>
<tr>
<td>CVE-2015-5260</td>
<td>spice</td>
<td>2015-09-08</td>
<td>&lt;= 0.12.5-1</td>
<td></td>
<td></td>
<td>Vulnerable</td>
<td>(FSa46738)</td>
</tr>
<tr>
<td>CVE-2015-6755</td>
<td>chromium</td>
<td>2015-10-13</td>
<td>&lt;= 45.0.2454.101-2</td>
<td>46.0.2490.71-1</td>
<td>&lt;1d</td>
<td>Fixed</td>
<td>ASA-201510-8F</td>
</tr>
</tbody>
</table>
Okay, how do we fix the security issue?

- Often, it has already been fixed, because Arch updates really fast.

Otherwise:

- Does a fix exist?
- Has a new version been released with that fix?
Fixing issues

If a new version is available:

- Flag the package as out-of-date, mentioning this is a security update
- After some time, open a bug and add the bug number to the CVE page
- Bully the packager via mail / IRC (hint: don’t do it)
- For community packages, Levente might fix the issue himself
If a fix is available, but not included in any release yet:

- Don’t flag the package as out-of-date
- Open a bug, with the security issue and a link to the fix, and add the bug number to the CVE page
- Bully the packager via mail / IRC (hint: still a big no-no)
Fixing issues

When there is no fix available:

- Don’t flag the package as out-of-date
- Don’t open a bug
- Search the relevant ML, take a look at what well-funded distros are doing
- Propose a patch upstream yourself if you know what you are doing
The issue has been fixed, the package is out of testing:

- Someone in the Security team takes ownership by scheduling an ASA
- Researchs the technicals details
- Writes and issues the advisory
Reproducible Builds
What are reproducible builds?

“Reproducible builds are a set of software development practices which create a verifiable path from human readable source code to the binary code used by computers.”\(^1\)

\(^1\)http://reproducible-builds.org
Why do we want reproducible builds?

Arch uses binary packages:

- We **don’t** have to trust the mirrors, thanks to package signing
- We **don’t** have to trust the network either, thanks to package signing again
- However, we need to trust the Trusted Users and Developers
- More importantly, we need to trust the hosts they build their packages on (pkgbuild.com, anyone?)

With reproducible builds, we can check that the binary packages matches the intended source code.
How do we get reproducible builds?

Reproduce the build on another host, and check that there is no difference.

- Get the **PKGBUILD** via **abs** or the **git** repository
- Build using **makechrootpkg**
- Check the cryptographic fingerprints of the files in the resulting package against those of the original one

At large scale:

- Automated using **Jenkins**\(^2\)
- Check the differences with **diffoscope**\(^3\)
- A lot of help from Lunar and h01ger of the Debian reproducible build team (thanks!)
- Using Debian infrastructure at https://reproducible.debian.net/archlinux/archlinux.html

\(^2\)http://jenkins-ci.org/
\(^3\)http://diffoscope.org/
Is it really that easy?

ONE DOES NOT SIMPLY

REPRODUCE BUILDS
Is it really that easy?

That's the theory, but you know the difference between theory and practice, right?

- Timestamps
- Paths
- Locale / Timezone
- CPU type
- UID / GID
- Randomness
- Build chain
A lot of fixes in our toolchain:

- Timestamps in static archives (#45935, --enable-deterministic-archive in binutils)
- Timestamps in packages
- Build chain versions and build options are added to the packages in .BUILDINFO\(^4\)
  
- ...

\(^4\)https://lists.archlinux.org/pipermail/pacman-dev/2015-October/020357.html
Ideally, we would like to see SOURCE_DATE_EPOCH specification\textsuperscript{5} being adopted:

- A UNIX timestamp.
- The value SHOULD be set to the time of the last modification time of the source, incorporating any packaging-specific modifications. For example, in Debian, the timestamp of the latest entry in debian/changelog.
- Upstream build processes MUST use this variable for embedded timestamps in place of the “current” date and time.

\textsuperscript{5}\url{https://reproducible-builds.org/specs/source-date-epoch/}
Fixing the reproducible build issues

- If you are developing a software, please do not include the build time, the builder uid/gid..
- Or at least include an option to get rid of that, like `--enable-reproducible`
- Good news is, we are not alone working on that, and a lot of fixes are pushed upstream
- If you are a Trusted User or a Developer, please build in a clean chroot with `makechrootpkg`
Fixing the reproducible build issues

IF YOU COULD BUILD IN A CLEAN CHROOT

THAT WOULD BE GREAT
Final words about reproducible builds

- As always with security, this is a process, not a product
- Reproducible builds are too valuable to neglect
- Respect the KISS philosophy
Hardening Arch
Why harden?

- Not-so-breaking news: there are vulnerabilities in Arch
- We are good at upgrading, so known vulnerabilities are patched fast
- Still, we depend heavily on upstream
- What about unknown vulnerabilities?
- Raising the exploitability bar
What kind of hardening?

- Hardening packages at build time
- Protecting pacman’s database

I will not talk about:

- Kernel Hardening with grsecurity: linux-grsec and paxd, maintained by Daniel Micay
- Configuration hardening: use the wiki

---

Hardening Arch: binaries
Arch does enable some interesting features by default:

- `CPPFLAGS=-D__FORTIFY_SOURCE=2`, buffer overflow prevention
- `CFLAGS=[..., ] -fstack-protector-strong` stack overflow prevention
- `LDFLAGS=-Wl,-O1,–sort-common,–as-needed,-z,relro` read-only relocation (partial)
Prevent some parts of ELF binaries (non-PLT GOT, dtors, ctors) from being writable

With “full” RELRO, even the PLT GOT is computed at load time and is not writable afterwards. The cost is minimal for daemons
Position-Independent Executable

- Thanks to No-eXecute (NX)/ PaX, you can’t just put your shellcode in memory and execute it, you have to use Return-Oriented Programming (ROP), exploiting already existing gadgets
- With recent kernels, PIC code from libraries is loaded at a random location, thanks to Address space layout randomization (ASLR)
- This makes it harder to find gadgets in it, but the code of the executable itself is predictable without Position-Independent Executables (PIE)
- PIE cost is now very low on x86_64, since gcc’s 5.1 new register allocation algorithm
- We need a gcc switch to make PIE the default: –enable-default-pie (in gcc 6.0, not backported to 5.x yet)
Hardening selected packages:

- Time-consuming, fail-open
- PIE, full-RELRO, non-executable stack for selected packages, ie network daemons, browsers (Firefox, Chromium)
- Need upstream support to do it right (recently pushed upstream to NSD, Unbound)
Using checksec to verify the results

* System-wide ASLR: **Pax ASLR enabled**

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>PID</th>
<th>RELO</th>
<th>STACK CANARY</th>
<th>NX/Pax</th>
<th>PIE</th>
</tr>
</thead>
<tbody>
<tr>
<td>systemd</td>
<td>1</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>pdns_server</td>
<td>1558</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>pdns_server</td>
<td>1561</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>postgres</td>
<td>1565</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>postgres</td>
<td>1571</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>postgres</td>
<td>1574</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>postgres</td>
<td>1577</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>master</td>
<td>16055</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>qmgr</td>
<td>16067</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>systemd-journal</td>
<td>159</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>systemd</td>
<td>1724</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>(sd-pam)</td>
<td>1725</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>screen</td>
<td>1735</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>bash</td>
<td>1796</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>lrvisi</td>
<td>1799</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>postgres</td>
<td>17901</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>postgres</td>
<td>1815</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>paxd</td>
<td>195</td>
<td>Partial RELRO</td>
<td>No canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>tlsrnr</td>
<td>20457</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>systemd-kubelet</td>
<td>259</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>sshd</td>
<td>22107</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>sshd</td>
<td>22109</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>bash</td>
<td>22110</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>screen</td>
<td>22113</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>pickup</td>
<td>22133</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>sshd</td>
<td>22156</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>systemd</td>
<td>22168</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>(sd-pam)</td>
<td>22172</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>bash</td>
<td>22173</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>sudo</td>
<td>22177</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>deluged</td>
<td>26222</td>
<td>Partial RELRO</td>
<td>No canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>systemd-logind</td>
<td>279</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>haveged</td>
<td>291</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>cron</td>
<td>282</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>dbus-daemon</td>
<td>285</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>agetty</td>
<td>290</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>sshd</td>
<td>369</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>ntpd</td>
<td>403</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>fail2ban-server</td>
<td>419</td>
<td>Partial RELRO</td>
<td>No canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
<tr>
<td>postgres</td>
<td>452</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>postgres</td>
<td>453</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>postgres</td>
<td>454</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>postgres</td>
<td>455</td>
<td>Partial RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>No PIE</td>
</tr>
<tr>
<td>unbound</td>
<td>7611</td>
<td>Full RELRO</td>
<td>Canary found</td>
<td>Pax enabled</td>
<td>PIE enabled</td>
</tr>
</tbody>
</table>
Hardening Arch: signing pacman’s database
Right now:

- Packages are signed using the packager’s PGP key
- Databases are not signed
What is in the database?

- Actually one database per repository: core, extra, community, ...
- Tarball of files, one file per package
- Package file contains meta-data: name, version, description, size, dependencies, PGP signature..
What is in the database?

getdns-0.3.3-1-x86_64.pkg.tar.xz

getdns

0.3.3-1

A modern asynchronous DNS API

http://getdnsapi.net/

x86_64

1443175743

Remi Gacogne <rgacogne-arch at coredump dot fr>

libev
libevent
libidn
libuv
unbound
While packages are signed, database is not, so a:

- Rogue mirror
- Man-on-the-middle
- Man-on-the-side

can:

- Hide packages
- Prevent upgrade

by altering the database.
Proposed design

Separate PGP keyring for signing the database:

- Database key is not allowed to sign packages
- Packagers are not allowed to sign the database
- Master database keys stay offline
- Database signing key is online, not readable by packagers, used by repo-add to sign the database
- Can be revoked if needed
- At worst, In case of compromission, we are back to where we are today
Conclusion
Help is always welcome!

There is always some interesting projects to work on, for every skill level, and nobody expects you to commit a lot of time.

- [https://wiki.archlinux.org/index.php/Arch_CVE_Monitoring_Team](https://wiki.archlinux.org/index.php/Arch_CVE_Monitoring_Team)
- [#archlinux-security on Freenode](https://freenode.net)
- arch-security@archlinux.org
- If you are willing to help but don’t know where to begin, please mail me: rgacogne@archlinux.org
Thank you! / Questions?