Trends in Open Source Security

FOSDEM 2013

Florian Weimer ‹fweimer@redhat.com›
Red Hat Product Security Team
2013-02-02

Overview

- Vulnerability tracking
- Tool-chain hardening
- Distribution-wide defect analysis

CVE-based vulnerability tracking

- http://cve.mitre.org/
- CVE-2013-0156
- CVE assignment alerts distributions
- Works well for public issues
  - oss-security mailing list and Kurt Seifried
- Many vendors also assign CVE identifiers

Version-based vulnerability tracking

- For each branch, note the minimum fixed version
- Very complicated, with subtle corner cases
- Tied to a version numbering scheme and branching model

Version-based tracking for CVE-2013-0156

```
CVE-2013-0156
(active_support/core_ext/hash/conversions.rb in Ruby on
Rails before ...)
   - rails 2.3.14.1 (bug #697722; high)
   [squeeze] - rails 2.3.5-1.2+squeeze4.1
   - ruby-activesupport-2.3 2.3.14-5 (bug #697789)
   - ruby-activesupport-3.2 3.2.6-5 (bug #697790)
   - ruby-extlib 0.9.15-3 (bug #697895)
   - libextlib-ruby <removed> (bug #697895)
```

Example packages for CVE-2013-0156

- `- rails 2.3.14.1 (bug #697722; high)`

  `[squeeze] - rails 2.3.5-1.2+squeeze4.1`
- Fixed package versions
  - 2:2.3.14.2 (testing/wheezy)
  - 2.3.5-1.2+squeeze6 (stable/squeeze)
- Unfixed package versions:
  - 2.3.11-0.1 (testing/wheezy)
  - 2.3.5-1.2+squeeze1 (stable/squeeze)
- Can be used to rate the packages on a system

Vulnerability tracking with tracker bugs

- bugzilla.redhat.com entry with the CVE as an alias
  - Will be made public after disclosure
  - Extensive metadata in the “Whiteboard” field
- This tracker bug depends on product-specific bugs
- Lots of automation, relying on Bugzilla features
  - Uploads to Fedora post information in the Bzs
- Rather different from version-based tracking

Tracker bugs example: CVE-2013-0156

- https://bugzilla.redhat.com/show_bug.cgi?id=892870 has these dependencies:
  - Fedora bug: 893281
  - Fedora EPEL bug: 847202
  - Red Hat OpenShift Enterprise internal bugs
    - Tied to RHSA-2013:0153-1
  - Red Hat Subscription Asset Manager internal bugs
    - Tied to RHSA-2013:0154-1
  - Red Hat CloudForms bugs
    - Tied to RHSA-2013:0155-1

Vulnerability tracking requirements

- Ubuntu, Gentoo, OpenSuSE etc. use similar schemes
- Most upstreams provide critical information
  - Analysis in their security advisory or bug tracker
  - Links to individual patches/commits
- Otherwise, it has to be reverse engineered
  - Time-consuming, better spend on patch review/testing
- Distributions publish isolated security patches
  - Related discussions on the public oss-security list

Cross-distro information sharing opportunities

- Package names and versioning schemes differ
- Encoding of upstream versions differs
- CVE ↔ packages mapping could be shared
- Application for Common Platform Enumeration (CPE)?

Public version control repositories

Please publish your security patches in a publicly accessible version control repository as separate commits!

There is really no point in hiding this information.

(Not a trend yet—let's hope it does not turn into one.)

Toolchain hardening

Toolchain hardening

- Probabilistic countermeasures against code execution
- Make the program crash, not run code
- These bugs still need fixing!

Toolchain hardening

- Address space layout randomization
- Non-executable stack, heap
- `malloc`/`free` hardening against direct exploitation of double-`free` bugs
- `-fstack-protector` (stack canaries, if enabled)
- Compiler warnings (errors for format strings)
- `operator new[]` hardening
  - New feature in GCC 4.8
  - Backported to Fedora 18

Toolchain hardening: FORTIFY_SOURCE

- GCC provides access to array sizes using `__builtin_object_size`
  - In cases where this is possible
  - GNU libc passes length to wrapper functions
- GNU libc disables %n in writable format strings

Unused hardening opportunities

- ~~32 bit~~
- Do not use `prelink`
- Randomization of program start address (PIE)
- `BIND_NOW` global offset table (GOT) protection
- `-fwrapv` (deterministic integer overflow)
- `-fstack-check`

Stack checking

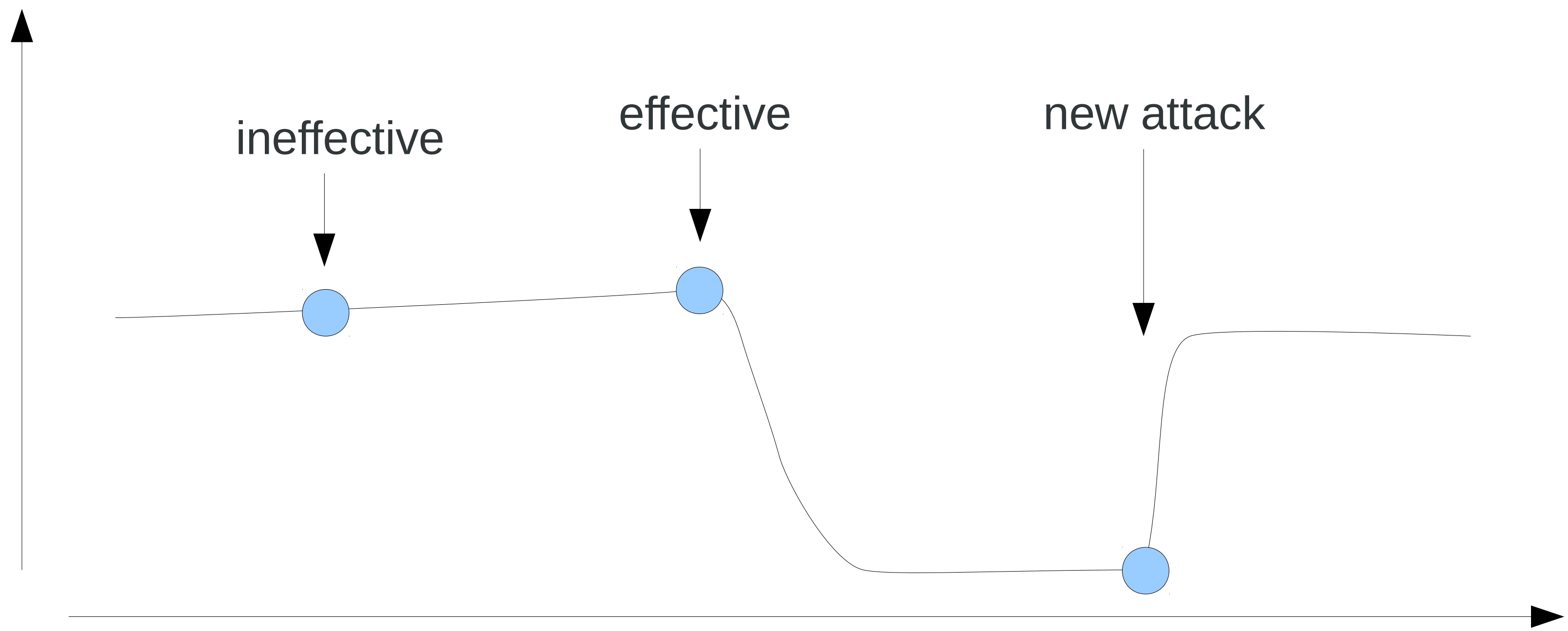
- `alloca` argument allows arbitrary stack pointer adjustment
- `-fcheck-stack` has considerable code size impact
- Some assembly required
- Use stack boundary provided by split stacks

Subscript checking for `operator[]`

- Affects `std::vector`, `std::string`, `std::array`
  - `vec[i]`
- C++ standard gives permission for bounds checking
- Library-only change has performance impact
- Further research needed
- Interim workaround: use `vec.at(i)`

Hardening and performance

- There is a trade-off
- Real-world attack data enables objective decisions



More far-reaching changes

- Improving memory safety for C/C++
  - Bounded pointers/array slices
  - Garbage collection
  - `__attribute__` annotations
  - Vtable dispatch changes (for C++)
  - Ranges instead of iterators (for C++)
- Library consolidation
- FLOSS-specific secure coding guidelines
- Better APIs?

Changing the game

- A bunch of new system programming languages
  - Go, LuaJIT, Rust
  - And a few older ones: Ada, Haskell, Java, Ocaml
- Incremental conversion requires deep embedding
  - No kernel threads, no changes to signal handlers
  - Isolated language run-time states
- This is an implementation issue.
- At the moment, only Ada and LuaJIT qualify

Vulnerabilities and side effects

- CVE-2013-0243: tls-extra certificate validation
- Haskell is a real programming language now!
- The vulnerability is in imperative code.
  - But it could have been pure/side-effect-free.
- Vulnerabilities are not necessarily side effects.

Distribution-wide defect analysis

Static analysis

- Year 2012 for Red Hat Enterprise Linux 6
  - ~600 changes (“errata”) in 340 source packages
- Before/after comparison for every errata
- Matches so far:
  - CVE-2012-3547 (freeradius)
  - Error handling improvements in PostgreSQL
  - Actual bug for psacct (837621), unixODBC (628909)

Fedora static analysis efforts

- mock-with-analysis
- “Firehose” exchange format
- https://fedoraproject.org/wiki/StaticAnalysis

Global analysis assistance

- Analysis of an entire distribution, not a single package
- Source code search engines
  - http://codesearch.debian.net
- Search for “`YAML\.load`”

Global analysis asssistence

- ELF symbol databases
  - https://github.com/vdanen/rq/
  - https://github.com/fweimer/symboldb/
- Simpler to set up than source code indexing
- Full power of PostgreSQL

Uses for symbol databases

- Joins and anti-joins point to potential vulnerabilities
- “billion laughs” denial of service with Expat
  - Program calls `XML_ParserCreate`, but not `XML_SetEntityDeclHandler`
- Privilege escalation via unsafe environment access
  - DSO defines PAM or NSS entry points *and*
  - DSO calls `getenv` or calls a function in a library which calls `getenv` (perhaps indirectly)

Improved tools for global analysis

- More detailed data than ELF symbols
  - Debugging information
  - Compiled binaries after disassembly
- Java, Python, … support
- Dynamic languages will need heuristics

Improved tools for global analysis

- Efficient search for function calls with certain arguments
  - `umask(0)`
  - `curl_easy_setopt(handle, CURLOPT_SSL_VERIFYHOST, 1L)`
  - `realpath(path, buffer)` where buffer is not `NULL`
- ELF symbols could locate binaries
- Disassembly could extract function arguments

Conclusion

- Let's try to fix `alloca`.
- Static analysis and code search engines are exciting.

Questions?

And: Please share your version control repository!