10 Years of Linux Security

A Report Card

Bradley Spengler
Open Source Security, Inc.
2020 Linux Security Summit NA

Bradley Spengler
Open Source Security, Inc.

Outline

- Timeline
- KSPP
- XLTS
- Syzkaller
- Exploitation Trends
- Recommendations

Bradley Spengler
Open Source Security, Inc.

Timeline

- Not an exhaustive timeline
- Primarily selected from changes that rose to the level of being mentioned on kernelnewbies.org
- Won’t include things that have a relation to security in some instances (e.g. ramp-up of KUNIT/selftests)
- X86-specific
    - In fact, the ARM64 and PowerPC maintainers are landing significant security functionality recently

Bradley Spengler
Open Source Security, Inc.

Timeline

- August 2010 – “Linux Security in 10 Years” presented at Linux Security Summit
  - Mentioned little mainline attention to kernel self-protection
  - 2009/2010 saw a large influx of published kernel exploits, many by me with techniques still in use today
  - Recommended:
    - Removing RWX from the kernel, eliminating information leaks, protecting sensitive kernel data (function pointers, IDT/GDT, etc)
    - Protecting against invalid userland memory accesses, refcount overflows, overflows of allocation sizes
    - Implementing CFI, preparing for data-only attacks

Bradley Spengler
Open Source Security, Inc.

Timeline

- June 2011 – kernel-hardening mailing list created
  - Helped track GSoC project for Vasiliy Kulikov
    - Port of Openwall's /proc restrictions via a mount-time 'hidepid' setting
      - Polkit/dbus-broker/etc refusing to support it:
        - https://github.com/bus1/dbus-broker/issues/207
        - https://github.com/systemd/systemd/issues/12955
      - Currently being re-implemented as a per-ns mount setting
    - Unprivileged ping sockets
      - Unfortunately, introduced some security flaws of its own
      - https://www.openwall.com/lists/oss-security/2017/03/25/1
  - During GSoC (June to August 2011), averaged 173 posts per month
    - Dropped to 131 in Sept, and 8 in Oct
    - Nov 2011 - Nov 2015, average monthly post count of 24, median of 3, 12 posts total in 2014

Bradley Spengler
Open Source Security, Inc.

Timeline

- July 2011 (v3.0) – Supervisor Mode Execution Prevention (SMEP) support added
  - Not usable until Ivy Bridge shipped in Q2 2012
- May 2012 (v3.4) – Yama LSM introduced
  - Implemented ptrace restrictions present in grsecurity
  - Originally manually stacked with other LSMs
- July 2012 (v3.5) – BPF-based seccomp introduced
  - Adopted early on by systemd and Chromium
- September 2012 (v3.6) – symlink/hardlink restrictions introduced
  - Adapted from functionality first implemented for Linux 2.0.22 by Andrew Tridgell in 1996, reimplemented shortly after by Openwall, and later by grsecurity
  - http://lkml.iu.edu/hypermail/linux/kernel/9610.2/0086.html
  - “Tightening security: not for the impatient”: https://lwn.net/Articles/503660/
  - Addresses common cases of /tmp race vulnerabilities

Bradley Spengler
Open Source Security, Inc.

Timeline

- December 2012 (v3.7) – Supervisor Mode Access Prevention (SMAP) support added
  - Not usable until Broadwell shipped in Q4 2014
- February 2013 (v3.8) – Unprivileged User Namespace support added
  - Patch to allow unprivileged users to create user namespaces was written Nov 2011, merged a year later
  - Greatly increased kernel attack surface, exposed many interfaces that previously saw little security scrutiny
  - Removal of privilege check wasn't contingent on any analysis of that attack surface, only on completion of the user namespaces work
- September 2013 (v3.11) – O_TMPFILE support added
  - Race-free temporary files, requiring application support

Bradley Spengler
Open Source Security, Inc.

Timeline

- March 2014 (v3.14) – Kernel Address Space Layout Randomization (KASLR) introduced
  - After publication of “KASLR: An Exercise in Cargo Cult Security” in March 2013: https://grsecurity.net/kaslr_an_exercise_in_cargo_cult_security
  - Large developer time sink, overstated security claims, defeated by single information leaks/generic attacks
  - As went unnoticed until April 18 2020, trivially defeated by coredumps also
- October 2014 (v3.17) – Signed kexec kernel support added
- December 2014 (v3.18) – bpf syscall added for direct eBPF use
  - In January 2016 (v4.4), unprivileged access became permitted
  - Frequent source of vulnerabilities due to churn, complexity, improper validation and complexity of validation

Bradley Spengler
Open Source Security, Inc.

Timeline

- April 2015 (v4.0) – Live patching and KASAN support added
  - Live patching already offered by various distributions at the time, e.g. ksplice
- August 26 2015 – Grsecurity stops making its stable patches publicly available
- November 2015 (v4.3) – userfaultfd() merged
  - (Ab)used by most Project Zero Linux kernel exploits
- November 5 2015 – “The Kernel of the Argument” Washington Post article published
- Coincidentally, also on November 5 2015 – KSPP announced on kernel-hardening mailing list
  - “I’m organizing a community of people to work on the various kernel self-protection technologies (most of which are found in PaX and Grsecurity)”
  - “Between the companies that recognize the critical nature of this work, and with Linux Foundation's Core Infrastructure Initiative happy to start funding specific work in this area, I think we can really make a dent.”
  - https://www.openwall.com/lists/kernel-hardening/2015/11/05/1

Bradley Spengler
Open Source Security, Inc.

Timeline

- March 2016 (v4.5) – UBSAN support added
  - Largely unused due to large amount of benign reports, barrier to entry
- March 2016 (v4.5) – Kernel memory accounting switch to whitelisting
  - Previously, allocations that shouldn't be accounted to a memory control needed a special flag
  - After "memcg: only account kmem allocations marked as __GFP_ACCOUNT", only explicitly marked allocations are accounted
  - Causes continued confusion, lack of knowledge of the change results in exaggerated isolation promises
  - https://twitter.com/dvyukov/status/1073627872366088193
- May 2016 (v4.6) – Initial Memory Protection Keys support added
  - Not usable until Skylake-server shipped in Q2 2017
  - Interface redesigned in December 2016 for v4.9

Bradley Spengler
Open Source Security, Inc.

Timeline

- July 2016 (v4.7) – Slab freelist randomization added
- October 2016 (v4.8) – weakened form of PAX_USERCOPY and GCC plugin support added
  - Plugin support added by Emese Revfy as part of CII funding
- December 2016 (v4.9) – VMAP_STACK merged
  - Weakened form of GRKERNSEC_KSTACKOVERFLOW, caused DoS or device malfunction in dozens of instances
- April 26 2017 – Grsecurity stops making its test patches publicly available
- April 2017 (v4.11) – PAX_STRUCTLEAK plugin and STATIC_USERMODEHELPER option added
  - STATIC_USERMODEHELPER inspired by auto-enabled grsecurity feature that doesn't require userland adjustments

Bradley Spengler
Open Source Security, Inc.

Timeline

- July 2017 (v4.12) – Basic support added for making LSM hook structures read-only when runtime SELinux disabling is disallowed
- September 2017 (v4.13) – REFCOUNT_FULL added, FORTIFY_SOURCE support added for some kernel string routines
    - FORTIFY_SOURCE follows earlier work by grsecurity in 2009 that was dropped as it didn't have sufficient coverage and added overhead for perfectly normal uses
        - Even earlier work in 2005 by Arjan van de Ven
    - REFCOUNT_FULL is a slower, opt-in reimplementation of PAX_REFCOUNT
        - Very recent rewrite provides performance closer to PAX_REFCOUNT, but coverage remains much lower
- November 2017 (v4.14) – Sane stack rlimits enforced on execution of privileged binaries
    - 7 reviewers, but still trivially bypassed: https://grsecurity.net/~spender/sorry_kees.c

Bradley Spengler
Open Source Security, Inc.

Timeline

- January 2018 (v4.15) – KPTI and Retpolines introduced to address Meltdown/Spectre v2
  - PTI initially only supported for x64, 32-bit support wasn't added until October 2018 in v4.19
- April 2018 (v4.16) – Additional bits of PAX_USERCOPY (slab whitelisting) added
- August 2018 (v4.18) – Unprivileged mount (within container) support added
- October 2018 (v4.19) – L1TF fixes and protection against attacker-controlled FIFOs/files in /tmp added
  - FIFO protection adapted from ancient Openwall (and later, grsecurity) feature
- May 2019 (v5.1) – LSM stacking
  - Nearly 20 years of missing LSM-based innovations because this functionality wasn't available

Bradley Spengler
Open Source Security, Inc.

Timeline

- July 2019 (v5.2) – Microarchitectural Data Sampling (MDS) fixes and reshuffling of existing hardening options out of “Kernel Debugging” area and into their own “Kernel Hardening” area
- November 2019 (v5.4) – Lockdown mode added
  - 29-patch series went through 40 published revisions by Matthew Garrett, David Howells, and others
    - https://lkml.org/lkml/2019/8/19/1190
  - Functionality already present in distros (and grsecurity) for many years
  - “Code I wrote 7 years ago is finally merged” https://twitter.com/mjg59/status/1178167356029169664
- January 2020 (v5.5) – Finer-grained ‘perf’ permission controls added

Bradley Spengler
Open Source Security, Inc.

KSPP

- Kernel-hardening mailing list statistics
  - 3638 mails in 2016
  - 5202 mails in 2017
  - 3759 mails in 2018
  - 2727 mails in 2019
  - 25% new patches submitted to the list
  - 29% updated patches submitted to the list
  - .5% resent patches
  - 19% replies to new patches
  - 13.5% replies to updated patches
  - .3% replies to resent patches
  - 2.7% involved the KSPP list due to get_maintainer.pl
  - 8.5% all other security discussion

Bradley Spengler
Open Source Security, Inc.

KSPP

- Top contributors in % of overall emails
  - keescook@chromium.org : 15.05
  - thgarnie@google.com : 3.44
  - ard.biesheuvel@linaro.org : 2.75
  - mark.rutland@arm.com : 2.49
  - jason@zx2c4.com : 2.29
  - mic@digikod.net : 2.04
  - me@tobin.cc : 1.85
  - luto@kernel.org : 1.61
  - labbott@redhat.com : 1.60
  - gregkh@linuxfoundation.org : 1.54
  - luto@amacapital.net : 1.51
  - torvalds@linux-foundation.org : 1.47
  - samitolvanen@google.com : 1.45
  - igor.stoppa@huawei.com : 1.38
  - peterz@infradead.org : 1.33
  - alex.popov@linux.com : 1.32
  - elena.reshetova@intel.com : 1.31
  - rick.p.edgecombe@intel.com : 1.23
  - jannh@google.com : 1.19
  - s.mesoraca16@gmail.com : 1.14
  - yanaijie@huawei.com : 1.08
  - igor.stoppa@gmail.com : 1.05
  - re.emese@gmail.com : 1.00

Bradley Spengler
Open Source Security, Inc.

KSPP

- “Other” category is where one would expect to see the most interesting discussion
  - Proposing new ideas/brainstorming prior to deciding on an implementation and submitting a patch
  - Very little of that (8%) happens on-list
- Patches themselves make up over half of all emails to KSPP list
- One third of emails are reviews of those patches
- Mails specifically mentioning KASLR make up nearly 9% of all mails
- Only 9% of mails were sent on the weekend
- Very long tail of people sending mails to the list
  - Only the 23 on previous slide were >= 1%, all 614 remaining were < 1%
  - Of that 614, only 17 between .5% and 1%
- For your own analysis (and reproducing tests):
  - Mail spool: https://grsecurity.net/kernel-hardening-cleaned.gz
  - Analysis script: https://grsecurity.net/KSPP-email-analysis.py

Bradley Spengler
Open Source Security, Inc.

KSPP

- Focused entirely on merging new security functionality into the latest upstream kernel
- Some of these changes involve large rewrites of kernel code
  - VLA removal (manual)
  - kmalloc(n * size) -> kmalloc_array(n, size) (automated)
  - refcount_t (manual)
- While this improves security of the latest upstream kernel, eventually improving security for many (perhaps even most) Linux users years from now, it doesn't benefit the majority today
- The large rewrites don't meet –stable inclusion criteria, don't get backported, similar to the majority of KSPP work
- No technical reason the functionality can't be backported and made available to a wide audience

Bradley Spengler
Open Source Security, Inc.

KSPP

- One trend since 2018 is that a lot of security work is bypassing the KSPP
  - PTI/retpolines/nearly all other Intel-flaw-fixes got developed in private without the KSPP list
  - Architecture maintainers pushing through some security changes on their own
    - Especially PowerPC, ARM64
- While this may signal waning relevance of the KSPP, good sign that some maintainers have high regard for security and accelerated development of security features
- Some developers that deserve special mention:
  - Will Deacon (ARM64)
  - Michael Ellerman (PowerPC)

Bradley Spengler
Open Source Security, Inc.

XLTS

- September 28th 2017 – Google announces at Project Treble presentation that Greg KH will increase support of LTS kernels from 2 years to 6 years
- Seemingly no prior public discussion of this move
- 3x longer support period, with backports being increasingly more difficult over time
- Nearly a year later, "What Stable Kernel Should I Use?" is published by Greg KH
- Contained this admission for the first time:
    - "There is one huge caveat when using a kernel like this. The number of security fixes that get backported are not as great as with the latest LTS release, because the traditional model of the devices that use these older LTS kernels is a much more reduced user model. These kernels are not to be used in any type of "general computing" model where you have untrusted users or virtual machines, as the ability to do some of the recent Spectre-type fixes for older releases is greatly reduced, if present at all in some branches."

Bradley Spengler
Open Source Security, Inc.

XLTS

- Why aren't known security issues being fixed in kernels marketed as being supported for 6 years?
- For the answer, we need to look at the process on the –stable mailing list

  - *Subject*: FAILED: patch "[PATCH] drm/qxl: qxl_release use after free" failed to apply to 4.9-stable tree
  - *From*: <gregkh@xxxxxxxxxxxxxxxxxxx>
  - *Date*: Mon, 04 May 2020 10:09:10 +0200
  - *Cc*: <stable@xxxxxxxxxxxxxxx>

```
The patch below does not apply to the 4.9-stable tree.
If someone wants it applied there, or to any other stable or longterm
tree, then please email the backport, including the original git commit
id to <stable@xxxxxxxxxxxxxxx>.

thanks,

greg k-h
```

- Even if a backport is trivial, any git cherry-pick failure results in the above

Bradley Spengler
Open Source Security, Inc.

XLTS

- What happens if no one volunteers to manually backport the security fix and submit it upstream?
  - The vulnerabilities simply do not get fixed
- Patches going to –stable often seem to receive little review
- Does not appear to be a process in place for ensuring backports get applied with commits that list it in their "Fixes" tag (i.e. known, broken fixes being applied)
- https://www.spinics.net/lists/stable/msg365359.html
  - KVM fix marked for –stable was backported all the way to 4.4. No one noticed the fix pulled in a brand-new 8000+ line file to older stable kernels
  - Took 15 days to be fixed in all affected kernels once it was introduced

Bradley Spengler
Open Source Security, Inc.

XLTS

- https://grsecurity.net/the_life_of_a_bad_security_fix
  - Remote heap overflow in mwifiex driver
  - Fix introduced a RCU lock imbalance (easily spotted when reviewed outside of diff context)
    - In this case, the RCU fix wasn't committed until Jan 27, despite being authored on Jan 6
  - Similar fix made to libertas driver, introducing a similar RCU lock imbalance
    - RCU fix authored Jan 14 2020, committed Jan 27 (with a proper Fixes tag)
    - Bad fix still applied to stable kernels on Jan 29, RCU fix not applied until Feb 14

Bradley Spengler
Open Source Security, Inc.

XLTS

- https://grsecurity.net/teardown_of_a_failed_linux_lts_spectre_fix
  - Spectre v1 fix for ptrace from May 2019 caused a compile warning fixed by Linus in a July 2019 “evil merge”
  - Rather than take that correct fix, -stable attempted to fix it differently without review
    - “Ah, didn't realize it was fixed during the merge, will do the same thing here and tweak the original patch.”
  - The different fix was obviously incorrect: it attempted to do masking on the index after the index was already used
  - Incorrect fix was then backported to all stable kernels later in July
  - Not publicly noticed or fixed by anyone until our September 2019 blog

Bradley Spengler
Open Source Security, Inc.

XLTS

- 4.4 XLTS will be supported for two more years (until Feb 2022)
- Based on the previous information, we know needed security patches are being missed
- Syzkaller could help find some of these issues (as well as errors in backports that have been applied)
- Let's see what syzkaller.appspot.com looks like for that kernel:

syzbot

Linux 4.4

fixed bugs (0)

Instances:

Name	Active	Uptime	Corpus	Coverage	Crashes	Execs	Kernel build			syzkaller build		
							Commit	Freshness	Status	Commit	Freshness	Status
ci2-linux-4-4	234d	2h26m				broken	5f090d83	237d		1e9788a0	248d	

- For as long as we've been looking, it's always looked like the above, not reporting anything
  - Not even the 4.4 Android kernel page will help, that hasn't been updated in 460+ days
- What the public thinks is happening in terms of support for these kernels simply doesn't match reality

Bradley Spengler
Open Source Security, Inc.

XLTS

- Our observations from having an independent stable backport process:
  - As of our final 4.4 patch, had at least 1250 security-relevant fixes missing in the upstream 4.4 LTS
  - https://twitter.com/grsecurity/status/1243523605264285696
  - AUTOSEL is getting better at spotting security fixes that miss the proper tags that previously only we were backporting
    - To see the full benefits, however, number of people involved in backporting has to increase, otherwise they simply get dropped at the –stable mailing list level
    - The above situation doesn't appear to have noticeably improved
    - https://lwn.net/Articles/805473/ : Dan Carpenter - "Everyone is over worked."
  - Patches selected by AUTOSEL could use more review however, to make sure it's proper to apply them to older kernels and whether they need adjustments regardless of whether they cherry-pick cleanly
    - https://www.spinics.net/lists/bpf/msg11641.html
      - Was exploited as part of ZDI's pwn2own

Bradley Spengler
Open Source Security, Inc.

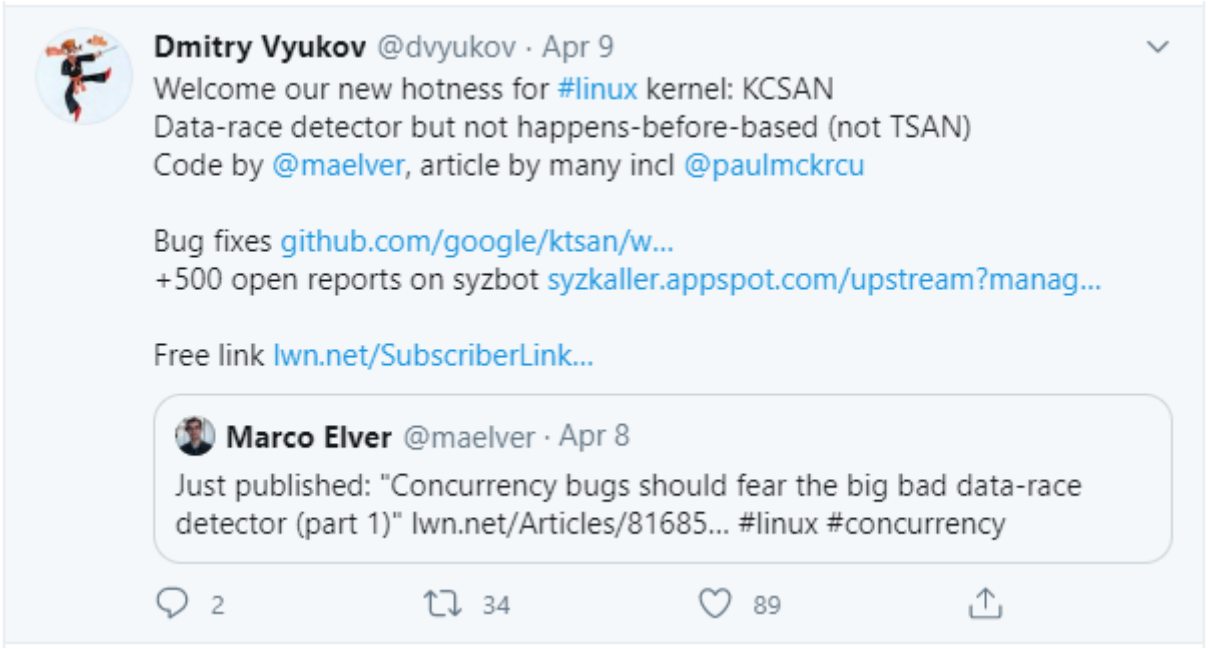
Syzkaller

- In the past 10 years, the adoption of syzkaller has had possibly the largest impact on upstream kernel development
  - Developers notified when an issue in their code is found
  - Has integrated well with the existing development processes
- Constantly improving, integrating new debugging/testing features
  - KCOV_ENABLE_COMPARISONS
    - Helps reach more code paths (and more quickly)
  - KASAN
    - Detects UAFs and other OOB memory accesses
  - KMSAN
    - Detects uninitialized memory accesses
  - KCSAN
    - Detects concurrency issues

Bradley Spengler
Open Source Security, Inc.

Syzkaller

- A bit of a double-edged sword
  - Custom syzkaller instances can find vulnerabilities syzkaller.appspot.com does not
  - Rapid pace of kernel development means there is always a plethora of unfixed issues
    - Particularly when syzkaller reaches new areas of the kernel (USB, etc)
    - https://twitter.com/dvyukov/status/1248241187854696450



Bradley Spengler
Open Source Security, Inc.

Exploitation Trends

- Data-only attacks are on the rise, as public attacks continue to focus on the path of least resistance
- STATIC_USERMODEHELPER doesn't seem to have caught on at all, UMH paths continue to be targeted
  - Kernel itself performs the task of breaking out of any container and executing a process with full privileges
- Public, unfixed syzkaller reports a good source of information
  - Tweaks to syzkaller, differences in configs discover vulnerabilities that do not appear in the Syzkaller dashboard the public uses (http://syzkaller.appspot.com/)
- Attack surface exposed by unprivileged user namespaces isn't decreasing anytime soon
  - Even more functionality being exposed:
    - 2abe05234f2e l2tp: Allow management of tunnels and session in user namespace
    - 4a92602aa1cd openvswitch: allow management from inside user namespaces
    - 5617c6cd6f844 nl80211: Allow privileged operations from user namespaces
  - "Does this newly-allowed code pass existing fuzzing tests?" doesn't appear to be a consideration for enabling such functionality

Bradley Spengler
Open Source Security, Inc.

Exploitation Trends

- Project Zero exploits mainly abusing FUSE and userfaultfd functionality for exploit reliability for many years
  - https://googleprojectzero.blogspot.com/2016/06/exploiting-recursion-in-linux-kernel_20.html
- Attackers still don't care about whether a vulnerability has a CVE assigned or not
  - https://googleprojectzero.blogspot.com/2019/11/bad-binder-android-in-wild-exploit.html
  - Above vulnerability was exploited with an in-the-wild 0day allegedly developed by NSO Group
  - Illustrates that security gaps will continue to exist with downstream users even if upstream's process works correctly
    - Fix here did get applied to –stable kernels
    - Commit message mentioned the UAF, contained syzbot Reported-by, and CC'd stable@
- Interpreter security will become increasingly important
  - O_MAYEXEC and the various interpreter work from Microsoft/Red Hat and others are necessary steps that ideally should have started long ago

Bradley Spengler
Open Source Security, Inc.

Exploitation Trends

- Reactionary security doesn't work anymore – outside of Project Zero, few exploits are published publicly
  - Today's security professionals (especially the younger, newer generation) expect to be paid for their work
  - Companies like Zerodium will buy exploits for widely-used software, including Linux
  - Several other security companies develop and sell these exploits with in-house talent
- Techniques more important than individual vulnerabilities
  - Exploit shops plug the vulnerability into their existing exploitation frameworks
- Can't wait for a public exploit to start thinking about how to improve security
  - The mouse in the cat and mouse game has slipped into the hole in the wall
- Automatic Exploit Generation
  - Current state of art is effective unless CFI is employed
  - https://i.blackhat.com/eu-19/Wednesday/eu-19-Chen-Hands-Off-And-Putting-SLAB-SLUB-Feng-Shui-In-A-Blackbox.pdf

Bradley Spengler
Open Source Security, Inc.

Recommendations

- KASLR has been repeatedly proven to be security theater
  - Operating under the assumption of no information leaks for the Linux kernel is a losing bet
  - Continues to consume a lot of developer time across multiple architectures
  - Confused printk %p hashing effort in support of KASLR has hampered bug reporting
  - Takes up an outsized focus on KSPP mailing list
    - PIE support to extend KASLR (for an extra 2 bits of equally-leakable entropy) currently on v11 of patch series that started in August **2017**
- Consider modern/realistic threat models and implement security that can be effective/performant under those models
  - Why is FG-KASLR being discussed when CFI has already been demonstrated for years?

Bradley Spengler
Open Source Security, Inc.

Recommendations

- Seemingly “simple” mitigations can create sunk cost fallacies
  - Arises when they cannot offer the security they purport under realistic threat models
  - Each workaround for an attack can add performance hit
  - Had all the issues been known from the beginning, mitigation wouldn’t have landed
  - After years of work, will you remove an indefensible mitigation, or keep adding to the hacks and performance hit?
  - FG-KASLR will prove to be an example of this
- Resist the thinking of “we have to do something / this is something / let’s do this”
  - http://addxorrol.blogspot.com/2020/03/before-you-ship-security-mitigation.html
  - With dearth of public exploits, developers only have to convince each other
  - Little to no independent/expert evaluation (outside of Jann Horn)

Bradley Spengler
Open Source Security, Inc.

Recommendations

- Improve the quality of benchmarks for security features before inclusion into the kernel
  - KAISER once claimed a 0.1% performance hit
  - 1% performance hit was claimed for STACKLEAK (not by us)
  - Retpolines ended up having a large impact on network throughput
  - Benchmarks mostly appear to involve kernel compilation, a task that only spends a tiny percentage of its time in the kernel, diluting any performance hit that exists
  - Ideally, a suite of benchmarks should be performed on a farm of consumer/server CPUs of at least the past 5 years

Bradley Spengler
Open Source Security, Inc.

Recommendations

- Development organization still focused on patches and mailing lists
  - In many instances (including some of the examples provided) information was available online pointing to a bug or a proper fix
  - Current process involves essentially ignoring that external information until it falls in line with the standard Linux development process
  - Maintainers exist to corral patches into the upstream kernel – what about corralling information into a centralized location to improve the quality of the kernel?
    - Example: a group in charge of ensuring stable kernels apply applicable security fixes
    - Problem: the public didn't sign up for the behind-closed-doors decision to extend stable kernel support from 2 years to 6 years for the benefit of phone vendors, certainly not for free

Bradley Spengler
Open Source Security, Inc.

Recommendations

- Fail fast, iterate faster
  - A lot of review for security patches is trivial nitpicking and can-kicking
  - Appears to be a different set of standards for merging non-security features vs even optional security code
  - Is having the KASLR PIE/S.A.R.A. code dragged out over 3 years an effective use of developers' time?
  - When an abysmal patch is presented that has no chance of being merged in any form, is it really helpful to lead the submitter on by suggesting trivial nitpicks and typo fixes and asking them to put out a v2/v3/v20?
- The time to decide on an "unprivileged" sysctl for an easily abused new feature is when the feature is first introduced, not years after the necessity is obvious via multiple exploits (unpriv userns, userfaultfd, etc)

Bradley Spengler
Open Source Security, Inc.

Recommendations

- Find a way to fund security work again
  - CII is long dead, but no obituary was ever posted
    - https://twitter.com/TomRittervg/status/1181572896579305473
    - Last announcement in 2017
    - Last mail on cii-discuss mailing list was December 2017 regarding CII support for OpenSSH having ended in 2016 but still being listed on the CII website
      - No reply, OpenSSH and OpenSSL still falsely listed as currently-supported projects today
- Actually practice what you preach

Millions of Dollars of Shared Vigilance

The stakes have never been higher for open-source software security. With millions of people around the world relying on open source software — and vulnerabilities like Heartbleed putting everyone at risk — it's time to change the way we support, protect, and fortify open software. With our Core Infrastructure Initiative, we're taking a collaborative, pre-emptive approach for strengthening cyber security. Many industry giants signed on to harden the security of key open source projects.

Thank You!

Questions?

Grsecurity is created by
Open Source Security, Inc.