OS Security Is Hard: Why All the Fuzzers in the World Won't Change the Way Platform Security Is Failing Us

Alex Ionescu (@aionescu)

USENIX 2020 – WOOT

Bio

- VP of Endpoint Engineering at CrowdStrike
  - Previously worked at Apple on iOS Core Platform Team
- Co-author of Windows Internals 5th-7th Editions
- Reverse engineering NT since 2000 – lead kernel developer of ReactOS
- Instructor of worldwide Windows internals classes
- Author of various tools, utilities and articles:
  - NTFS On-Disk Structure, Visual Basic Metadata Format
- Conference speaking:
  - SyScan 2012-15, Infiltrate 2015, 2019, OffensiveCon 2018-19
  - NoSuchCon 2013-14, Breakpoint 2012, EkoParty 2017
  - Recon 2010-19, EuskalHack 2017, CanSecWest 2018
  - Blackhat 2008, 2013, 2016, 2018, BlueHat 2016-17
- For more info, see www.alex-ionescu.com

Context

- I've had the chance to see infosec evolve, especially the way software bugs are treated
- In NT4, Mark Russinovich writes NTCRASH.EXE, followed by NTCRASH2.EXE in Windows 2000.
  - The word "fuzzer" doesn't even exist yet
  - Bugs are slowly fixed in "Service Packs" and fixes are publicized in marketing materials
- In Windows XP SP2, the creation and enforcement of the Security Development Lifecycle (SDL) revolutionizes Windows development
- Thanks to the work of Katie Moussouris, Microsoft begins paying bug bounties for RCEs in Internet Explorer
- For about three years, an *infoleak* in Windows vNext gets you $10,000.00 US
  - Starting last week, it's $2,000 USD for an RCE*

Outline

- Bug Bounty Economy
- InfoSec Schools
- Mitigations != Solutions
- Parting Thoughts

Bug Bounty Economy

Bugs at any Cost

Before the Bug Bounty

- Finding bugs wasn't always that obvious
  - Few, if any, automated tools
  - Little knowledge on how to find bugs
  - Bugs without obvious exploits, or at least without a PoC proving severity, were never prioritized
  - The know-how on exploitation, and the various techniques we now take for granted did not yet exist (i.e.: ROP)

And there wasn't a legitimate, paid, public, identifiable mechanism for those who had these skills to submit vulnerabilities – so most of them went to black market exploit brokers

- Yup, we should **thank** everyone who got us here today instead

Today?

- Microsoft, Apple, Google, Facebook and hundreds of other software companies and cloud vendors pay millions of dollars for bug bounties
  - Microsoft paid out ~35M in FY20
- Platforms such as BugCrowd and HackerOne now manage the discovery and triage process for over 10K products and services…
  - This has reduced the effectiveness of black markets and/or increased their costs
  - BugCrowd paid out ½ million dollars in a single week in November
- Microsoft will fix an OS-level bug in 3 months or less (usually!), not 1-2 years
  - And Web service bugs can take weeks
- What kind of bugs are being found, though?

Economic Truths

- One can make $500-$1,000 US a pop by finding XSS bugs that let you pop a message box
- Automated tools to find these types of bugs are readily available and can easily find 5 bugs a month
  - Just find a few obscure sub-domains owned by companies that didn't realize they're public (marketing, etc)
- So on average, and with a bit of luck, you can make $2,000-$2,500 US a month, or $25,000.00-$30,000.00 US a year
  - In a lot of **industrialized countries**, that's what a software intern might make
  - In countries that are in development, this is 5-10X the average salary of a full-time employee…

This Leads To…

- A shortage of "qualified software developers"
- So even more outsourcing in countries with high degrees of STEM talent in the field (China, Romania, India, Israel, etc…)
  - But now people can make 10X the money finding bugs instead of writing code
- So what happens to the local innovation in this countries?
  - How can a local company hire local developers to work on local innovative software, when now there's not only foreign companies offering better economical and social incentives for the same job, but also local websites that will just pay 10X for them to sit at home and find bugs?

Total Lack of Resources

- Bug trackers of open source projects are full of bugs
  - Drupal, 7-Zip, Notepad++, etc…
- Nobody has time to fix them, and there's never any PRs
  - Developers prioritize feature development over bug finding and fixing – this has almost never changed
  - And there's not enough developers anyway – FOSS isn't how you usually feed a family!
- In many cases, developers actively refuse to fix security issues, even if you give them a pull request
  - *Slows things down, adds compatibility issues*
- It's OK – the government is here to help you

Nationalized Bug Bounties

- In the EU, there's now up to 1M EUR being offered as bug bounty money for various "critical" FOSS projects
  - But nobody is paying the developers for the fixes
  - Who will even triage the bugs?
- Software isn't lacking in bugs, it's lacking in quality and security – finding more bugs doesn't fix that

Cyber Schools

Learning to break, not build

Fuzzer Schools

There's now more and more schools and programs which are only teaching offensive security and bug finding

17-year-old kids with their OWASP, MCSE, A+, CISSP finding $5,000.00 US bugs each month, maybe hitting a $25,000.00 jackpot occasionally

Meanwhile, developers their parent's age are making half that money trying to fix the bugs before retirement

And it's not only commercial schools…

#Fail(ure) Camps

- In some countries you find entire government programs that teach kids to quit/delay school in order to learn infosec instead
- Teens are being paid by their government to participate, and sometimes foreigners are brought in (with taxpayer money) to "teach" the kids
  - Hey PHP kids, let's learn about SMM ACPI Hyper-V RCEs!
- When it's all said and done, they're guaranteed pre-reserved places in large local (and multi-national) companies
- No, I'm not making this up, I participated in a few of these…

Mitigations Aren't Solutions

Case Studies and Thoughts

NtUserGetBitmapOemSize

- Here was the Windows 10 1809 system call I harped on:

```
OEM_BITMAP_INFO g_OemBitmapSizes[72];

size_t NtUserGetBitmapOemSize (size_t OemIndex)
{
    return g_OemBitmapSizes[OemIndex].Size;
}
```

- I pointed out how outside of the human element (code reviews), everything from a static analyzer, to the world's simplest fuzzer should've caught this
  - A more important point that was missed was me asking – why does such a function even need to be a system call?
- People (correctly, I guess?) criticized that I was taking a one-off example (really?) to sh*t on Microsoft – that's not the case
  - Let's look at a few more examples – including some non-Microsoft ones!
  - I think it's silly that this bug paid out 10K – and write would've been 20K

RtpFrameReferenceFinder::MissingRequiredFrameVp9

- Here's a lovely bug in WebRTC found by GPZ (Natashenka)

```
size_t missing_frames_for_layer_[5];
size_t temporal_idx = info.gof->temporal_idx[gof_idx];
for (size_t l = 0; l < temporal_idx; ++l)
{
    auto missing_frame_it = missing_frames_for_layer_[l].lower_bound(ref_pid);
}
```

- The bug was in the population of `temporal_idx`, which could have values as high as 7 since 3 bits from the stream were parsed
- This was fixed, but to this day, there's still not a single `assert()` anywhere in the array dereferences

NtCreateEnclave

- One of a half-dozen recent finds by Waleed Assar:

```
NTSTATUS NtCreateEnclave(..., void** BaseAddress, size_t InfoLength) {
    if (InfoLength != 0) {
        if (ExGetPreviousMode() == UserMode) {
            ProbeForWritePointer(BaseAddress);
        }
    }
    status = MiCreateEnclave(..., &localBase);
Exit:
    if (NT_SUCCESS(status)) {
        *BaseAddress = localBase;
    }
    return status;
}
```

- The function is new to Windows 10 Threshold 2 for SGX support, but was not vulnerable until after VSM Enclaves were added in 19H1
  - New enclave type made the SEH checks for BaseAddress encapsulated only under the "`InfoLength != 0`" check, which is now optional

CVE-2020-1337

- This is the redux of the PrintDemon vulnerability that Yarden Shafir & I blogged about – colliding with SafeBreach Labs' finding
  - The original bug (CVE-2020-1048), was "fixed" by moving a client-side file system validation check to become a server-side file system validation check
- This second finding (which we *also* collided on, including with a few other researchers!) is that the server-side check is vulnerable to file system redirection attacks
  - How is this not an obvious thing to fix when moving the check from an unprivileged client to a privileged server, when $3M USD+ of bug bounty money was paid on such attacks in FY20?

How Did We Get Here?

- Each of these bugs had code reviews, static analysis, and fuzzers running
  - "Someone forgot to add the function to the fuzzer (it's a manual process)"
  - "The fuzzer doesn't run on SGX/HVCI hardware"
  - "It takes 3 months to look at the fuzzer's output – we're backlogged"
  - "Warning level 4 isn't enabled for this project"
  - "We prioritize fuzzing brand-new or really old code, not recent code"
  - "Nobody thought about fuzzing this attack surface"
  - "The servicing team's guiding principle is to minimize the lines of code and impact of any fix"

It's OK!

- I'm not here to whine about fuzzers (clearly, Waleed's is working great!)
  - I hear the next talks got some good stuff too
- Or all the complicated things that go in an engineering team's decision making and prioritization process
  - Hint: unlike 99% of infosec folks, I run an entire engineering org – I get it!

So then what?

- ~~"Leverage MPX in the fuzzer"~~
- ~~"Hire a team to write more fuzzers"~~
- ~~"Run the fuzzer on SGX hardware"~~
- ~~"Fuzz more often"~~

STOP! PUT DOWN THE FUZZER*!

* I'm not suggesting these are bad or invalid ideas, but they attack the symptom, not the disease

Crazy Thoughts

For a better future?

Education?

- Let's not repeat the mistakes of the past
  - Microsoft lowering their bug bounties may lead to a return to black market growth…
- The complete and total lack of security awareness in the 7oies and 8oies is what led to today's insecure code, tooling, and mentalities
  - I'm not advocating some nostalgic return to the days of yore!
- But my college-aged friends are still learning *strcpy*
  - The ones are ivy-league schools where they teach C – otherwise they're probably learning `OSString::Copy(OSString*)` or worse, not learning about strings at all)
- We need to revolutionize computer science instruction in schools
  - Most of it is outdated, useless, and insecure
  - I **hope** Google's 6-month program addresses this – and doesn't make it *worse*

Licensing & Training?

- The mechanical engineering equivalents of the computer bugs shown earlier, in a bridge design, would amount to criminal negligence
- Using high-school level formulas for objects in free-fall (ignoring air resistance, etc.) – the equivalent of *strcpy* in aeronautics – would get you stripped of your license
- I'm not advocating for "you need a special license and degree to use a compiler and innovate in the Valley" – this is the common strawman
  - But maybe, just maybe, if you hire people to write software that runs on 1B+ devices, they should have some training & responsibility?
    - Pilots can't fly random planes whenever they want to, even after finishing flight school – and they need to keep their skills updated yearly

Incentivize Creativity?

- I'm not saying reverse engineering / bug finding isn't creative…
  - … but it's fundamentally a process of taking someone else's work apart
- If everyone gets paid $300K a year and is guaranteed a job, taking stuff apart…
  - …it follows that nobody will want to get paid $50K a year struggling to find a job to build stuff
- Yes, our society has loads of highly difficult, laborious, underpaid jobs, contrasted with certain high-paying blue-collar jobs that are "easy"
  - But they're usually somewhat balanced with education, and never in direct inverse relationship with each other
- How about "code bounty" programs?
  - "Pay for the fix, not the bug" – such programs are now starting!

Conclusion

- We live in a highly polarized society these days and everyone wants to violently agree or disagree with an extreme viewpoint
  - "Alex says fuzzers are bad and bug bounties are bad"
- All I'm trying to suggest is that we're over-focusing on
  - Finding bugs & creating better processes for finding bugs
  - Discussing why we didn't find enough bugs & how to find more bugs
  - How to build systems where bugs don't matter & exploits for bugs are too hard
  - The right economic model of how and when to pay for people finding bugs
- At the expense of figuring out ways to
  - Teach, train, and incentivize humans to write less bugs
  - Use programming languages and paradigms that make it harder to write bugs
  - Maybe possibly even have less humans writing code (aka bugs)?
- Microsoft's recent interest in Rust, the emergence of "bounty for fix" programs, and some of the lowering of bounty payouts makes me hopeful

Thank You

Q & A