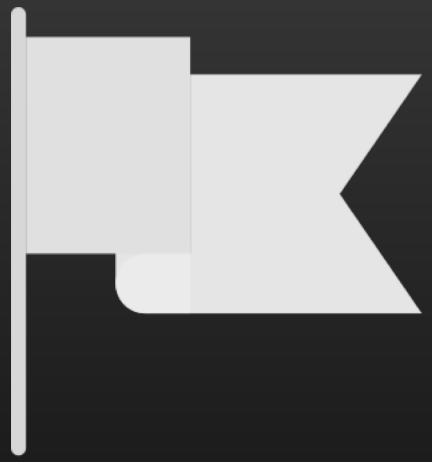
Hacking Competitions

a Grand Unified Theory

Capture the Flag



Pwn2own-style

HACK2WIN Extreme!

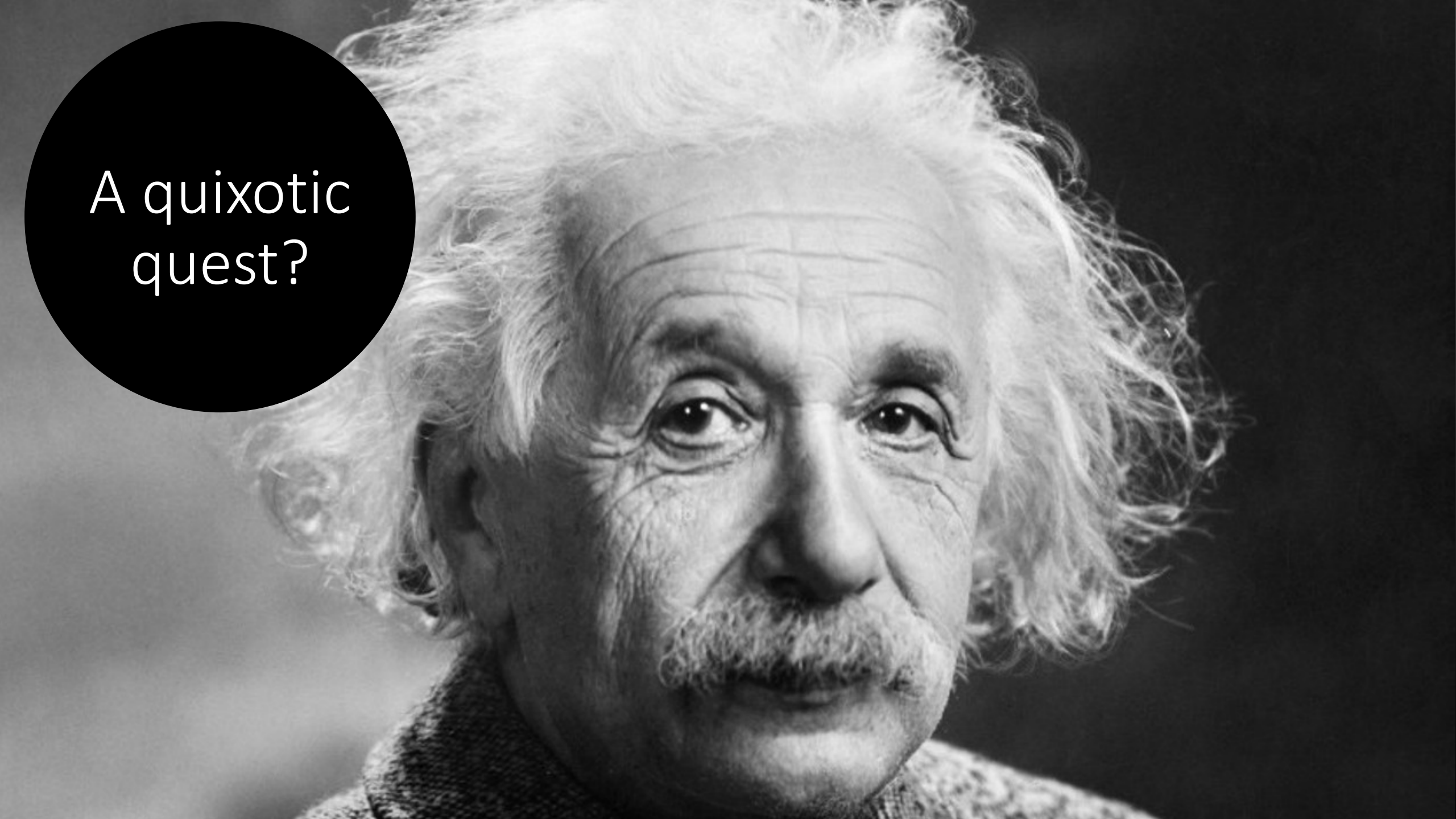
PWN2OWN

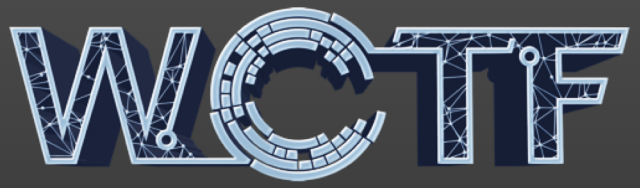
Bug Bounty

bugcrowd

hackerone

bevx


A quixotic quest?



WE CREATE THE FUTURE

2018.7.6 – 2018.7.8

BEIJING · CHINA

Capture The Flag

Jeopardy

Attack-Defense

CTFs are great, but way too many vulndev or RE tools simply do not fly on non-CTF problems. Would love to see CTFs ramp up pwnable sizes to improve general tooling.

The dissonance between CTFs and real-world software has always turned me off to them. Contrived challenges often lack the nuance of a real developers mistakes.

Plus in a CTF, there *is* a bug. That certainty isn't the same in real software.

What are the goals?

CTF

- Educational
- Approachable
- Challenging
- Fun

Pwn2Own

- Fix bugs in real software
- Evaluate current mitigations
- PR
- Fame and Fortune

Has CTF been successful?

@lokihardt

@rz_fluorescence

@nedwilliamson

CTFs: Educational

Capture the flag competitions are a form of **deliberate practice.**

Deliberate Practice

How you practice is as important as how much you practice

You must practice at a level that is slightly more challenging than your current skill

Practice requires focused attention

Professor K. Anders Ericsson

bevx

CTFs: Educational

CTFs are a form of “Deliberate Practice”

- PlaidCTF, WCTF, and others continue to produce difficult challenges that force participants to stretch their skills
- CTFs require “focused attention” from the participants
- Write-ups and solutions from other participants provide feedback that further improves skills

CTFs: Approachable

A CTF that is too difficult is not fun or useful for a participant.

- Not all CTFs and CTF players are **equal**. Some people have more experience and a higher skill level.
- We need to be honest about the difficulty level for a CTF and encourage players to pick CTFs that are appropriate for **them**.
- Easy and hard CTFs are plentiful. High-quality CTFs that are in between are much rarer.

CTFs: Challenging

CTF exploitation is as difficult as real-world exploitation.

- Modern CTF binaries typically include: ASLR with PIE, NX, FORTIFY_SOURCE, and stack canaries.
- Heap overflows, use-after-free, and info leaks are now the norm.
- Linux challenges require the exploiter to understand glibc internals and develop exploitation strategies on-the-fly.
- Real-world software with their heaps and mitigations.

CTFs: Challenging

Avoid complexity that is just for the sake of wasting time

- Tasks should be challenging because you require new skills or growth.
- Avoid random obfuscation of binaries just to make a challenge take longer to solve.
  - Obfuscation itself can be a good challenge but map it to something real.
- When writing a challenge, ask yourself what you want people to learn

CTFs: Fun

In the end, CTF is a competitive game, not a job.

- While some CTFs have a cash prize, many do not.
- CTFs are time bound which makes it easier to incorporate with life.
- In-person CTFs provide an opportunity to meet and hang out with people who share the same passion for hacking.

CTF + Pwn2Own

What elements of Pwn2Own competitions are important?

- Fixing bugs in real software
- Evaluating current mitigations
- Provide examples of state-of-the-art exploits

CTF + Pwn2Own

How can we incorporate these elements into CTFs?

Fixing bugs in real software

- Use real software with real bugs in CTFs
- CTF organizer must find at least one exploitable bug first
- The software should not have a monetary reward for finding bugs
  - Bug bounties, Pwn2Own, interest from 0-day vendors, etc.

CTF + Pwn2Own

How can we incorporate these elements into CTFs?

Evaluating current mitigations

- CTFs already do a good job of this with Linux
- iOS and Windows challenges would help test mitigations that are specific to those platforms
- New control flow integrity techniques in compilers are ripe

CTF + Pwn2Own

How can we incorporate these elements into CTFs?

Examples of state-of-the-art exploits

- Finding vulnerabilities can be time-consuming
- Use patched vulnerabilities to develop 1-day exploits
  - Exploits must still bypass mitigations and work remotely
- Is there a way to encourage “weaponized” exploits?
  - Highly reliable (>95% success rate)
  - Resilient to minor version changes

Pwn2Own-style will always be different

- “Unlimited” time to find a vulnerability, exploit it, and test
- Bug hunting requires extra time that does not fit within a weekend
- Developing a 0-day exploit for Chrome, iOS, or Windows is always more glamorous than winning a CTF
- Higher cash prize because competing against 0-day vendors
  - $100k from Pwn2Own is less than $250k market price, but > $0

Pwn2Own and Bug Bounties

- The reward does not generally match with the work required
- If you are exchanging cash for 0-day, then at a minimum researchers should be sufficiently paid for their time
- Companies should generally value exploits and bugs higher
  - Zerodium prices x2 is a good approximation of current market

- Globalization of vulnerability research
  - Salaries in some parts of the world are lower than others
  - Good for companies, may be good or bad for researchers

Real World CTF

- A recent CTF hosted by Chaitlin Tech
- Demonstrated the potential for using real software in a CTF, in a way that is still approachable and fun
- In Dash, a macOS app, multiple 0-days were found by teams and reported to vendor who quickly fixed them
- The result: a real difference for the security of real users without any additional burden on the CTF players

RWCTF-esque

- Other CTFs should consider a similar style for some challenges
- It should be clearly marked that finding a 0-day is intentional, and that the organizer has an exploit that works
- Avoid software that is annoying to setup or difficult to reverse engineer
- Emphasize open source software as a way to give back to the computing community at-large

9 years of playing CTF

- Joined PPP in 2009
- 1st place at Defcon CTF four times
- Played in many CTFs and host PlaidCTF every year
  - CSAW
  - Codegate
  - PhDays
  - SecuInside
  - WCTF

9 years of playing CTF

- I agree with the criticism that CTFs tend to focus on *tricks*
  - As a player, I want to actively learn and not be stumped by a *trick* for hours
- I play less CTF now because I get to grow my skills at my day job
  - I try to incorporate what I learn into PlaidCTF if possible
- The competitive nature of CTF and the opportunity to play with and against my friends and coworkers keeps me playing CTF

Summary

- CTFs are supposed to be fun and educational, not a job
- CTF organizers are busy, we should avoid making that task harder
- Encourage CTFs to use real world software
- A challenge can use 1-day vulnerability and still have a higher purpose
- If a CTF can help improve the security of open source software by exposing it to elite hackers, it is a win for computing

Thank you!

Send hate mail to andrew@theori.io or @zoaedk on Twitter