

Who Is Bobby Tables? Exploring Security with



Zach Zagorski
MIT Splash, Fall 2020

“Course Staff”

Me (Zach):

- Brown University '17, Computer Science
- Software Security Engineer, intentionally unnamed large tech company
- Help other engineers write code that is “secure by default”

Danna:

- Helping out, calling on students

Outline

1. Philosophy
2. Security for everyday humans
3. Security for programmers

What is cybersecurity?

OUR OVERALL FY2015
CYBERINTELLIGENCE
BUDGET WAS \$8.1 BILLION—

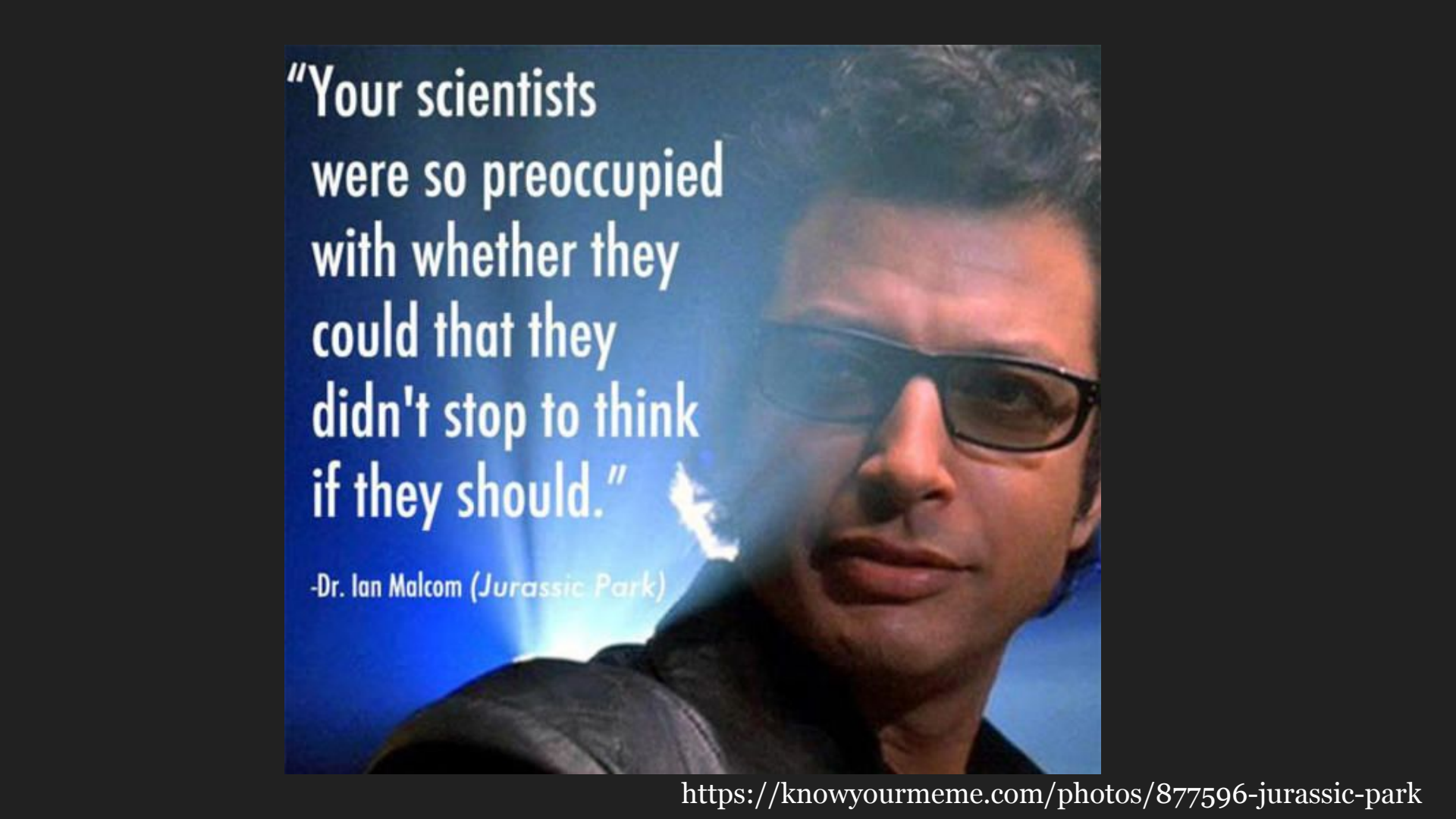
—YET IT WASN'T ENOUGH
TO PICK UP ON THE FACT
THAT NO ONE ELSE HAS
USED THE PREFIX "CYBER—"
FOR LIKE A DECADE?

SHUT UP.



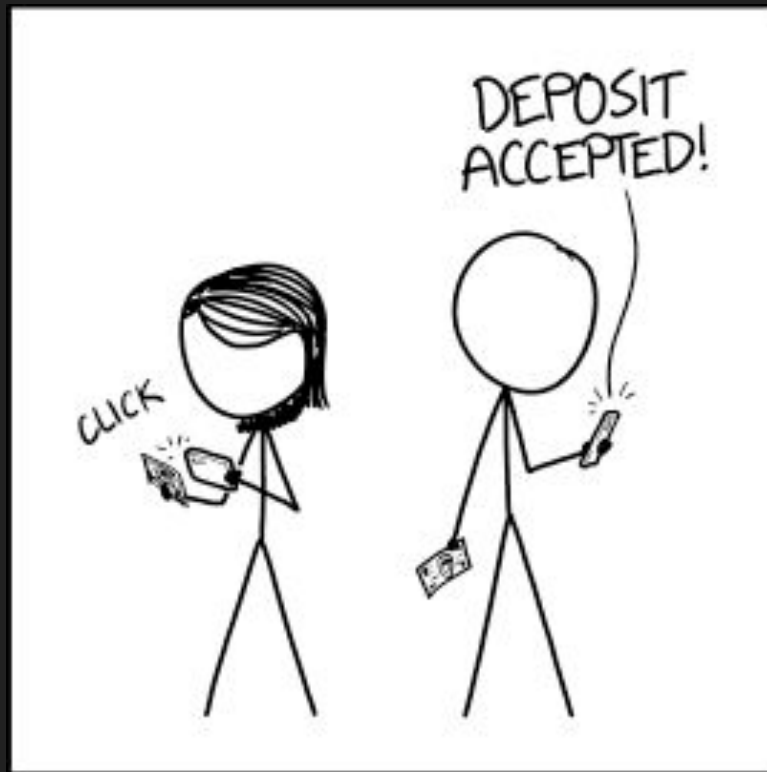
What is **security**?

Security is hard.



"Your scientists
were so preoccupied
with whether they
could that they
didn't stop to think
if they should."

-Dr. Ian Malcom (*Jurassic Park*)



AFTER A LUCRATIVE SIX HOURS FOR US,
OUR BANK REMOVED THE NEW FEATURE
IN THEIR APP THAT LET YOU DEPOSIT
CASH BY TAKING A PICTURE OF IT.

Security is hard. Why?

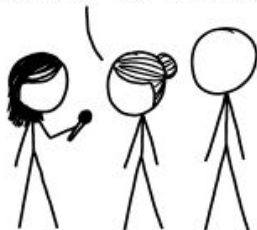
Engineers/developers build without considering security implementations (thinking about the consequences).

“Some people create accounts and then never use them because they forgot their passwords. What if we sent them an email with a link they could click that would log them in without a password?”

“Adversarial mindset”/“red teaming” - think like the person who’s going to attack the product, find and exploit the bugs.

ASKING AIRCRAFT DESIGNERS ABOUT AIRPLANE SAFETY:

NOTHING IS EVER FOOLPROOF, BUT MODERN AIRLINERS ARE INCREDIBLY RESILIENT. FLYING IS THE SAFEST WAY TO TRAVEL.



ASKING BUILDING ENGINEERS ABOUT ELEVATOR SAFETY:

ELEVATORS ARE PROTECTED BY MULTIPLE TRIED-AND-TESTED FAILSAFE MECHANISMS. THEY'RE NEARLY INCAPABLE OF FALLING.



ASKING SOFTWARE ENGINEERS ABOUT COMPUTERIZED VOTING:

THAT'S TERRIFYING.



WAIT, REALLY?

DON'T TRUST VOTING SOFTWARE AND DON'T LISTEN TO ANYONE WHO TELLS YOU IT'S SAFE.

WHY?

I DON'T QUITE KNOW HOW TO PUT THIS, BUT OUR ENTIRE FIELD IS BAD AT WHAT WE DO, AND IF YOU RELY ON US, EVERYONE WILL DIE.



THEY SAY THEY'VE FIXED IT WITH SOMETHING CALLED "BLOCKCHAIN."

AAAAA!!!

WHATEVER THEY SOLD YOU, DON'T TOUCH IT. BURY IT IN THE DESERT. WEAR GLOVES.



Security is hard. Why?

“Our entire field is bad at what we do.”

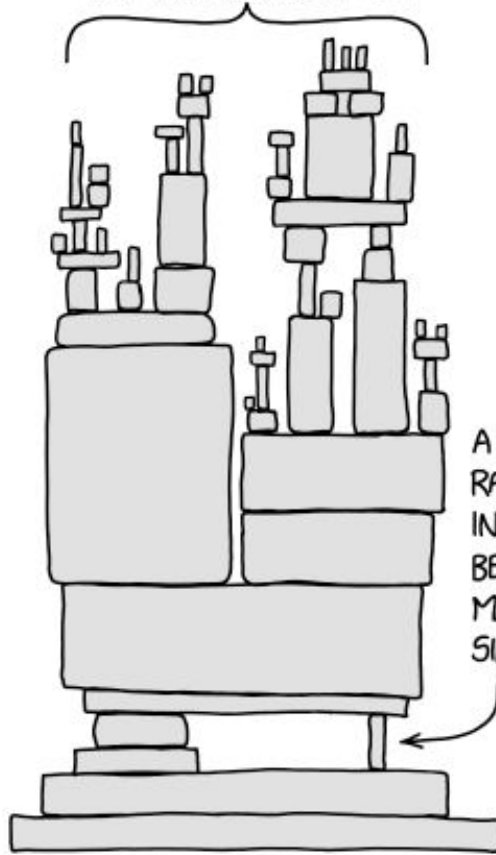
95%* of bugs are programmer error - the computer is just doing what we told it to do!

We have a much greater tolerance for building flawed software than we do real-world products since we can just release an update.

But will ~~users~~ people install updates? Not as often as we'd like.

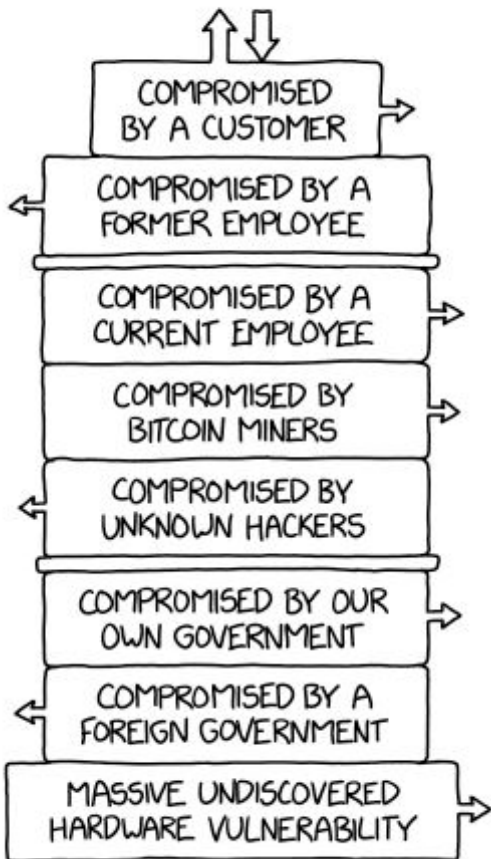
*approximately. Other ~5% is OS, hardware, etc... outside the developer's control.

ALL MODERN DIGITAL
INFRASTRUCTURE



A PROJECT SOME
RANDOM PERSON
IN NEBRASKA HAS
BEEN THANKLESSLY
MAINTAINING
SINCE 2003

THE MODERN TECH STACK



Security is hard. Why?

Most software is not written from scratch - why reinvent the wheel?

Use other people's code ("libraries") to build yours.

Examples: bootstrap & jQuery (frontend web dev), Java's standard library

Hundreds if not thousands of (transitive) libraries!

Over-dependence on other people's code can be dangerous (example: left-pad).

And what if those have vulnerabilities?

A CRYPTO NERD'S
IMAGINATION:

HIS LAPTOP'S ENCRYPTED.
LET'S BUILD A MILLION-DOLLAR
CLUSTER TO CRACK IT.

BLAST! OUR
EVIL PLAN
IS FOILED!

NO GOOD! IT'S
4096-BIT RSA!



WHAT WOULD
ACTUALLY HAPPEN:

HIS LAPTOP'S ENCRYPTED.
DRUG HIM AND HIT HIM WITH
THIS \$5 WRENCH UNTIL
HE TELLS US THE PASSWORD.

GOT IT.



Security is hard. Why?

Humans are bad at adversarial thinking, often easier than a technical exploit.

- Social engineering (IT support, crying baby)
- Phishing (click a malicious link/attachment, fake web page)

Example: Twitter Hack (July 2020) used social engineering (IT support) + phishing (fake page to harvest credentials) + multi-factor authentication bypass.

Usability: increasing security often makes a system more annoying and/or difficult for the average person to use. Example: OS updates (do you *really* want to restart your computer right now?)

The Only Secure System

- One that has its disk drive demagnetized
- Turned off
- Inside a Faraday cage (isolated from electric fields)
- Buried in a hole and
- Sealed in concrete

Adversarial Mindset: Practice

The US Government issues citizens a 9-digit ID (Social Security Number, or SSN) at birth, along with a physical card with their name and SSN. Certain processes (opening a bank account, some standardized tests, MIT background check, etc...) ask or require you to provide all or part of your SSN, while others (getting a driver's license) require you to show your Social Security card.

What are some of the problems with this setup?

Adversarial Mindset: Practice

What are some of the problems with the setup of Social Security (as described)?

- You might lose or destroy your card (e.g. if it gets wet).
- You might forget your SSN.
- A malicious person who knows your SSN can impersonate you.
- Cannot be changed or invalidated - you can't* get a new number, and once you lose your original card, even if you get a new card you can't stop someone using the old one (which - conveniently - has your name on it too!).

Authentication

“How do I prove that I am who I say I am?”

- Something I have - key, ID card
- Something I am - fingerprints, retina scan, DNA test
- Something I know - password, encryption key, combination to lock
- Something I can do - signature, voice recognition, CAPTCHA

Authentication

What can go wrong with each method?

- Something I have - can be lost, stolen, or forged. Once distributed can be difficult to take back.
- Something I am - can be lost or forged (fingerprints), can't be changed.
- Something I know - easy to share, can be forgotten, verifier knows the secret
- Something I can do - what if you lose the ability to do it?

Passwords

Passwords are terrible.

We don't choose good passwords, and we never have.

Worst Passwords of 2014 (25 most common):

- | | | |
|--------------|-------------|--------------|
| 1. 123456 | 11. 1234567 | 21. superman |
| 2. password | 12. monkey | 22. 696969 |
| 3. 12345 | 13. letmein | 23. 123123 |
| 4. 12345678 | 14. abc123 | 24. batman |
| 5. qwerty | 15. 1111111 | 25. trustno1 |
| 6. 123456789 | 16. mustang | |
| 7. 1234 | 17. access | |
| 8. baseball | 18. shadow | |
| 9. dragon | 19. master | |
| 10. football | 20. michael | |

Source: <https://www.teampassword.com/blog/worst-passwords-of-2014>

Worst Passwords of 2018 (25 most common):

- | | | |
|--------------|---------------|---------------|
| 1. 123456 | 11. princess | 21. charlie |
| 2. password | 12. admin | 22. aa123456 |
| 3. 123456789 | 13. welcome | 23. donald |
| 4. 12345678 | 14. 666666 | 24. password1 |
| 5. 12345 | 15. abc123 | 25. qwerty123 |
| 6. 111111 | 16. football | |
| 7. 1234567 | 17. 123123 | |
| 8. sunshine | 18. monkey | |
| 9. qwerty | 19. 654321 | |
| 10. iloveyou | 20. !@#\$%^&* | |

Source: <https://www.teampassword.com/blog/worst-passwords-of-2018>

Other bad passwords?

Almost 10% of people have used at least one of the 25 worst passwords on this year's list, and nearly 3% of people have used the worst password, 123456. (<https://www.teampassword.com/blog/worst-passwords-of-2018>)

Add dictionary words (“cat”, “dog”), dictionary words plus digits (“cat1”, “pencil4”), uppercase first letter (“Cat”, “Dog”), all caps (“CAT”, “DOG”), and all numbers up to 999,999,999 to get much higher percentage (approximately 20%, based on dump of 5 million passwords).

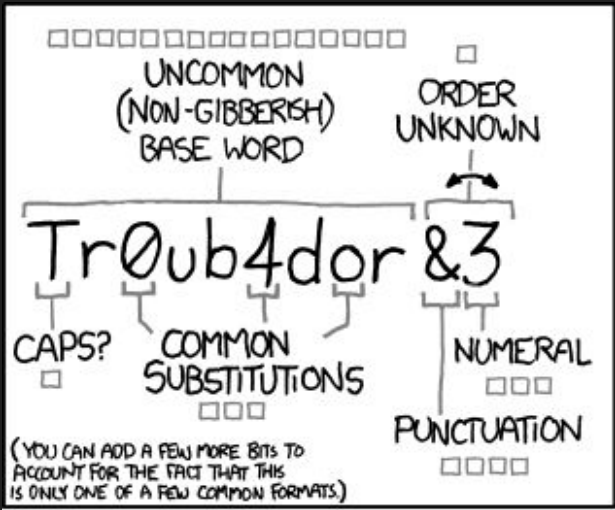
Given an encrypted password, can you figure out what it is? If you know the encryption algorithm, and have an easy way to test potential passwords, trying all of these isn't computationally expensive... about 2^{30} operations, which takes no more than a few hours.

Passwords are terrible.

Security vs. usability: the more complicated a password is, the harder it is to remember.

Especially with site-specific password requirements (lowercase letter, uppercase letter, number, special character, length minimums or maximums).

“Through 20 years of effort, we’ve successfully trained everyone to use passwords that are hard for humans to remember but easy for computers to guess.” - Randall Munroe, XKCD



~28 BITS OF ENTROPY

$2^{28} = 3 \text{ DAYS AT } 1000 \text{ GUESSES/SEC}$

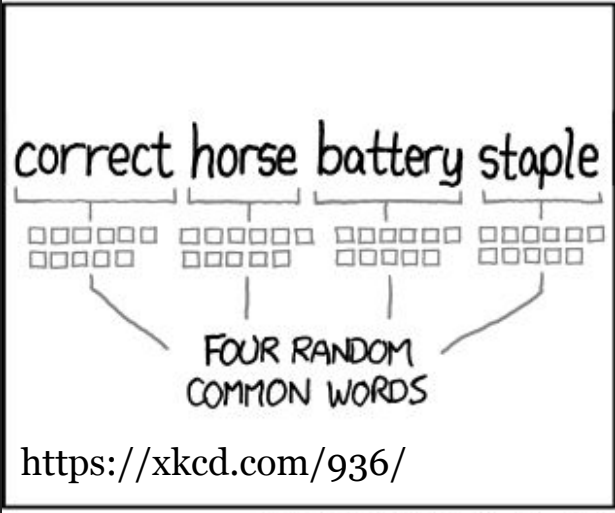
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)

DIFFICULTY TO GUESS: **EASY**

WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE O's WAS A ZERO?

AND THERE WAS SOME SYMBOL...

DIFFICULTY TO REMEMBER: **HARD**



~44 BITS OF ENTROPY

$2^{44} = 550 \text{ YEARS AT } 1000 \text{ GUESSES/SEC}$

DIFFICULTY TO GUESS: **HARD**

THAT'S A BATTERY STAPLE.

CORRECT!

DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT

Is “correct horse battery staple” still sound advice?

While there’s plenty of spirited debate on the Internet (of course) about whether this still holds up (especially as computers get faster / guesses become cheaper), consensus seems to be that picking **six** random words (not four) will lead to a solid password. See <http://www.dicewarepasswords.com/about/>

Math: For n dictionary words, guessing every combination of k requires n^k guesses. 171, 476 words in Oxford English Dictionary, so 4 words becomes $8.6 * 10^{20}$ (800 quintillion), 6 words becomes $2.5 * 10^{31}$ (25 nonillion).

Using a password manager will ensure you don’t have to remember more than a couple of these, and it will automatically generate secure passwords. Some web browsers also provide this functionality.

MySpace hack: 33GB file containing more than 360 million user accounts leaks online



By Jason Murdock

May 31, 2016 15:55 BST Updated 12 hr ago



money.cnn.com/2016/05/19/technology/linkedin-hack/

CNN Money U.S. +

Business Markets Tech Media Personal Finance S

LinkedIn was hacked four years ago, and what initially seemed to be a theft of 6.5 million passwords has actually turned out to be a breach of 117 million passwords.

More than 65m Tumblr emails for sale on the darknet

Tuesday 31 May 2016 05.11 EDT

Company only now discloses scale of hack three years ago - shortly before purchase by Yahoo - as database of passwords is leaked

SONY GOT HACKED HARD: WHAT WE KNOW AND DON'T KNOW SO FAR

Hackers Reveal 4.6 Million Snapchat Usernames and Phone Numbers

Phone numbers, usernames and location data are all part of the database that appeared online

By Denver Nicks @DenverNicks | Jan. 01, 2014

JUN 15, 2015 @ 07:35 PM 25,412 VIEWS

Password Manager LastPass Hacked, Exposing Encrypted Master Passwords

Exclusive — Hacker Steals Over 218 Million Zynga 'Words with Friends' Gamers Data

📅 September 29, 2019 👤 Swati Khandelwal

Popular This Week

CONTROL, WE HAVE FLOWN TO THE USA AND BREACHED THE TARGET'S HOUSE.

THEY WROTE ALL THEIR PASSWORDS IN A BOOK LABELED "PASSWORDS"!

THE FOOL!



HOW PEOPLE THINK HACKING WORKS

HEY LOOK, SOMEONE LEAKED THE EMAILS AND PASSWORDS FROM THE SMASH MOUTH MESSAGE BOARDS.

COOL, LET'S TRY THEM ALL ON VENMO.



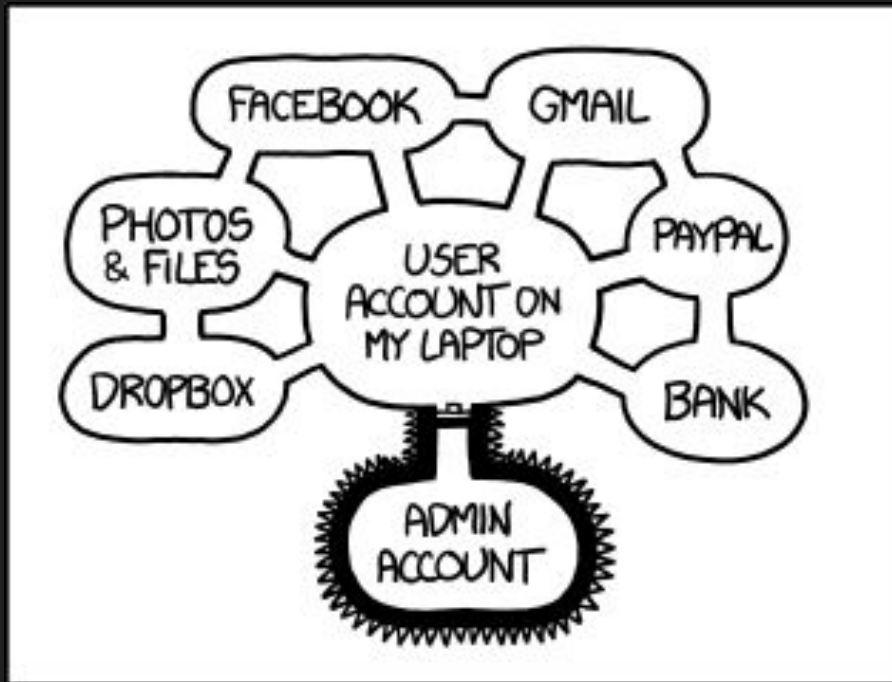
HOW IT ACTUALLY WORKS

Passwords are terrible.

Humans generally have bad “password hygiene” - we’ll come up with a password we think is good and reuse it across multiple sites because it’s a strong password that we can remember.

What happens if one of those sites has a password breach and attackers are able to decrypt/crack the passwords? Individual accounts on other sites (same username/password combo) will often be cracked.

<https://haveibeenpwned.com/> can give a lower bound on which data breaches your account is included in.



IF SOMEONE STEALS MY LAPTOP WHILE I'M LOGGED IN, THEY CAN READ MY EMAIL, TAKE MY MONEY, AND IMPERSONATE ME TO MY FRIENDS, BUT AT LEAST THEY CAN'T INSTALL DRIVERS WITHOUT MY PERMISSION.

Passwords are terrible. So are “security” questions.

Security questions may actually provide a path of less resistance for an attacker focused on a specific account.

Am I more likely to be able to guess your password or your pet’s name? What if it’s something I can just look up?

Introducing a less-secure authentication method lowers the security of the entire system.



<https://twitter.com/mattblaze/status/779117609891409927>

Passwords are terrible. What's the solution?

Multifactor Authentication - require a password **and something else!** Another example of security vs usability trade-off.

- Get a nonce sent to your device (SMS)
- Get a nonce from an app on your device (Duo or Google Authenticator).
- Fingerprint (take advantage of what you are)

How are things vulnerable?

OWASP (Open Web Application Security Project) Top 10 Threats:

1. Injection
2. Broken authentication
3. Sensitive data exposure
4. XML External Entities (XXE)
5. Broken access control
6. Security misconfiguration
7. Cross-site scripting (XSS)
8. Insecure deserialization
9. Using components with known vulnerabilities
10. Insufficient logging and monitoring

How are things vulnerable?

OWASP (Open Web Application Security Project) Top 10 Threats:

1. Injection
- 2. Broken authentication**
3. Sensitive data exposure
4. XML External Entities (XXE)
5. Broken access control
6. Security misconfiguration
7. Cross-site scripting (XSS)
8. Insecure deserialization
- 9. Using components with known vulnerabilities**
10. Insufficient logging and monitoring

How are things vulnerable?

OWASP (Open Web Application Security Project) Top 10 Threats:

1. **Injection**
2. **Broken authentication**
3. Sensitive data exposure
4. XML External Entities (XXE)
5. Broken access control
6. Security misconfiguration
7. **Cross-site scripting (XSS)** (which is really a form of injection)
8. Insecure deserialization
9. **Using components with known vulnerabilities**
10. Insufficient logging and monitoring

Heartbleed (OpenSSL)

Disclosed April 2014

Heartbleed

HEARTBLEED MUST BE THE WORST WEB SECURITY LAPSE EVER.

WORST SO FAR. GIVE US TIME.



I MEAN, THIS BUG ISN'T JUST BROKEN ENCRYPTION.

IT LETS WEBSITE VISITORS MAKE A SERVER DISPENSE RANDOM MEMORY CONTENTS.



IT'S NOT JUST KEYS. IT'S TRAFFIC DATA. EMAILS. PASSWORDS. EROTIC FANFICTION.

IS *EVERYTHING* COMPROMISED?



WELL, THE ATTACK IS LIMITED TO DATA STORED IN COMPUTER MEMORY.

SO PAPER IS SAFE. AND CLAY TABLETS. OUR IMAGINATIONS, TOO. SEE, WE'LL BE FINE.



What is SSL?

Secure Sockets Layer

A **layer** (level of network communication)

For **sockets** (client/server communication)

To communicate **securely**.

“SSL creates an encrypted connection between a web server and a web browser allowing for private information to be transmitted without the problems of eavesdropping, data tampering, or message forgery.” -

<https://www.sslshopper.com/what-is-ssl.html>

What is the Heartbleed bug?

In OpenSSL for two years before it was discovered.

Allowed clients to abuse the **heartbeat** between client and server (making sure connection is still alive) to get more information about the server.

Technical explanation: “This serious flaw (CVE-2014-0160) is a missing bounds check before a `memcpy()` call that uses non-sanitized user input as the length parameter. An attacker can trick OpenSSL into allocating a 64KB buffer, copy more bytes than is necessary into the buffer, send that buffer back, and thus leak the contents of the victim's memory, 64KB at a time.”

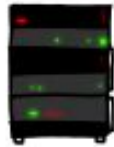
If you understood that, great. If not, ...

HOW THE HEARTBLEED BUG WORKS:

SERVER, ARE YOU STILL THERE?
IF SO, REPLY "POTATO" (6 LETTERS).



...this pages about "books". User since page
secure connection using key "4538538374224"
User Meg wants these 6 letters: POTATO. User
da wants pages about "irl games". Unlocking
secure records with master key 5130985733435
...this pages about "books". User since page



POTATO



...this pages about "books". User since page
secure connection using key "4538538374224"
User Meg wants these 6 letters: **POTATO**. User
da wants pages about "irl games". Unlocking
secure records with master key 5130985733435
...this pages about "books". User since page

SERVER, ARE YOU STILL THERE?
IF SO, REPLY "BIRD" (4 LETTERS).



ser Olivia from London wants pages about "na
bees in car why". Note: Files for IP 375.381.
83.17 are in /tmp/files-3843. User Meg wants
these 4 letters: BIRD. There are currently 346
connections open. User Brendan uploaded the file
selfie.jpg (contents: 234ba962e2c0eb9ff89b43b4ff8



HMM...



ser Olivia from London wants pages about "na
bees in car why". Note: Files for IP 375.381.
83.17 are in /tmp/files-3843. User Meg wants
these 4 letters: **BIRD**. There are currently 346
connections open. User Brendan uploaded the file
selfie.jpg (contents: 234ba962e2c0eb9ff89b43b4ff8

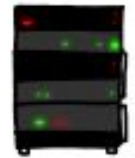
BIRD



SERVER, ARE YOU STILL THERE?
IF SO, REPLY "HAT" (500 LETTERS).

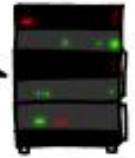


a connection. Jake requested pictures of deer.
User Meg wants these 500 letters: HAT. Lucas
requests the "missed connections" page. Eve
(administrator) wants to set server's master
key to "14835038534". Isabel wants pages about
snakes but not too long". User Karen wants to
change account password to "CoffeeSt". User



HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about "snakes but not too long". User Karen wants to change account password to "CoffeeSt". User Isabel requests pages

a connection. Jake requested pictures of deer.
User Meg wants these 500 letters: HAT. Lucas
requests the "missed connections" page. Eve
(administrator) wants to set server's master
key to "14835038534". Isabel wants pages about
snakes but not too long". User Karen wants to
change account password to "CoffeeSt". User



Why is it still relevant today?

Similar to common classes of issues in native code (C and similar languages with manual memory management).

Not checking that two implicitly related values (length of string, amount of memory to read) match, which often leads to Buffer Overflow.

Buffer overflow: allocate some amount of space, write more data than you have space allocated for. The rest of the data goes elsewhere in memory. Carefully craft that data to look like code, and you can convince the computer to run it, which gives you control over the code being executed.

Injection Attacks

SQL Injection, XSS

Software is just code. What is code?

Two main types of programming languages: compiled and interpreted.

Interpreted language: a program (the “interpreter”) reads a human-readable text-like file containing code and executes it.

Compiled language: a program (the “compiler”) reads a human-readable text-like file containing code and converts it to a machine-readable file containing “machine code”, which is then run by another program.

`eval` is evil: take a string containing code in that language and execute it.

What is an injection attack?

Injection attacks happen when we take data (usually user-provided data) and treat it like code provided by the programmer.

This data will often look like a fragment of code. When we run it, we've "injected" that code into the running program, hence the name.

What is SQL?

Structured Query Language for accessing tabular relational databases.

id	title	author	body
1	Databases	John	Message1
2	Technology	Joe	Message2
3	Security	Julia	Message3

“SELECT * FROM table WHERE id = 2” will select

2	Technology	Joe	Message2
---	------------	-----	----------

“SELECT author FROM table WHERE title = ‘Security’” gives “Julia”.

SQL Example

Suppose:

```
usernm = mumble() //get input from user
```

```
userpass = mumble() //get password from user
```

```
SELECT * FROM user WHERE name = `usernm` AND password =  
`userpass`;
```

SQL Example

Suppose:

```
usernm = mumble() //get input from user
```

```
userpass = mumble() //get password from user
```

```
SELECT * FROM user WHERE name = `usernm` AND password =  
`userpass` ;
```

What happens if usernm is “zach”?

```
SELECT * FROM user WHERE name = `zach` AND password =  
`123456` ;
```

Injecting into SQL

Suppose:

```
usernm = mumble() //get input from user
userpass = mumble() //get password from user
SELECT * FROM user WHERE name = `usernm` AND password =
`userpass` ;
```

Some SQL syntax:

-- begins a comment - anything after it on a line is ignored.

What happens if usernm is “admin’ ; --”

Injecting into SQL

What happens if `username` is “`admin' ; --`”?

```
SELECT * FROM user WHERE name = 'admin' ; --' AND password =  
'123456'
```

Returns any row where name is “admin” - which may include admin’s password.

Injecting into SQL

What happens if usernm is “admin' ;--”?

```
SELECT * FROM user WHERE name = 'admin' ;--' AND password =  
'123456'
```

Alternative: What if you don't know a valid username?

usern is “' OR 1=1 ;--”

```
SELECT * FROM user WHERE name = '' OR 1=1 ;--' AND password =  
'123456'
```

Returns every row in the table...

Other SQL Commands

```
INSERT INTO table_name VALUES (value_1, value_2);
```

```
DELETE FROM table_name WHERE condition;
```

```
UPDATE table_name SET column_name = column_value;
```

```
DROP TABLE table_name;
```

SQL Injection Attacks

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.



OH, DEAR - DID HE BREAK SOMETHING?
IN A WAY--



DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?



OH, YES. LITTLE BOBBY TABLES, WE CALL HIM.

WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.



AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Preventing SQL Injection

Sanitize inputs! We have a few options:

1. Block certain words, like 'DROP TABLE' - if you see it, remove it. Can't possibly go wro... 'DR**DROP** TABLEOP TABLE' ... oops.
2. PHP's `mysql_escape_string()` and `mysql_real_escape_string()`
3. Prepared Statements: instead of treating data (supplied by user) as code (via substitution), treat it as data (arguments) to be used by code (function)

Other types of injection: XSS

Cross-Site Scripting - inject Javascript (inside `<script>` tags) into raw HTML. The browser executes anything inside these tags whenever it finds it on the page... including malicious redirection to other site, sending data to certain URL, retweeting a tweet...

Solution: sanitize inputs (but don't do it yourself! You'll probably miss something).

Self-Retweeting Tweet

Exploited a vulnerability in Tweetdeck, a platform for viewing Tweets. All a user would see is the heart at the end... but they would have retweeted it as soon as they saw it. Regular Twitter users saw the full thing.



The screenshot shows a tweet from user *andy (@derGeruhn) on the Tweetdeck interface. The tweet content is a JavaScript payload: `<script class="xss">$($('.xss').parents().eq(1).find('a').eq(1).click());$('[data-action=retweet]').click();alert('XSS in Tweetdeck')</script>` followed by a red heart icon. The interface shows the tweet has 39,027 retweets and 2,531 favorites. The tweet was posted at 5:36 PM on June 11, 2014. A 'Blocked' button is visible in the top right corner of the tweet card.

*andy
@derGeruhn

Blocked

`<script
class="xss">$($('.xss').parents().eq(1).find('a')
.eq(1).click());$('[data-
action=retweet]').click();alert('XSS in
Tweetdeck')</script>` ❤️

Reply Retweet Favorite More

RETWEETS 39,027 FAVORITES 2,531

5:36 PM - 11 Jun 2014

Security is hard.

But we're getting better at it.

We know how to fix these things.

Or at least how to make them more difficult for attackers.

Passwords: password managers, multi-factor authentication, password generators.

Buffer overflow: stack canaries, non-executable stack.

SQL injection: prepared statements

XSS: Content Security Policy, Trusted Types

We also know how to find issues.

Password cracking attempts: “what would happen if someone got their hands on our encrypted passwords?”

Static analysis: detect untrusted data flowing into somewhere it could be interpreted as code.

Runtime analysis: fuzzing

So it's not all doom and gloom.

Questions?