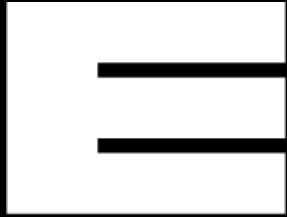


# An Information Theoretic Model of Privacy and Security Metrics

- or -

How I learned to stop worrying about password meters  
and love the dice

Bill Budington  
 @legind



**ELECTRONIC  
FRONTIER  
FOUNDATION**



# Who are we?

- Digital rights nonprofit
- Technologists, Lawyers, Activists
- Fight for Encryption, Privacy, and Security on the Internet
- <https://eff.org/>

# Who am I?



- **Senior Staff Technologist  
at EFF's Threat Lab**
- **Digital Security Trainer**
- **Privacy & Security Auditor**
- **HTTPS Everywhere,  
Cover Your Tracks**

# Cover Your Tracks

<https://coveryourtracks.eff.org/>

- Formerly called “Panopticlick”
- Uses different characteristics of the browser (web headers, JS derived properties)
- Combines these characteristics into unique “Fingerprint” of your browser
- Compares browser fingerprint against others we’ve recently seen

# COVER YOUR TRACKS

A large, stylized paw print graphic in a dark green color, serving as a background for the main title.

See how trackers view your browser

Test your browser to see how well you are protected from tracking and fingerprinting:

[Learn](#)

[About](#)

**STOP  
ANIMATION**

## HOW CAN I USE MY RESULTS TO BE MORE ANONYMOUS?

Knowing how identifiable you are, or whether you are blocking trackers, can help you take steps to better protect your privacy. Browser add-ons or protection mechanisms built into the browser can help. Even so, the sneakiest trackers have ways around even the strongest security. We recommend you use a tracker blocker like [Privacy Badger](#), or a browser with built-in fingerprinting protection.

## WHAT IS A BIT OF INFORMATION?

A “bit” is a basic unit of information for computers. The bit represents a logical state with one of two possible values, often represented as “1” or “0”, for example. In your results from Cover Your Tracks, some metrics may be listed as “1” or “0”, or “true” or “false”, indicating whether a setting is enabled or disabled. While each individual metric’s details may seem like a small amount of information, when combined with your browser’s other metrics, they can uniquely identify your browser. Your results are measured in “bits of identifying information,” which is a combined summary of all these metrics.

LEARN MORE

*Note: because tracking techniques are complex, subtle, and constantly evolving, Cover Your Tracks does not measure all forms of tracking and protection.*

## Your Results

Your browser fingerprint **appears to be unique** among the 282,012 tested in the past 45 days.

Currently, we estimate that your browser has a fingerprint that conveys **at least 18.11 bits of identifying information**.

The measurements we used to obtain this result are listed below. You can [read more about our methodology, statistical results, and some defenses against fingerprinting here](#).

## Detailed Results

Here’s some more granular information we gathered about your browser. Your report includes examples of several different kinds of metrics:

### WEB HEADERS

# Fingerprint Metrics

## SYSTEM FONTS

Arial, Arial Narrow, Bitstream Vera Sans Mono, Calibri, Cambria, Courier New, Times New Roman (via javascript)

## WHAT IS THIS?

To determine your system fonts, tracking sites commonly display some text in an **HTML `<span>` tag**. Trackers then rapidly change the style for that span, rendering it in hundreds or thousands of known fonts. For each of these fonts, the site determines whether the width of the span has changed from the default width when rendered in that particular font. If it has, the tracker knows that font is installed.

## HOW IS THIS USED IN YOUR FINGERPRINT?

The list of fonts you have installed on your machine is generally consistent and linked to a particular operating system. If you install just one font which is unusual for your particular browser, this can be a highly identifying metric.

**Bits of identifying information: 8.45**

**One in  $x$  browsers have this value: 349.46**

# Fingerprinting

Rijksoverheid Sans Web Text Regular

ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789

abcdefghijklmnopqrstuvwxyz éèüñçô?!.,:-) 0123456789

Rijksoverheid Sans Web Text Italic

ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789

abcdefghijklmnopqrstuvwxyz éèüñçô?!.,:-) 0123456789

Rijksoverheid Sans Web Text Bold

ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789

abcdefghijklmnopqrstuvwxyz éèüñçô?!.,:-) 0123456789

```

1 GET /
2 Host: commons.wikimedia.org
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_
64; rv:40.0) Gecko/20100101 Firefox/40.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 Referer: https://commons.wikimedia.org/wiki/Cate
gory:Fonts
9 Cookie: WMF-Last-Access=06-Dec-2015; GeoIP=US:CA
:Oakland:37.83:-122.22:v4; CP=H2; commonswikimwu
ser-sessionId=a8f3987a024fdea3
10 Connection: keep-alive
11 Cache-Control: max-age=0
  
```



# Calculating Entropy

***Entropy:*** a mathematical quantity which allows us to measure how close a fact comes to revealing a person's identity uniquely.

***Surprisal:*** a quantity measuring how unexpected a new piece of information is, which allows us to recalculate entropy.

# Calculating Entropy

$$\Delta S = \log_{-2}(\text{Pr}(X=x))$$

Starsign  $\Delta S = \log_{-2}(\text{Pr}(\text{Starsign}=\text{Capricorn})) = \log_{-2}(1/12) = 3.58$  bits

Birthday  $\Delta S = \log_{-2}(\text{Pr}(\text{DOB}=\text{Jan 2})) = \log_{-2}(1/365) = 8.51$  bits

# Possible State Bits $\neq$ Identifying Bits

Cookies being disabled is very rare.

“True” or “False”  $\rightarrow$  1 bit of stored information

“I have cookies enabled”  $\rightarrow$  0.13 bits of  
identifying information

Physical analogue: green eyes



1 **D**ate: Mon, 24 Feb 2020 01:28:51 -0700

2 To: panopticlick@eff.org

3 Subject: User Agents

4

5 Supposing I wanted to make my browser blend in as much as possible, what is  
6 a good source of the most common User Agents? I'm having trouble finding  
7 good sources thus far.

# Customization Not Recommended!

- **If it looks like Safari on iOS, acts like Safari on iOS, but says it's Chrome on Windows 10...**
- ***More unique than Safari on iOS announcing itself as such***

# Customization Not Recommended!



**How do you do, fellow kids?**

# In order for browser fingerprinting to work...

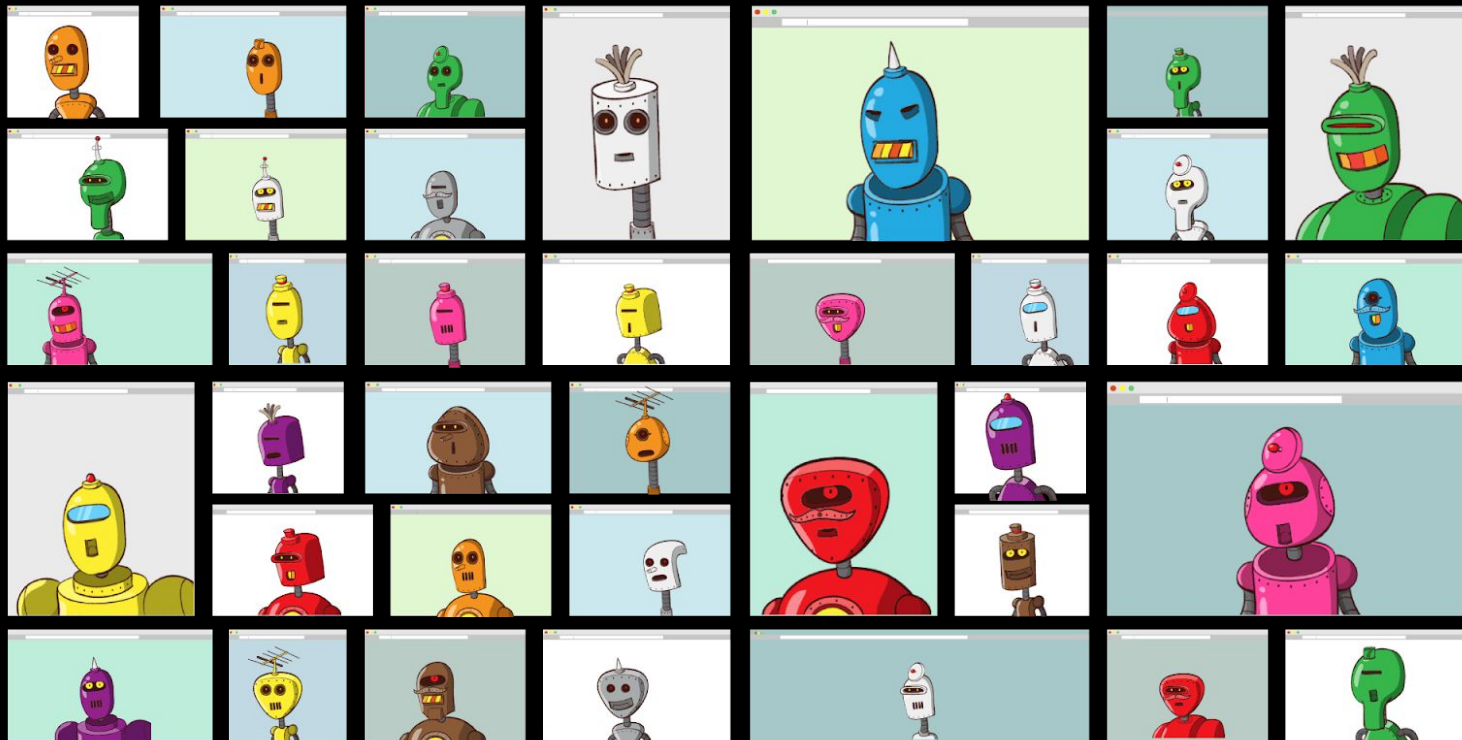
- **Unique enough to be tracking individual browsers**
- **Constant enough to be a stable identifier**

# Tor Browser anti-FP

**Goal is to make every TB instance  
look exactly the same (mod OS)**

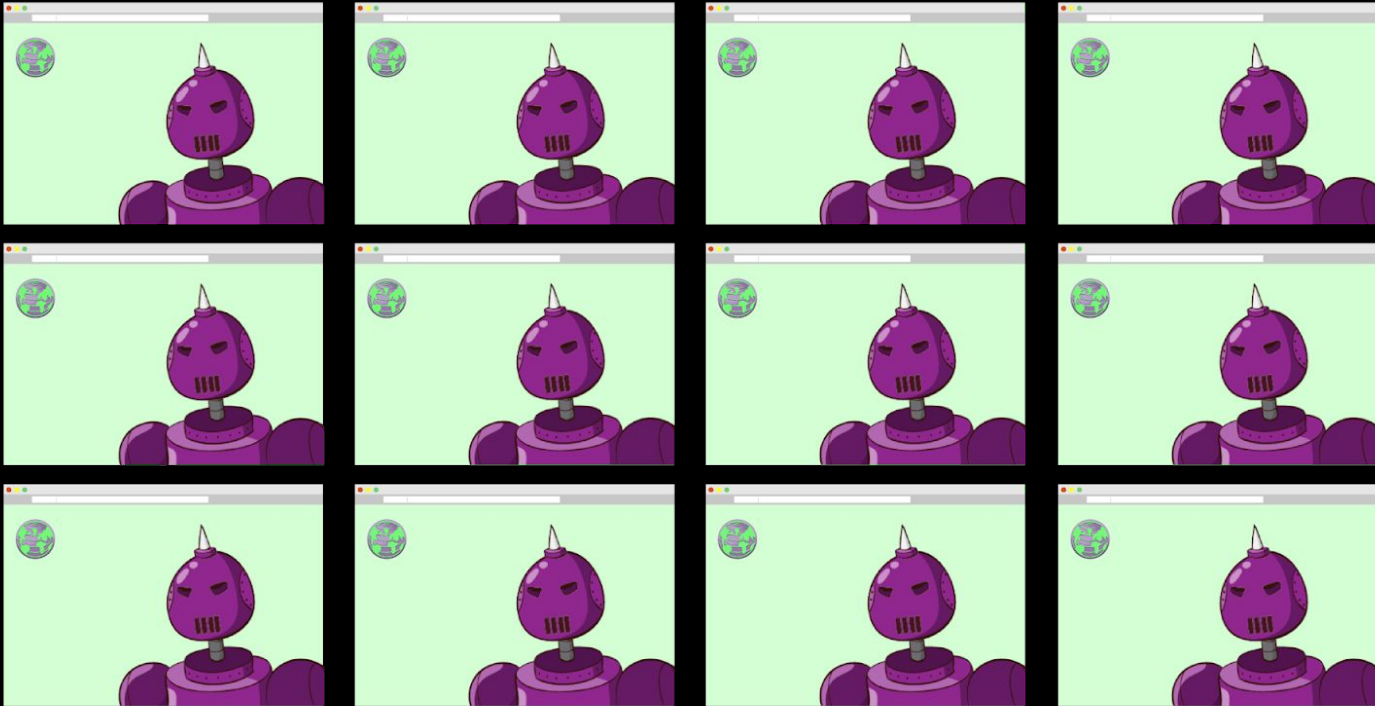


# Tor Browser anti-FP



CREDIT FOR ROBOT IMAGES TO ROBOHASH.ORG

# Tor Browser anti-FP





## FPRandom: Randomizing core browser objects to break advanced device fingerprinting techniques

Pierre Laperdrix, Benoit Baudry, Vikas Mishra

► **To cite this version:**

Pierre Laperdrix, Benoit Baudry, Vikas Mishra. FPRandom: Randomizing core browser objects to break advanced device fingerprinting techniques. ESSoS 2017 - 9th International Symposium on Engineering Secure Software and Systems, Jul 2017, Bonn, Germany. pp.17. hal-01527580

# Brave anti-FP

**Certain metrics requested by 3rd parties (AudioContext, Canvas Hash, WebGL hash, list of plugins, CPU concurrency) are randomized. Randomization seed: first party domain**

# Brave anti-FP



loaded on **example1.com** →



loaded on **example1.com** →




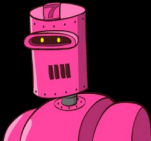

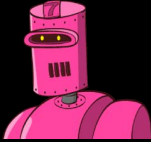




loaded on **example2.com** →



loaded on **example2.com** →



# Brave anti-FP

-  loaded on **example1.com** → 
-  loaded on **example1.com** → 
-  loaded on **example2.com** → 
-  loaded on **example2.com** → 

# Brave anti-FP



# View of Brave from Trackers

➔ **Dumb Trackers:**

**“Check out this one weird trick -  
Trackers HATE it!”**

➔ **Smart Trackers:**

**Able to determine randomization is used  
and use that fact as a fingerprinting metric  
itself (still less useful)**



# Goal:

**Reduce the amount of usable information trackers can gather.**

# Behavioral Fingerprinting

- **Separate from the browser, what behaviors can be observed that, in combination, identify particular users?**
- **What can be done in the browser to mitigate the effectiveness of using these behaviors to fingerprint users?**

# Behavioral Fingerprinting

- **Highlighting text while reading an article**
- **Typing speed and cadence**
- **Cursor movement**
- **Scroll patterns**
- **Switching of tabs**

# Calculating Entropy

$$\Delta S = \log_{-2}(\text{Pr}(X=x))$$

let event = User highlights text while reading article

$$\Delta S = \log_{-2}(\text{Pr}(\text{event}))$$

# Entropy Considerations

Entropy calculation is limited by *predefined parameters* and when we consider user behavior, misses a lot of valuable information.

e.g. User highlights only first word of a paragraph, or only in the middle of a text block, or only in the morning after they drink coffee, etc etc

# Entropy Considerations

**Unlike browser characteristics, behavior of users is not confined to discrete states. It is *open-ended* and *complex*.**

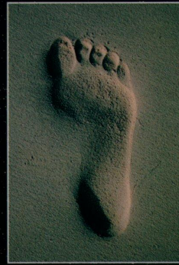
**Open-ended: could be a pattern not easily recognized.**

STUART KAUFFMAN

AT HOME  
IN THE



UNIVERSE



○ The Search for  
the Laws of  
Self-Organization  
and Complexity

**Chaotic systems are subject to massive perturbations from small changes.**

**Ordered systems are too rigid to exhibit interesting characteristics.**

**Complexity emerges at the edge of chaotic systems, between chaotic and ordered regimes. Adaptable & resilient.**

# Human Behaviors are Complex

- **Allows us to adapt to and navigate our environment**
  - **Physical**
  - **Social / Group**
  - **Societal**



# Passwords & Human Neurology

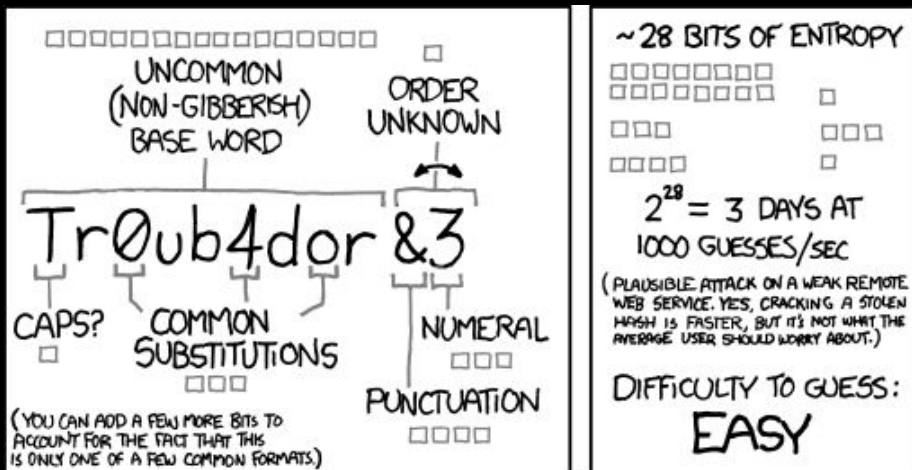
- “Complexity” in this sense does not lend itself to good password choices!
- For cryptographic application (where server-side rate limiting is not an option) actual, strong randomness is *necessary*

## Reduce complexity

NIST recommends **minimizing** password **complexity requirements**, like the necessary inclusion of upper case letters, symbols, and numbers. As with frequent password change policies, these requirements can result in passwords that decrease usability and hamper employee efficiency. Reducing password complexity can be another great step on the road to better security practices that employees find easier to manage.<sup>1</sup>

# Passwords & Human Neurology

Not only are password choices cryptographically weak, but they also exhibit extreme bias (read: patterns)



# Password Meters on Human Input

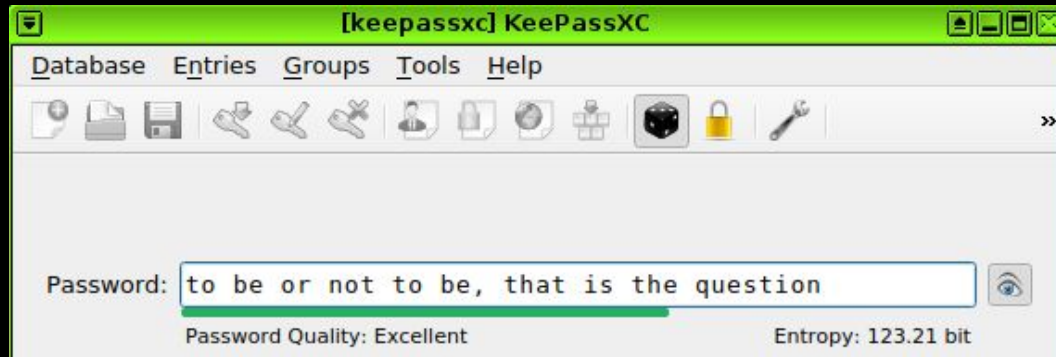


.....|

**Password must:**

- Have at least one letter
- **Have at least one capital letter**
- **Have at least one number**
- Not contain more than 3 consecutive identical characters
- Not be the same as the account name
- Be at least 8 characters

# Password Meters on Human Input



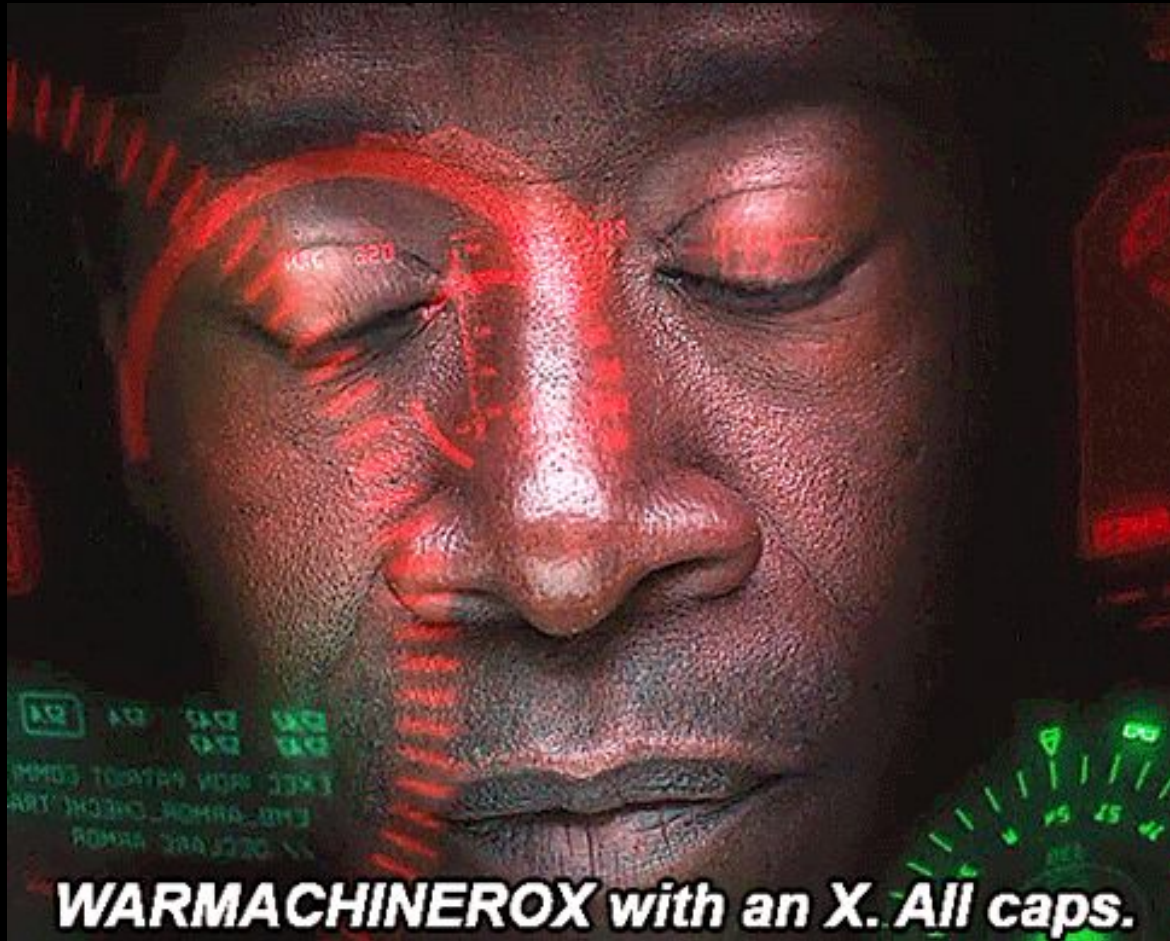
# Entropy meters

## 1. Apply intrinsic criteria to an extrinsic data set

Simple example: the word “question” measured as  $\log_2(26^8)$  bits

## 2. Do not even attempt to determine the source of entropy

**EFF**



**WARMACHINEROX with an X. All caps.**

# Fundamental Principle

**Any entropy calculation run on an open system (e.g. user input data) will fail because it cannot accurately model the source of entropy and data set available to that system.**



# Practical Implication

- **Any mismatch between pattern recognition of meter (if it even has one) and pattern generation of attacker leads to enormous advantage of attacker.**
- **This can include any personalized knowledge of the target.**

# Practical Implication

## Personalized wordlists - extremely common offsec practice

```
[*]-[root@parrot]-[~/home/parrot]
#cupp
```

```
cupp.py!
```

```
# Common
# User
# Passwords
# Profiler

[ Muris Kurgas | j0rgan@remote-exploit.org ]
```

```
[ Options ]
```

```
-h You are looking at it baby! :)
    For more help take a look in docs/README
    Global configuration file is cupp.cfg

-i Interactive questions for user password profiling

-w Use this option to improve existing dictionary,
    or WyD.pl output to make some pwnsauce

-l Download huge wordlists from repository

-a Parse default usernames and passwords directly from Alecto DB.
    Project Alecto uses purified databases of Phenoelit and CIRT
    which where merged and enhanced.

-v Version of the program
```

```
[+] Insert the informations about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)
```

```
> First Name: James
> Surname: Franco
> Nickname: J-dizzle
> Birthdate (DDMMYYYY): 21011991

> Partners) name: Jess
> Partners) nickname: Turner
> Partners) birthdate (DDMMYYYY): 01041982

> Child's name: Billy
> Child's nickname: Franco
> Child's birthdate (DDMMYYYY): 02032018

> Pet's name: Rabby
> Company name: Top Business Direct
```

Home » Security Bloggers Network » 8 Scary Statistics about the Password Reuse Problem



## 8 Scary Statistics about the Password Reuse Problem



by Kim Jacobson on April 30, 2020

As we rapidly move everything online in response to the global pandemic, this has put passwords front and center again. With the latest [Marriott breach](#), it's like groundhog day when it comes to passwords with both organizations and users failing to take the necessary measures to step up their password hygiene.

Passwords remain a weak link and are the source of many cybersecurity vulnerabilities. From companies failing to implement technology detecting and preventing the use of compromised credentials to users having one core password for every single account, we seem oblivious to the risks.

Here are some staggering statistics that show the magnitude of the password reuse problem.

1. A Google survey found that at least [65% of people](#) reuse passwords across multiple, if not all, sites.
2. Another recent survey found that [91% of respondents](#) claim to understand the risks of reusing passwords across multiple accounts, but 59% admitted to doing it anyway.
3. Microsoft recently announced that a staggering [44 million accounts](#) were vulnerable to account takeover due to compromised or stolen passwords.
4. The average person reuses each password as [many as 14 times](#).
5. [72% of individuals](#) reuse passwords in their personal life while nearly half (49%) of employees simply change or add a digit or character to their password when updating their company password every 90 days. These [forced resets](#) are an ineffective tactic.
6. And it is not just personal accounts. [73% of users](#) duplicate their passwords in both their personal and work accounts.
7. Security.org found that [76% of millennials](#) recycle their passwords.
8. This is why compromised passwords are responsible for [81% of hacking-related breaches](#), according to the Verizon Data Breach Investigations Report.

# Why leave the choice of good random passwords to chance?

- **For orgs: generate random passphrases.**
- **For site logins, a using a good PBKDF (scrypt) in case of DB compromise and server-side rate limiting *may* be sufficient. Still won't help against password reuse, but won't frustrate users. If you don't want to frustrate users, mandating U2F also won't work. Or, highly encourage using generated passphrase, but give a fallback.**

## EFF Dice-Generated Passphrases

Create strong passphrases with EFF's new random number generators! This page includes information about passwords, different wordlists, and EFF's suggested method for passphrase generation. Use the directions below with any set of dice.

And now, a message from internationally renowned security technologist, author, and EFF Board Member [Bruce Schneier](#):



[Privacy info](#). This embed will serve content from [archive.org](#)

### Directions

We'll walk you through how to use [EFF's Long Wordlist \[.txt\]](#) to generate a passphrase. For most applications, we suggest making a six-word passphrase.



## EFF'S **NEW WORDLISTS** FOR RANDOM PASSPHRASES

56362	stinking
56363	stinky
56364	stipend
56365	stipulate
56366	stir
56411	stitch
56412	stock
56413	stoic
56414	stoke
56415	stole
56416	stomp
56421	stonewall
56422	stoneware
56423	stonework
56424	stoning
56425	stony



stonks

# Advantages of Diceware

- **Can be guaranteed secure\***
- **Memorable**
- **Kind of fun**

# Disadvantages of Diceware

- **Pretty anglocentric, not available in many languages**
- **Maybe not fun**



# SecureDrop Sources

The Washington Post



## Welcome

Please either write this codename down and keep it in a safe place, or memorize it.

This codename is what you will use in future visits to receive messages from our team in response to what you submit on the next screen.



wrangle nuptials hurricane negligent barbell majestic curling



Because we do not track users of our **SecureDrop** service, in future visits, using this codename will be the only way we have to communicate with you should we have questions or are interested in additional information. Unlike passwords, there is no way to retrieve a lost codename.

SUBMIT DOCUMENTS

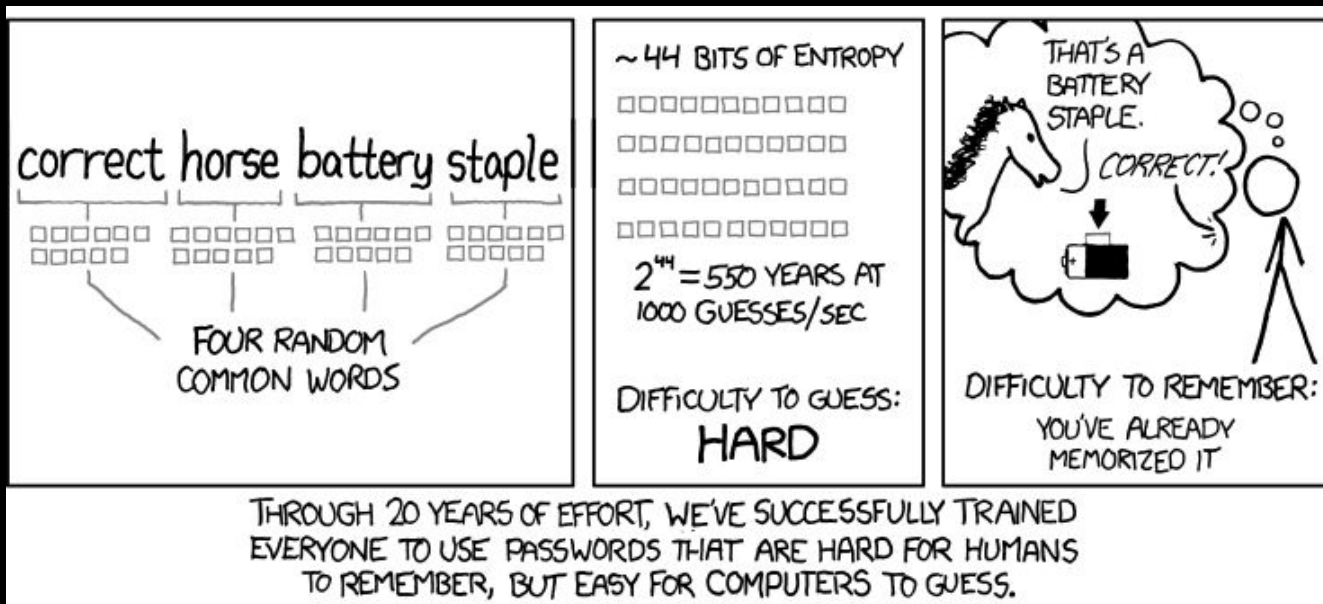
# Diceware Entropy

$$\text{bits} = \log_2(\text{words in wordlist}) * \# \text{ of words}$$

$$\text{bits} = 12.92 * \# \text{ of words}$$

$$\text{bits} = 12.92 * 5 = 64.6$$

# But are they memorable?



# Random Diceware Trial

## 5 words

baritone repeater mower unzip pretext  
viewless undead purify habitable theology  
jargon context woof acquaint bruising  
giblet issuing cattail handgrip immature

# Discarding passphrases is a-OK\*

**\*Depending on how many you discard on average**

**Only like 1 in 2? Lose one bit**

**64.6 → 63.6**

**1 in 4? Lose two bits**

**64.6 → 62.6**

**1 in 8? Lose three bits**

**64.6 → 61.6**

# In conclusion...

- **Password meters are free as in beer, and they also suck as in free beer...**
- **Use Diceware to generate your master passphrase if you're a user.**
- **Generate user password/phrase (using Diceware or other means) if you are a security engineer and care about your users' accounts.**

**EFF**



**Thank You!**



**Questions?**

**Bill Budington**

 **@legind**