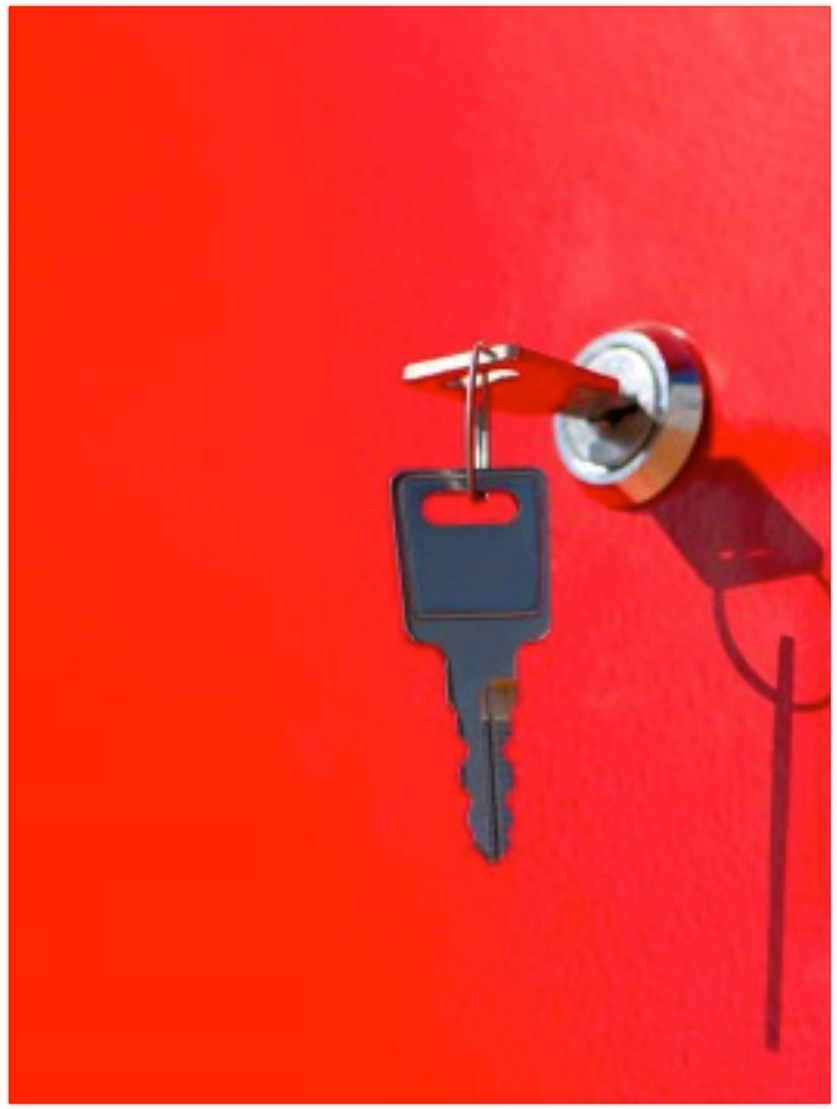
Misusing Passwords

Using Weak Passwords
Password Iteration
Default Passwords
Rainbow Tables Explained
Too Much Information



Password Risks

- Attacker finds the password
- Attacker gains access without the password

Getting Passwords

- Phishing
- Key loggers
- Using other sites
- Default passwords
- Listening
- Guessing (Brute Force)

Password Phishing

Using Other Sites

- Attacker hacks into crappysite.com
- Downloads user details and passwords from crappysite.com
- Surprise ! it’s also the gmail password

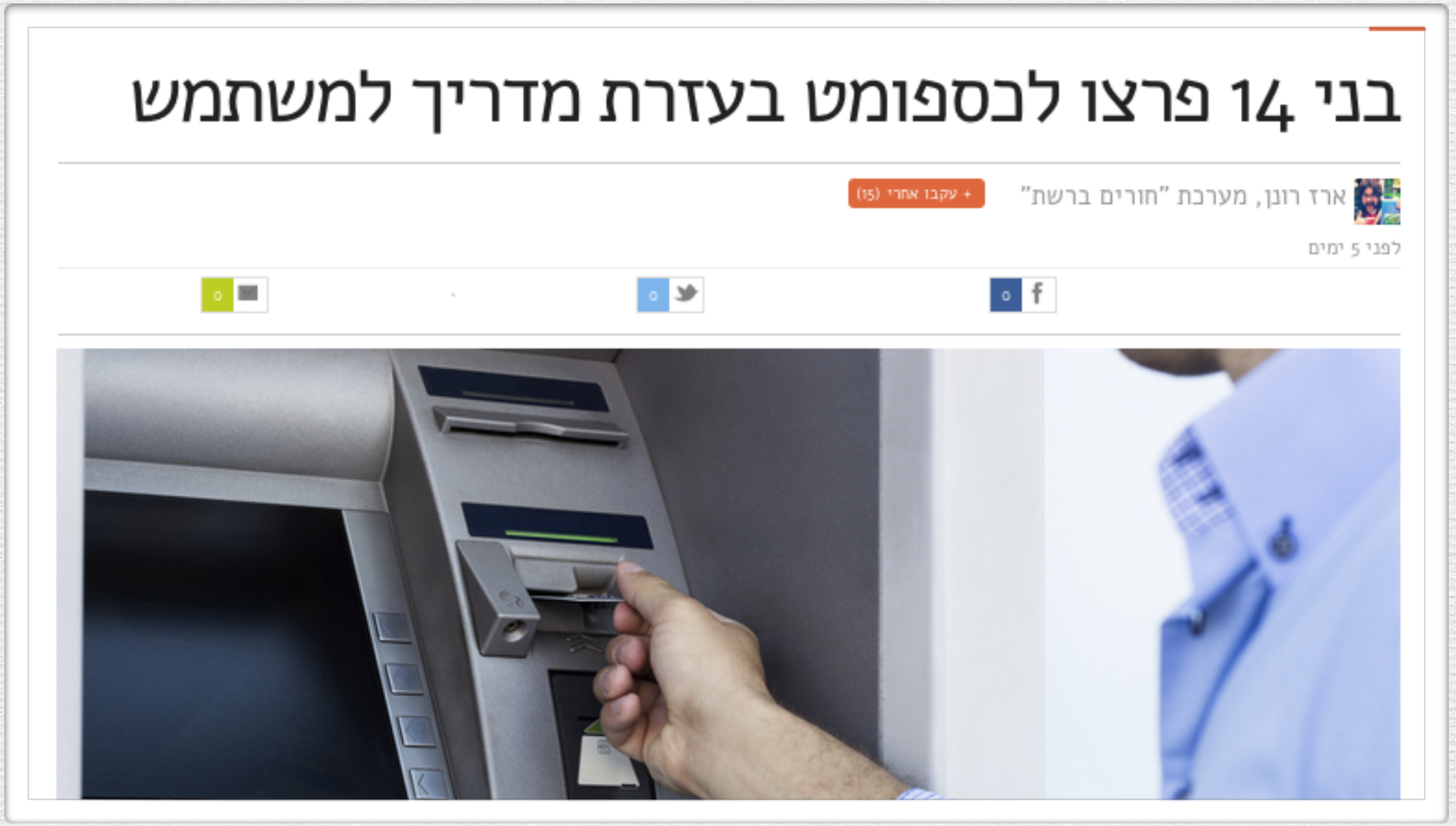
Password Iteration

- crappysite.com told all users their password was compromised and they need to change
- mypassword -> mypassword1

Default Passwords

- Usually for hardware
- Remember to change them

Default Passwords



בני 14 פרצו לכספומט בעזרת מדריך למשתמש

ארז רונן, מערכת "חורים ברשת"

+ עקבו אחרי (15)

לפני 5 ימים

Info Leak

```
if ( invalid_username ) {
   return 0;
} else if ( invalid_password ) {
   // prevent brute-force attacks
   sleep(5);
   return 0;
}
```

Info Leak

```
if ( invalid_username ) {
   return 0;
} else if ( invalid_password ) {
   // prevent brute-force attacks
   sleep(5);
   return 0;
}
```

Attacker can first guess a username, and then password

Plaintext Passwords

WARNING: Don't Use

Username	Password
john@hotmail.com	password
betty@gmail.com	ninja

Plaintext Passwords

- It's insecure
- You may find yourself shamed: http://plaintextoffenders.com/

Password Verifiers

- A hash stored in the DB instead of the password

Username	Password
john@hotmail.com	5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8
betty@gmail.com	cbfdac6008f9cab4083784cbd1874f76618d2a97

Common Verifiers

```
md5(hello)  = 5d41402abc4b2a76b9719d91101
sha1(hello) = aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d

sha2_224(hello) =
ea09ae9cc6768c50fcee903ed054556e5bfc8347907f12598aa24193
```

WARNING: Don't Use

Attacks on Verifiers

- Brute Force Attack
- Dictionary Attack
- Rainbow Tables

Dictionary Attack

- Brute forcing takes too long
- Try every reasonable password

Google Attack

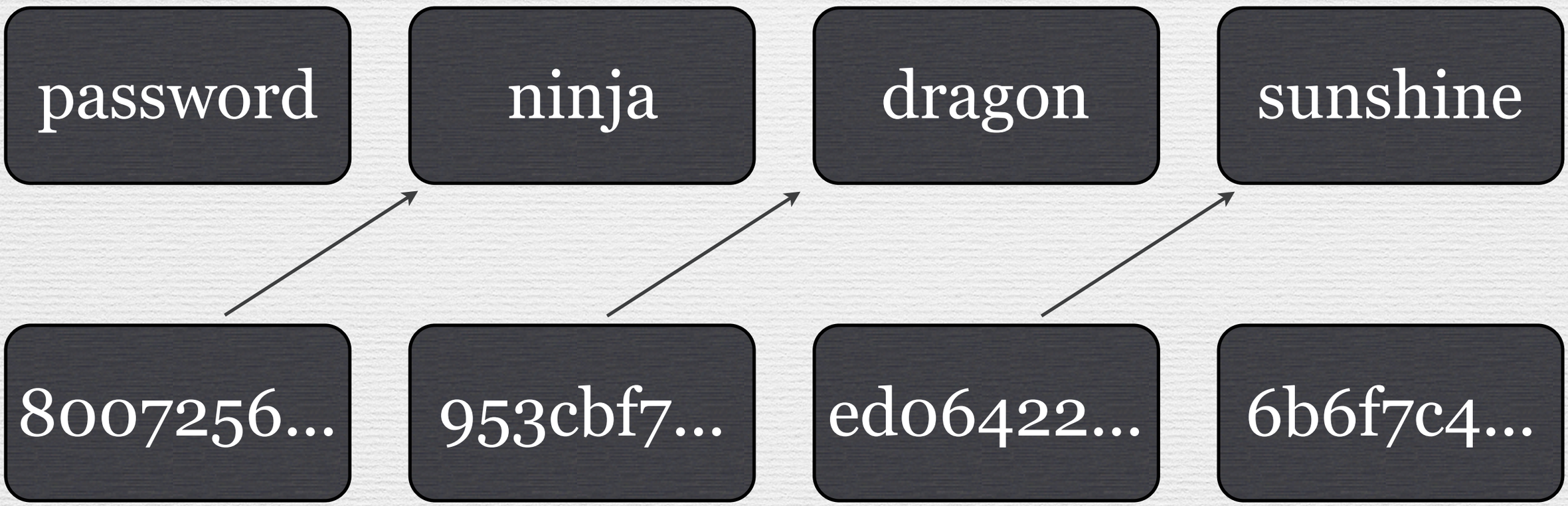
- Password: letmein
- MD5 Hash: 0d107d09f5bbe40cade3de5c71e9e9b7
- Google: https://www.google.co.il/search?q=0d107d09f5bbe40cade3de5c71e9e9b7

Rainbow Tables

password	ninja	dragon	sunshine
8007256...	953cbf7...	ed06422...	6b6f7c4...

Rainbow Tables

r(8007256...) = ninja r(ed06422...) = sunshine



Rainbow Tables

r(8007256...) = ninja r(ed06422...) = sunshine

password

sunshine

Finding Password

- Check if hash is in chain
- Recreate the chain

Rainbow Tables

r(8007256...) = ninja

r(ed06422...) = sunshine

r(953cbf7...) = dragon

password

sunshine

Is 953cbf7 in chain ?

Rainbow Tables

- Store all possible passwords in an efficient way (Hash Chains)
- http://www.freerainbowtables.com/en/tables2/

Salting Back Security

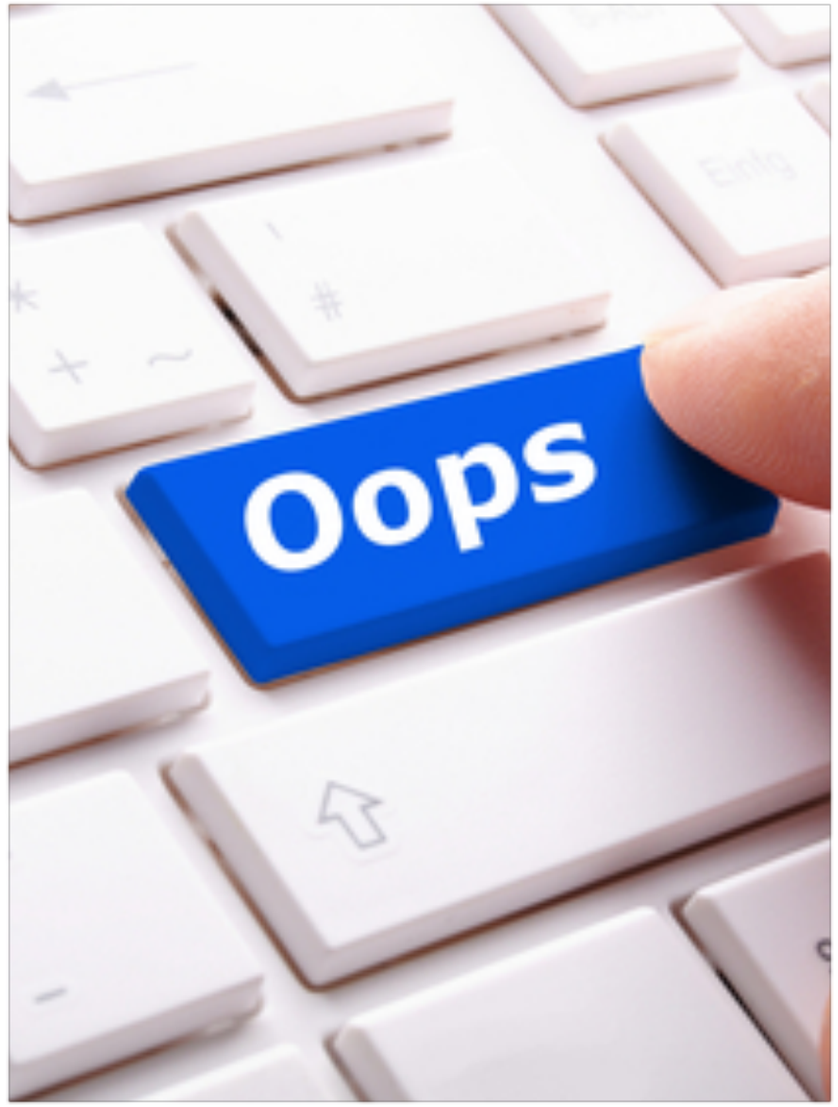
- Add random bytes to the beginning of each password before hashing it to protect against rainbow attacks

Quiz

- Code on the right is used to add a new user
- What's wrong with the code?

```
function add_user(...) {

  $pass = md5($password) ;

  $data = array(
      'username' => $username  ,
      'password' => $pass ,
      'nickname' => $nickname ,
      'org_id' => $org_id ,
      'site_id' => $site_id
      );

  $this->db->insert('users', $data);
  $new_id = $this->db->insert_id() ;

  return $new_id ;

}
```

Password Fails



Leaked Gmail Accounts

5 מיליון כתובות ג'ימייל וסיסמאות דלפו

רשימה של כ-5 מיליון כתובות בשירות ג'ימייל וסיסמאות הופיעה בפורום ברשת. נראה שלא מדובר בפריצה לג'ימייל עצמה, אלא בדליפות משירותי רשת פחות בטוחים ומפישינג. חשוב לבחור סיסמה חזקה, ואחת נפרדת לכל שירות

אהוד קינן

פורסם: 22:26 10/09/2014

BEA Weblogic

- (**CVE-2005-0432**)
- Different error message for invalid user or password
- Versions 7/8

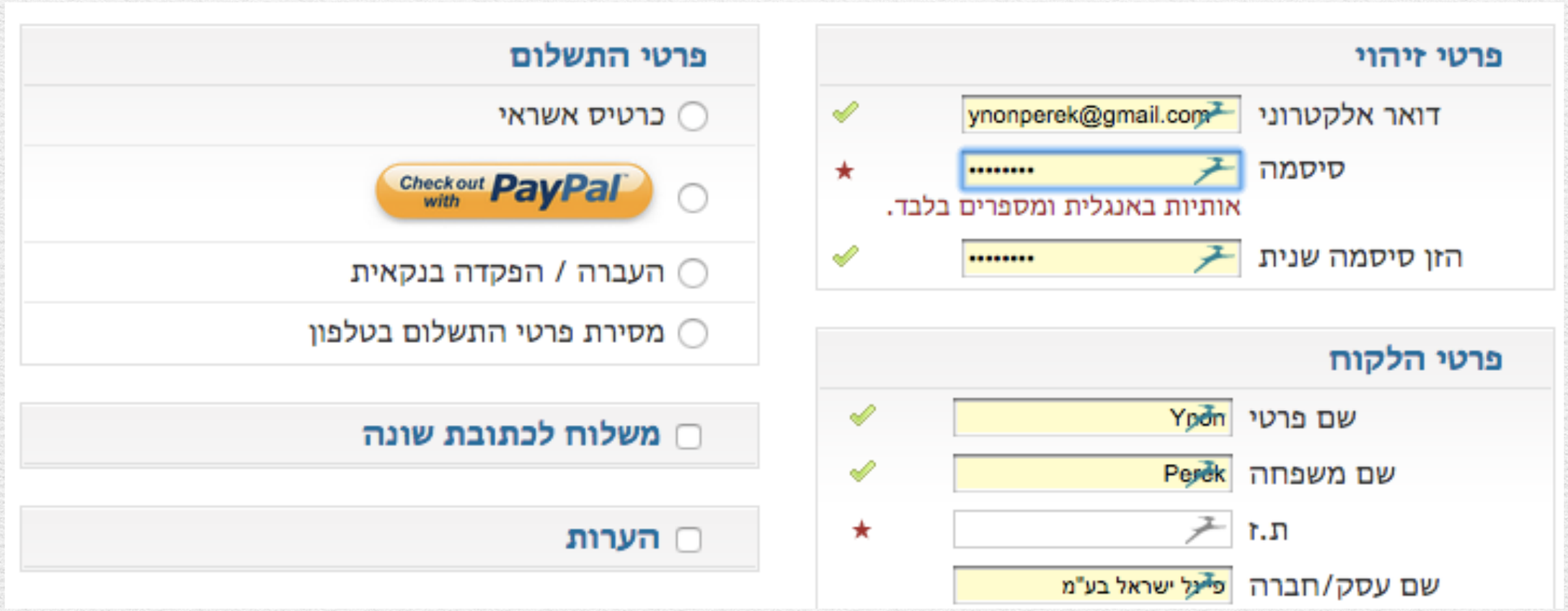
Sarah Palin E-mail

- Yahoo used security question to reset password
- Sarah's info was public

Cold Fusion

- (**CVE-2013-0629**)
- Adobe ColdFusion 9.0, 9.0.1, 9.0.2 and 10
- Allow access restricted directory if password is not configured

Forcing insecure



Passwords Best Practices

Hashing Passwords

- Use salt
- Use iterations
- Better yet: Use bcrypt/scrypt/argon2
- Demo

Java BCrypt

```
String hash = BCrypt.hashpw(userProvidedPassword,
                            BCrypt.gensalt());

if (BCrypt.checkpw(userProvidedPassword, hash)) {
    // Login successful.
}
```

Ruby BCrypt

```ruby
require "bcrypt"

# Calculating a hash
my_password = BCrypt::Password.create(usersPassword)
# Validating a hash
if my_password == usersPassword
  # Login successful
  puts "Welcome"
```

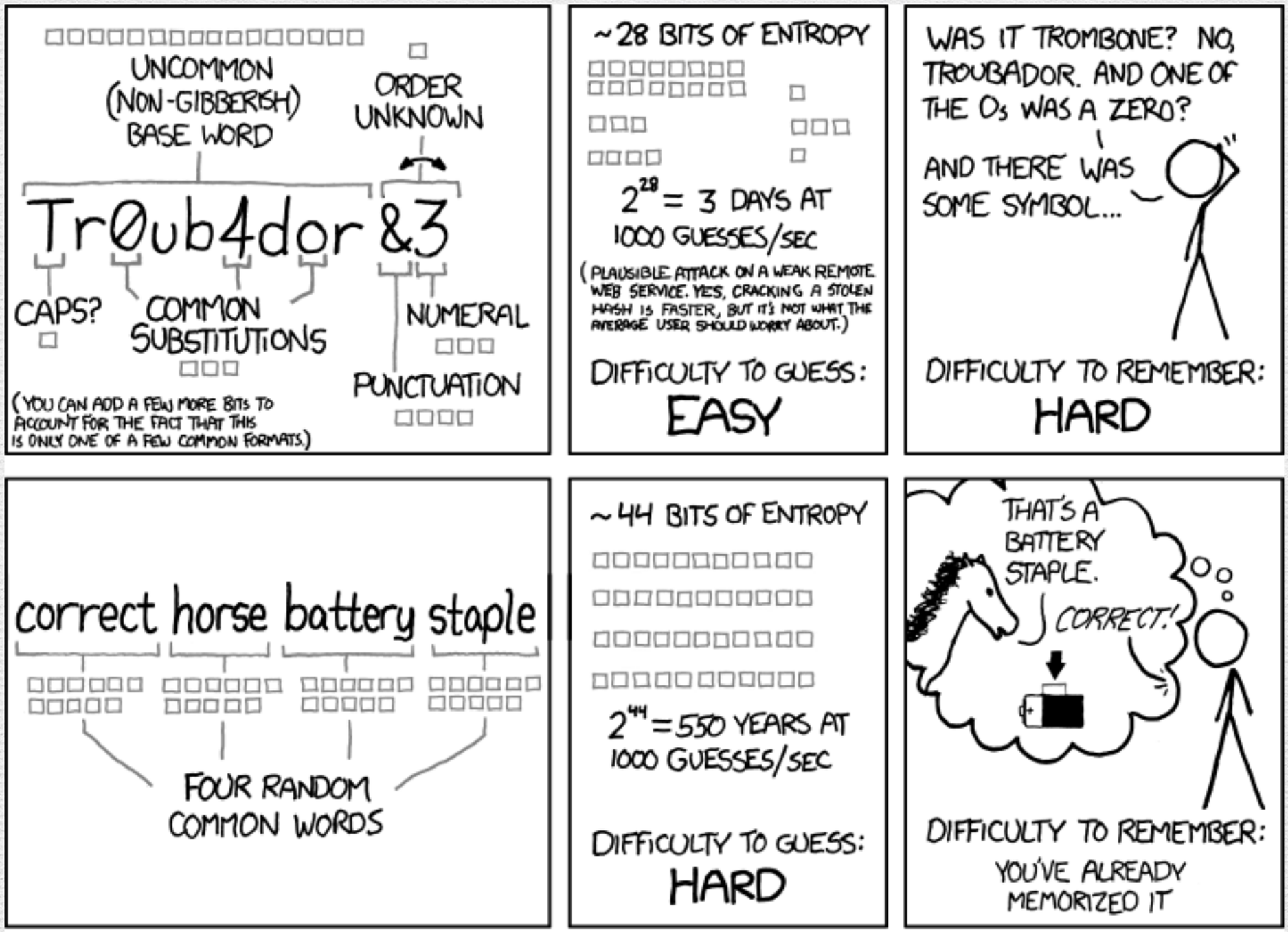
Other Languages

- https://paragonie.com/blog/2016/02/how-safely-store-password-in-2016

Lab

- Given a ruby app that manages passwords badly
- Fix it and upgrade existing passwords
- Starter files in: 03_Crypto/lab_passwords

Choosing Passwords



Choosing Passwords

- As long as possible
- Large character set

Change Passwords

- Changing passwords helps in two ways:
  - Compromised passwords stop working
  - Fights password re-use

Default Passwords

- Before setting initial password use a “safe mode”
- If possible, randomize initial password

Replay Protection

- Use a protected channel
- SSL/TLS or IPSec

Forgot Password

- Don’t use a security question
- Never send plaintext password

Quiz

- WPA-PSK uses PBKDF2 to hash the password
- salt value is the SSID
- Iteration count 4096
- Is it vulnerable to rainbow attack ?

Q & A

Misusing Passwords