



# HOW SECURITY AFFECTS THE PEOPLE BEHIND THE CODE

---

**DR. PHILIPPE DE RYCK**

<https://PragmaticWebSecurity.com>

# DR. PHILIPPE DE RYCK

- Deep understanding of the web security landscape
- Google Developer Expert (not employed by Google)
- Course curator of the  **SecAppDev** course  
(<https://secappdev.org>)



## Pragmatic Web Security

High-quality security training for developers and managers

Custom courses covering web security, API security, Angular security, ...

Consulting services on security, OAuth 2.0, OpenID Connect, ...

@PHILIPPEDERYCK

[HTTPS://PRAGMATICWEBSECURITY.COM](https://pragmaticwebsecurity.com)



**Have you ever felt unappreciated trying to get security right?**



# 21 Facebook Stored Hundreds of Millions of User Passwords in Plain Text for Years

MAR 19

Hundreds of millions of **Facebook** users had their account passwords stored in plain text and searchable by thousands of Facebook employees — in some cases going back to 2012, KrebsOnSecurity has learned. Facebook says an ongoing investigation has so far found no indication that employees have abused access to this data.







**Philip McHugh**

@PhilipMcHugh\_IE



Replying to [@briankrebs](#)

I would of been surprised if it was salted 😊 but yea, big user base storing plain text passwords, come on, really 🙄

4:25 PM · Mar 21, 2019 · [Twitter for Android](#)

---

**1** Retweet   **1** Like



@PhilippeDeRyck



**♠ Calamity ♠**  
@TenCentsIsFree



Replying to [@joemccann](#)

Even the best have lazy developers 😄

5:09 AM · Mar 22, 2019 · [Twitter for Android](#)



# Twitter advising all 330 million users to change passwords after bug exposed them in plain text

*There's apparently no evidence of any breach or misuse, but you should change your password anyway*

By [Chaim Gartenberg](#) | [@cgartenberg](#) | May 3, 2018, 4:21pm EDT

# GitHub says bug exposed some plaintext passwords

A small but unspecified number of GitHub staff could have seen plaintext passwords.



By [Zack Whittaker](#) for [Zero Day](#) | May 1, 2018 -- 21:23 GMT  
(22:23 BST) | Topic: [Security](#)





fb_user_id	email	password
138263115	info@secappdev.org	CantHackThis
138263116	rmunroe@example.com	Correct horse battery staple
138263117	philippe@pragmaticwebsecurity.com	P@zzw0rd!







fb_user_id	email	password
138263115	info@secappdev.org	\$2a\$13\$548oGCnkyiJ8RMJzozExpu1q2PJ3IH4qJqopdN068vb76q87lqjjS
138263116	rmunroe@example.com	\$2a\$13\$yoQBhPFrCwBofYoewc7hhO73ZFv9VmD8ecek.9JNHKS9bfqbS8cPq
138263117	philippe@pragmaticwebsecurity.com	\$2a\$13\$fBzHQZvCBG5nxPD.Kpzbluc9KEycyE1n7D9ARvJ1ZOi49t0KNuaES



GEFORCE GTX

GEFORCE GTX

GEFORCE GTX

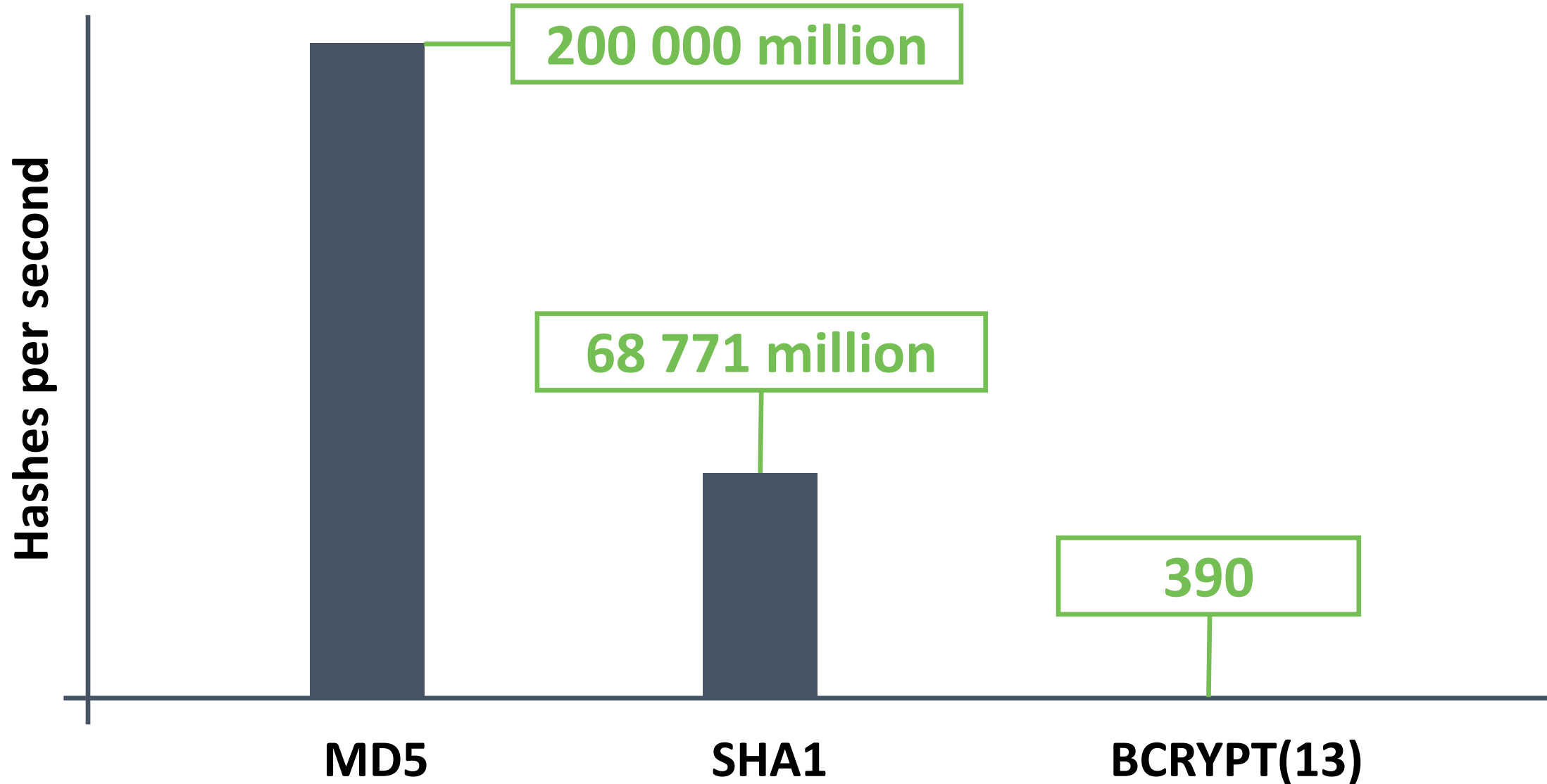
GEFORCE GTX

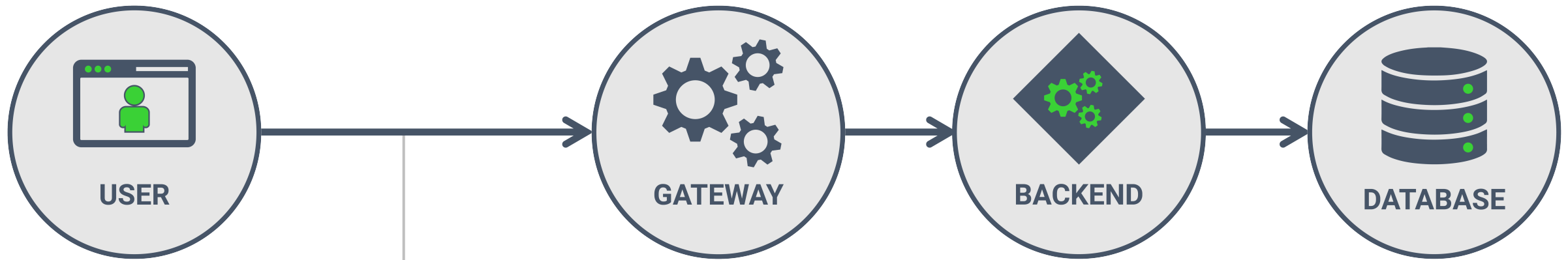
GEFORCE GTX

GEFORCE GTX

GEFORCE GTX

# IS HASH CRACKING REALLY THAT FAST?



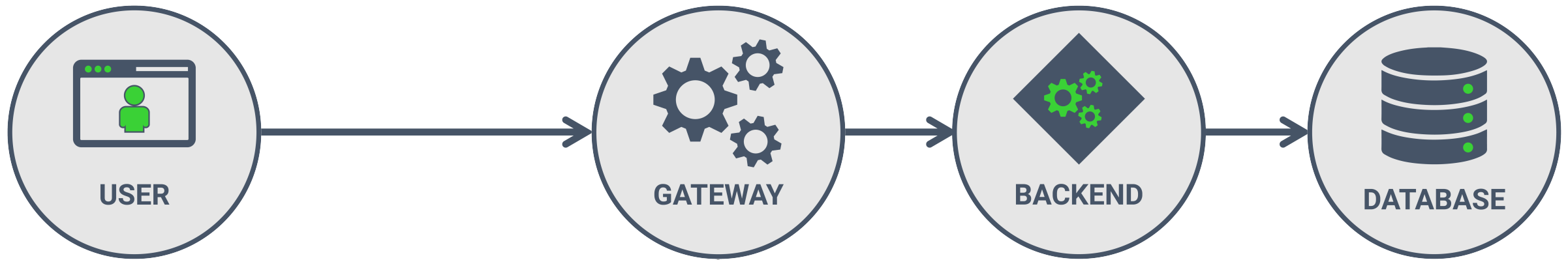


• *Submitting the login form*

---

```
1 POST /login
2 Host: facebook.com
3 Content-Type: application/x-www-form-urlencoded
4
5 username=philippe@pragmaticwebsecurity.com
6 &password=P%40zzw0rd%21
```

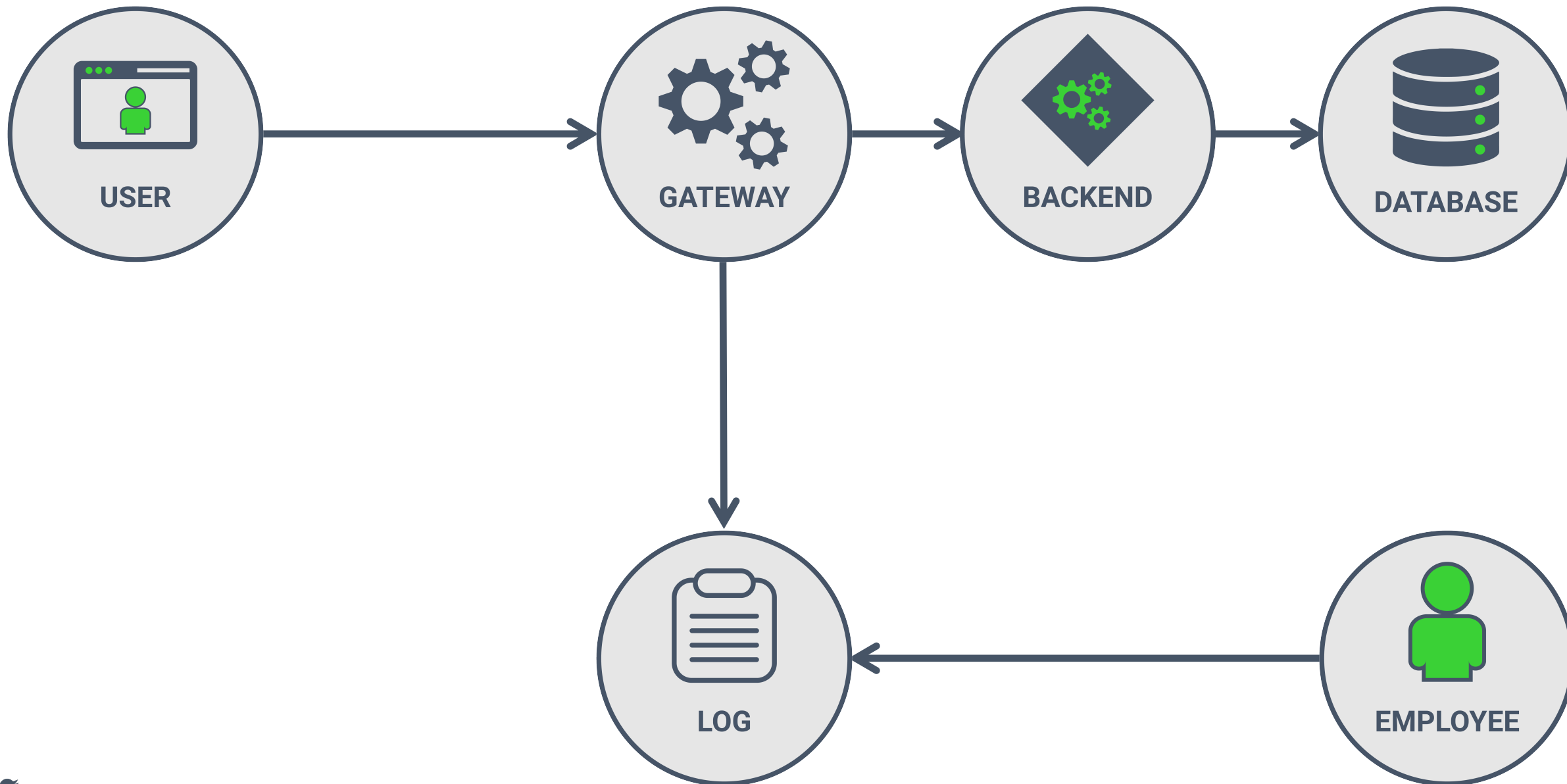
---



*Logging the POST request*

```
1  {
2    "url": "https://facebook.com/login",
3    "method": "POST",
4    "type": "application/x-www-form-urlencoded",
5    "body": {
6      "username": "philippe@pragmaticwebsecurity.com",
7      "password": "P@zzw0rd!"
8    }
9  }
```







19

The logback version 0.9.27 introduced [replacement capability](#). Replacements support regular expressions. For example, if the logged message was "userid=alice, pswd='my secret'", and the output pattern was

```
"%d [%t] $logger - %msg%n",
```

you just modify the pattern to

```
"%d [%t] $logger - %replace(%msg){\"pswd='.*'\", \"pswd='xxx'\"}%n"
```

Note that the above makes use of [option quoting](#).

The previous log message would be output as "userid=alice, pswd='xxx'"



Analyze your application to identify sensitive input fields

Configure automated logging to replace sensitive fields

Carefully comb through your logs to find exposed data









*Equifax uses Apache Struts 2 to build applications*

**a patched version of *Struts2* fixes a remote code execution vulnerability**

March 7<sup>th</sup>, 2017

*Equifax* discovers the breach of their systems

July 29<sup>th</sup>, 2017

*Equifax* announces the breach of 147 million people

September 7<sup>th</sup>, 2017

May 2017

**attackers escalate the attack to full-scale data exfiltration**

March 10<sup>th</sup>, 2017

**attackers start probing *Equifax* systems using the *Struts* vulnerability**





**Ask Equifax**

@AskEquifax

Follow



Happy Friday! You've got Stevie ready and willing to help with your customer service needs today!

5:00 AM - 8 Sep 2017

11 Retweets 9 Likes



66



11



9



# Former Equifax CEO blames breach on one IT employee

Someone didn't install a patch when they should've. That's it. That caused the 145-million person data leak.



David Lumb, @OutOnALumb  
10.03.17

75  
Comments

3762  
Shares



U.S. House of Representatives  
Committee on Oversight and Government Reform



### The Equifax Data Breach

Majority Staff Report  
115th Congress

December 2018

● An audit in 2015 already revealed a grossly inadequate patching process

● The interview with Equifax executives revealed that they had to guess who the owner for a particular application was

● Following the initial alert to patch, the matter was mentioned on a Powerpoint slide once



**Equifax's CEO did not prioritize cybersecurity.**

**[The CIO] testified [the CEO] held quarterly senior leadership team meetings where IT security was just one of the many topics discussed.**

**The information [the CEO] did receive was presented by the head of the legal department, who did not have any background in IT or security – rather than [the CSO], the company's IT security expert.**







**Equifax had a network monitoring device installed**



**Which failed to detect the ongoing attack**



**Because it's TLS certificate expired**



**19 months before the attack**





**Security cannot be handled by engineers alone**

**Buy-in from the top is required to drive security forward**

**Focus on building the right security culture**



> 97%

of code in a modern web app are dependencies

```
$ ng new clean-app
```

```
? Would you like to add Angular routing? Yes
```

```
? Which stylesheet format would you like to use? Sass
```

```
added 1169 packages from 1030 contributors and audited  
42445 packages in 28.75s
```



```
$ cloc node_modules/
```

Language	files	blank	comment	code
JavaScript	12683	145344	525680	1773037
JSON	1555	104	0	161571
Markdown	1385	65564	4	157446
TypeScript	2892	9625	90588	104376
HTML	274	1656	218	33724
CSS	148	299	2301	22382
C++	75	3784	3501	22332
Python	51	4205	7606	18695
C/C++ Header	101	2758	1858	15114
LESS	482	1611	410	11321
XML	20	3237	1300	7617
YAML	163	140	112	2416
Bourne Shell	18	292	333	1500
SVG	8	2	2	776
make	30	236	39	715
Windows Module Definition	7	115	0	641
DTD	1	179	177	514
...				
SUM:	19983	239598	634354	2336228



```
$ ng new clean-app
```

```
? Would you like to add Angular routing? Yes
```

```
? Which stylesheet format would you like to use? Sass
```

```
added 1169 packages from 1030 contributors and audited  
42445 packages in 28.75s
```



# 78%

**of vulnerabilities occur in indirect dependencies**

```
bash
Path      @angular-devkit/build-angular > webpack >
          terser-webpack-plugin > serialize-javascript
More info  https://npmjs.com/advisories/1426



|                 |                                                        |
|-----------------|--------------------------------------------------------|
| <b>Moderate</b> | <b>Cross-Site Scripting</b>                            |
| Package         | serialize-javascript                                   |
| Dependency of   | webpack [dev]                                          |
| Path            | webpack > terser-webpack-plugin > serialize-javascript |
| More info       | https://npmjs.com/advisories/1426                      |



found 20 vulnerabilities (3 low, 5 moderate, 12 high) in 45666 scanned packages
run `npm audit fix` to fix 19 of them.
```





- Alerts
- Advisories
- Policy

## Security Alerts

Automated security updates ▾

Dismiss all ▾

⚠ 3 Open

✓ 0 Closed

Sort ▾

⚠

handlebars

high severity

📅

29 Dec 2019 by GitHub

📄 tournament-frontend/package-lock.json

🔗 #25

⚠

serialize-javascript

moderate severity

📅

08 Dec 2019 by GitHub

📄 tournament-frontend/package-lock.json

⚠

com.fasterxml.jackson.core:jackson-databind

critical severity

📅

24 Sep 2019 by GitHub

📄 tournament-api-tests/pom.xml

🔗 #24

GitHub tracks known security vulnerabilities in some dependency manifest files. [Learn more about security alerts.](#)

A person wearing glasses is looking at a computer screen. The screen displays some code or data. The background is dark and out of focus.

**Keep applications up to date,  
so you can apply security patches quickly**

**Have a fast security release process,  
next to a slower feature release process**

**Test your process with  
a non-critical vulnerability to make sure it works**





**Open source developers are awesome!**

**You don't owe them anything for using their software**

**They don't owe you anything either!**



**Get off your ass.**





**dougwilson** commented on 5 May 2019

Member



I don't think you understand that when this happens, I literally get hate emails to my email address. I removed my email address from GitHub, but they still get it. Here is a literal quoted email:

Fucking release an express that does not have a security vulnerability out of the box. I can no longer check in code to my company CI because audit blocks it. Get off your ass.



**dougwilson** commented on 5 May 2019

Member



I should just disappear from GitHub / Node.js at this point... I can't catch a break and even enjoy this weekend :(



*finalhandler* sets a default CSP policy on its error pages (`default-src 'self'`)

A researcher opens an issue to recommend a stricter policy (`default-src 'none'`)  
February 20<sup>th</sup>, 2019

Dependency audits start failing on the NPM advisory

May 3<sup>rd</sup>, 2019

**Package owner suffers outrageous abuse from other developers**

May 5<sup>th</sup>, 2019

May 5<sup>th</sup>, 2019

SourceClear picks up the report as well, restarting the cycle

May 3<sup>rd</sup>, 2019

A user warns that this problem is now an NPM security advisory

May 2019


The issue is also reported on HackerOne, with severity "none"

March 4<sup>th</sup>, 2019

The package owner and researcher exchange a couple of messages







Let us develop respect for all living  
things. Let us try to replace violence  
and intolerance with understanding  
and compassion. And love.

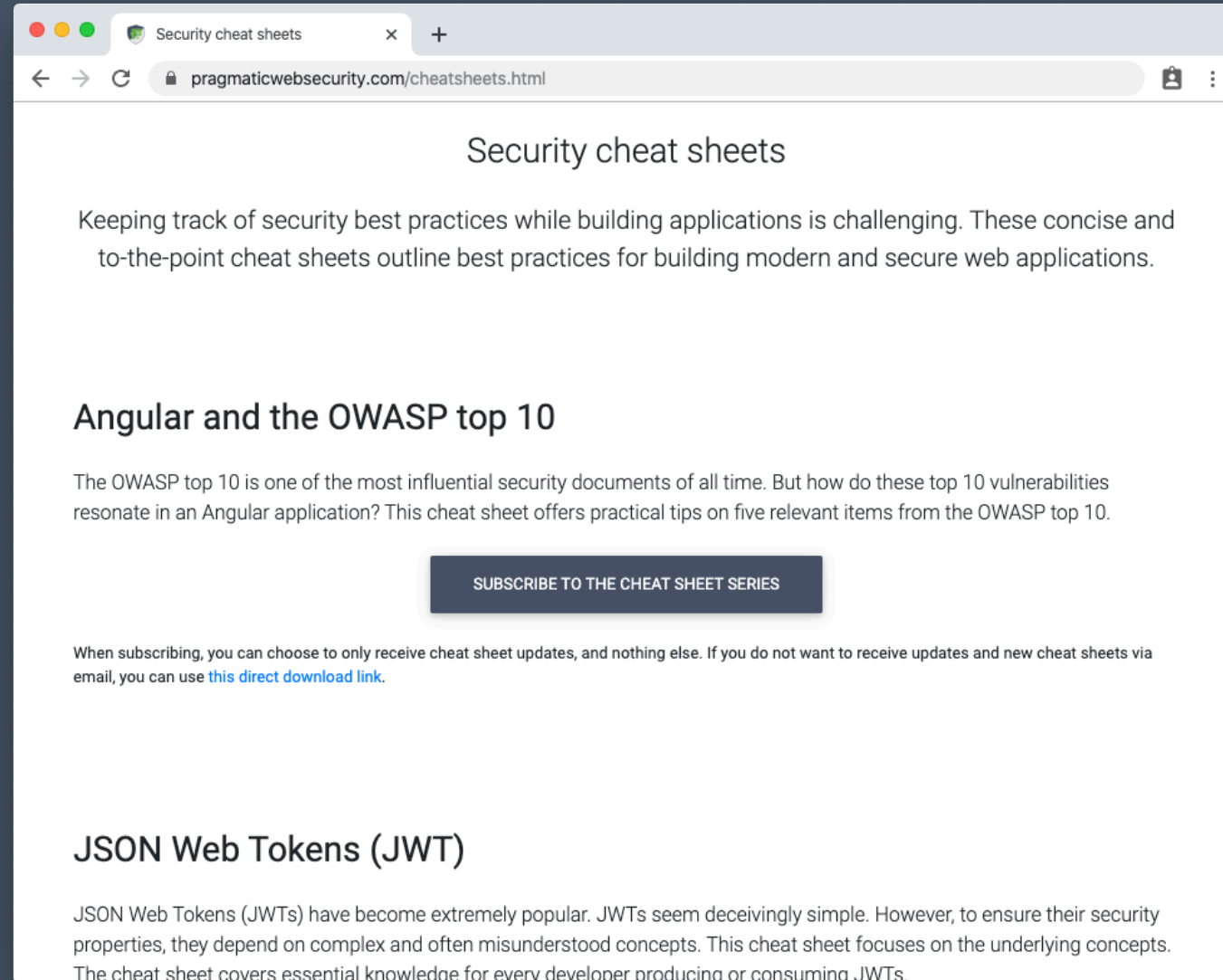
Jane Goodall

quote fancy





# FREE SECURITY CHEAT SHEETS FOR MODERN APPLICATIONS





March 9<sup>th</sup> – 13<sup>th</sup>, 2020  
Leuven, Belgium

A **week-long course** on Secure Application Development

Taught by **experts** from around the world

**38** in-depth lectures and **3** one-day workshops

*<https://secappdev.org>*

*A yearly initiative from the SecAppDev.org non-profit, since 2005*





**Pragmatic Web Security**

Security for developers

# THANK YOU!

*Follow me on Twitter to stay up to date  
on web security best practices*



**@PhilippeDeRyck**