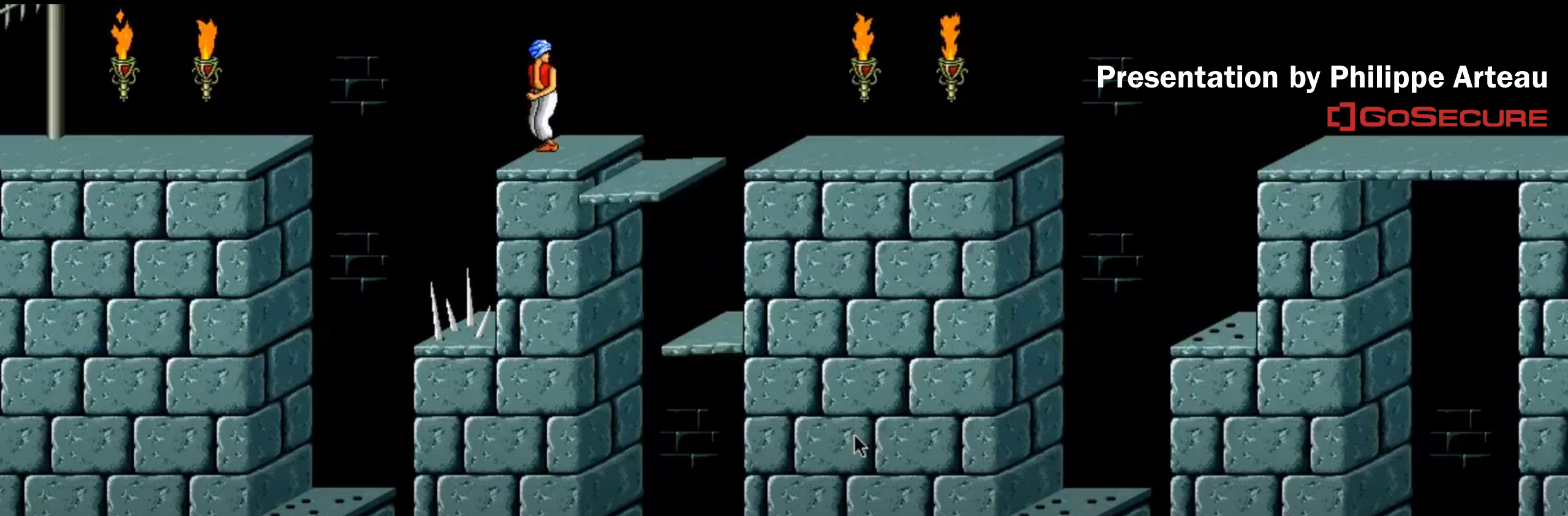
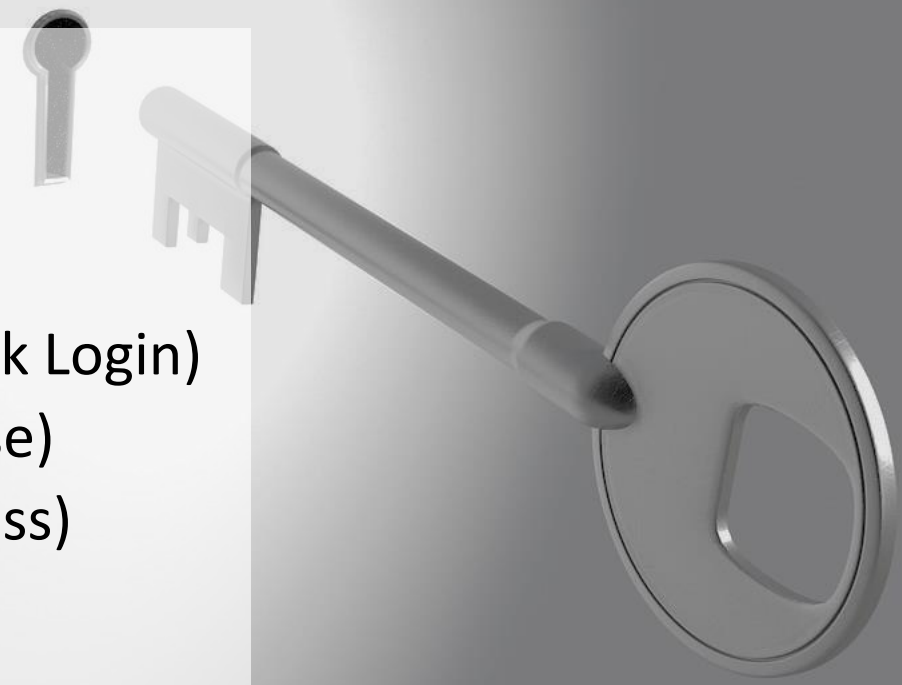
Privacy pitfalls for your web application



Presentation by Philippe Arteau
GoSecure

Agenda

- Introduction
- Information leakage
  - Hashed email (Gravatar)
  - Third party profile integration (Facebook Login)
  - Metadata in documents (Image, Keybase)
  - Search features (SonarSource, WordPress)
  - GraphQL / REST data binding
  - Side-channel attacks (Outlook, GMail)
- Conclusion



What is OSINT?

Open-Source Intelligence

“Collection and analysis of data gathered from open sources to produce actionable intelligence”

In this context, it is unrelated to Open-Source

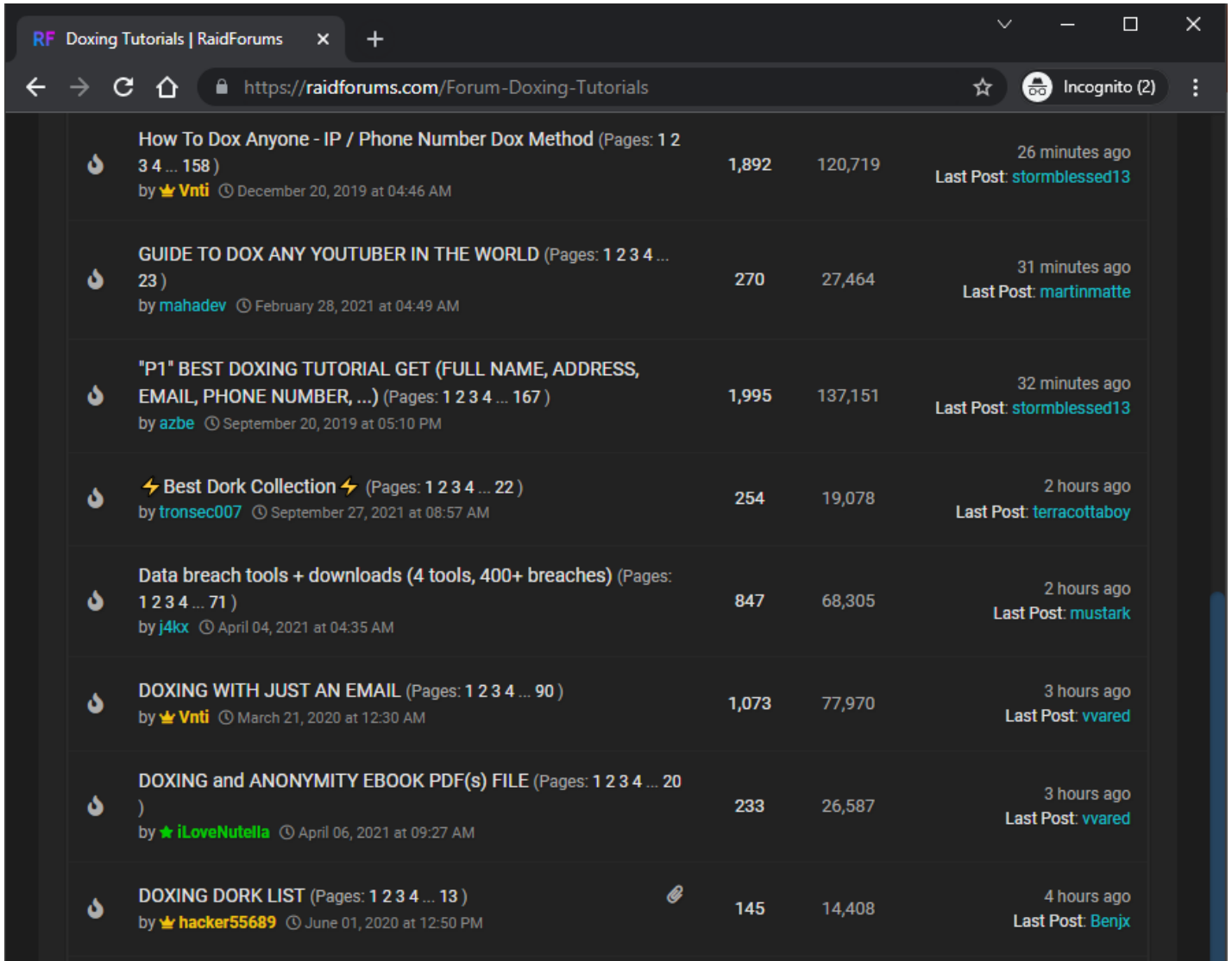
Open-source refer to online publications, discussion groups and social media.

Your web application could become an “open-source”.

https://en.wikipedia.org/wiki/Open-source_intelligence

Information asymmetry

- Lots of documentation for attackers
  - Techniques
  - Tools
  - Shared leaks
- No clear documentation for application developers for defence



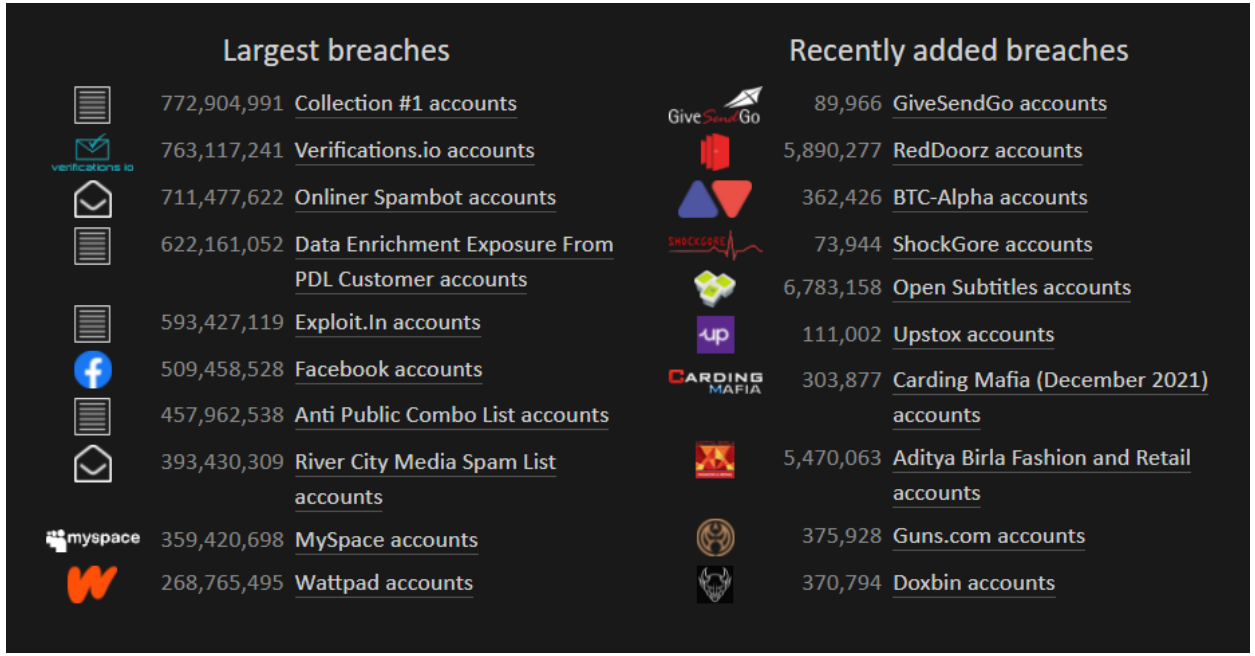
Leaks and emails

In this presentation, most of metadata leakage are going to either:

- Disclose emails, name, organization, physical address, social network connection

Why email privacy is important?

- The reality of password leaks
- Account correlation
- Privacy law:
  - GDPR (EU)
  - CCPA (California)
  - Loi sur la Protection des Renseignements Peronnels (Québec) *



* Coming soon: https://www.legisquebec.gouv.qc.ca/fr/document/lc/P-39.1

Source: ';--have i been pwned?

Gravatar URLs

Gravatar

Gravatar is a service for providing globally unique avatars.

Example of Gravatar integration

14 Comments

punchonpress Jan 31st at 11:24 pm

Thanks for info. I was wondering how I can buy some plugins that have been bugging mw such as wp bakery

★ Liked by 2 people

Tex Feb 1st at 9:52 am

This is all great features. Has made my turned all of my websites into native apps. I love it. I won't have to pay dev fees to publish on Ios or Droid. Connecting my site to jetpack was a great experience for me. Thanks again for making development more easy.

★ Liked by 2 people

Tony Feb 2nd at 4:20 pm

I feel it would be useful if all paid plans had access to at least some free or paid plug-ins

★ Liked by 4 people

Gravatar URLs

MD5("**philippe@confoo.ca**") => 06b856f7ee1266fbf86eaa018f5b0906

https://secure.gravatar.com/avatar/
06b856f7ee1266fbf86eaa018f5b0906?s=96&d=identicon&r=G

- 06b856f7ee1266fbf86eaa018f5b0906 : email hash
- s=96 : image size
- d=identicon : default image if profile is not found
- r=G : Rating (May contains profanity, violence, nudity or drugs)

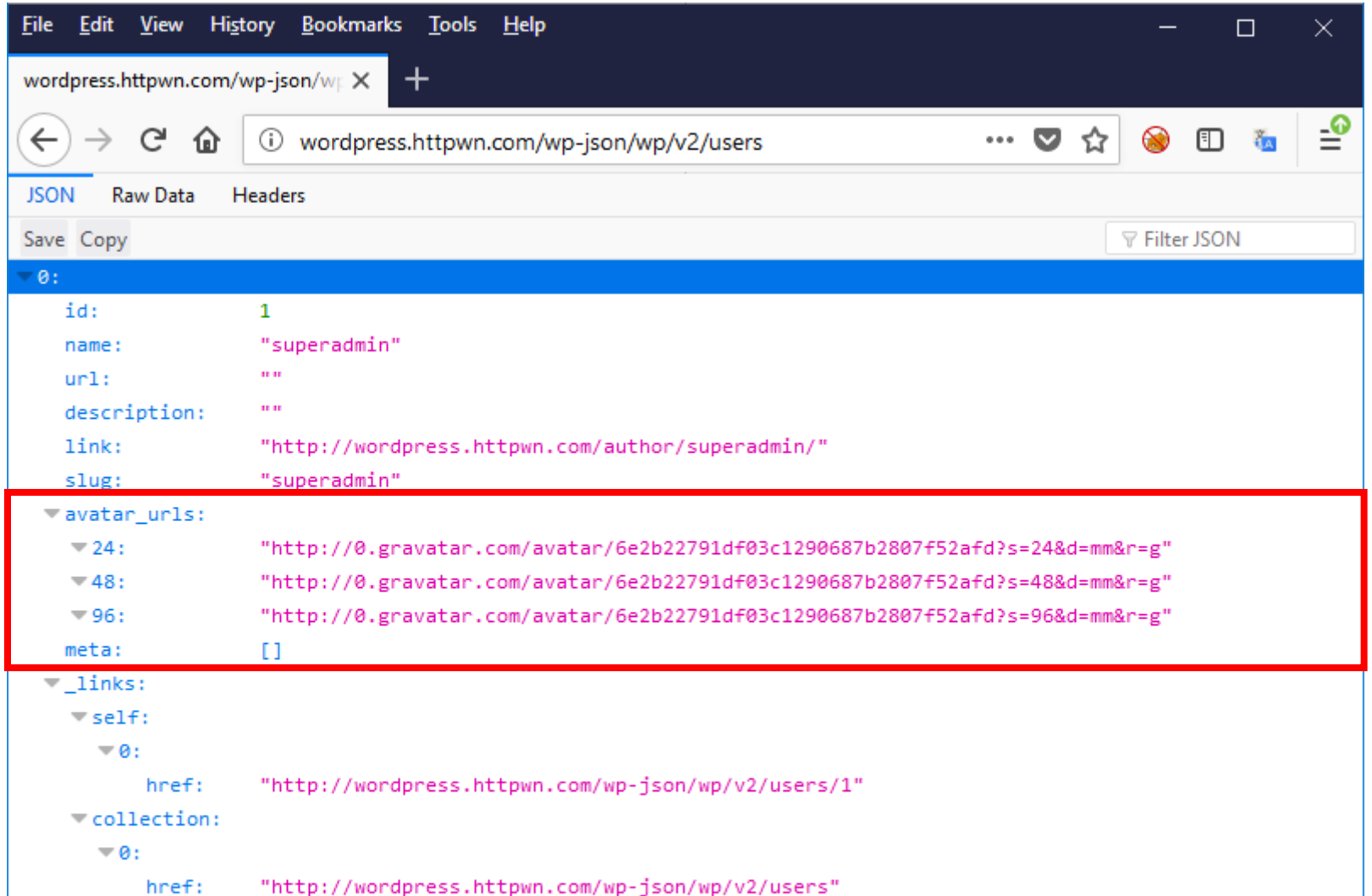
Sites using Gravatar

- WordPress blogs

Organization	Number of Users	Alexa Rank
newsfeed.time.com	186	1524th
vimeo.com/blog	177	170th
blog.ted.com	120	1291st
blog.etsy.com	2570	76th
techcrunch.com	57	1754th
wired.com	1052	1525th
devblogs.microsoft.com	1000+	21st
gblogs.cisco.com	123	824th
godaddy.com/garage	1430	172nd
books.disney.com	14	4436th

- StackOverflow
- SuperUser
- AskUbuntu
- Gitlab
- Bitbucket
- SonarSource

Wordpress Users



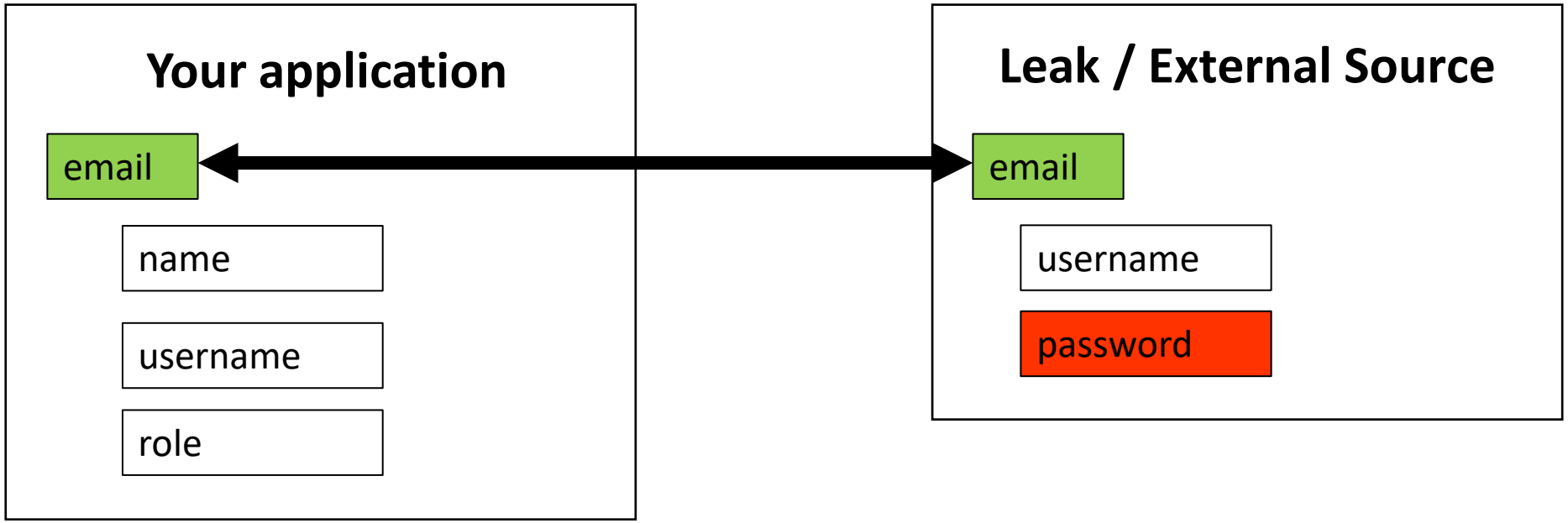
The risk

- When an URL is exposed, you should considered that the hash email can be reversed. 60%+ of all registered account were
- The risk need to be evaluated based on the information attached to the users.
  - Is the user anonymity important?
  - Could the metadata affect the user elsewhere?
  - IP / Geolocation : Located individual, Targeted Phishing
  - Company name : Target attack to employee of a specific company
  - Phone number, email: Building clients list (spam list)

More info : https://www.gosecure.net/blog/2021/03/02/emails-disclosure-on-wordpress/

Credential Stuffing

An adversary tries known username/password combinations against different systems, applications, or services to gain additional authenticated access.



Recommendations

- Avoid using Gravatar if user anonymity is important
  - If the avatar are not shown publicly, the risk is much lower.
- Download the image server-side rather than pointing to Gravatar service
- Implement/Integrate a second factor for authentication

Login with Facebook

(Also applies to other SSO)

Login with Facebook & Graph API

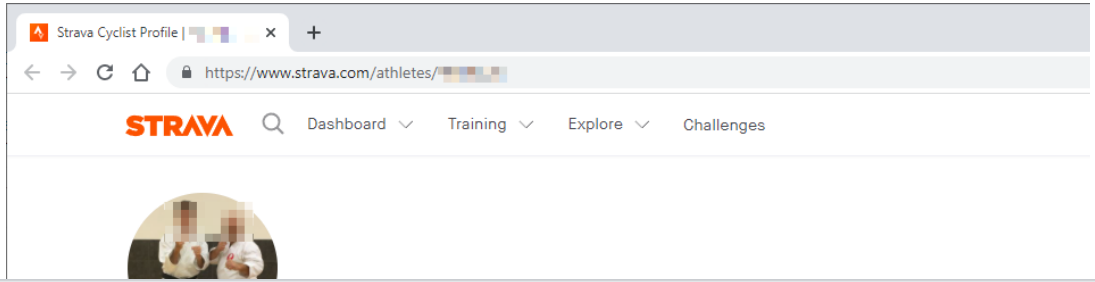
When integrating Login with Facebook feature, it is possible to obtain the email, the full name and a profile image.

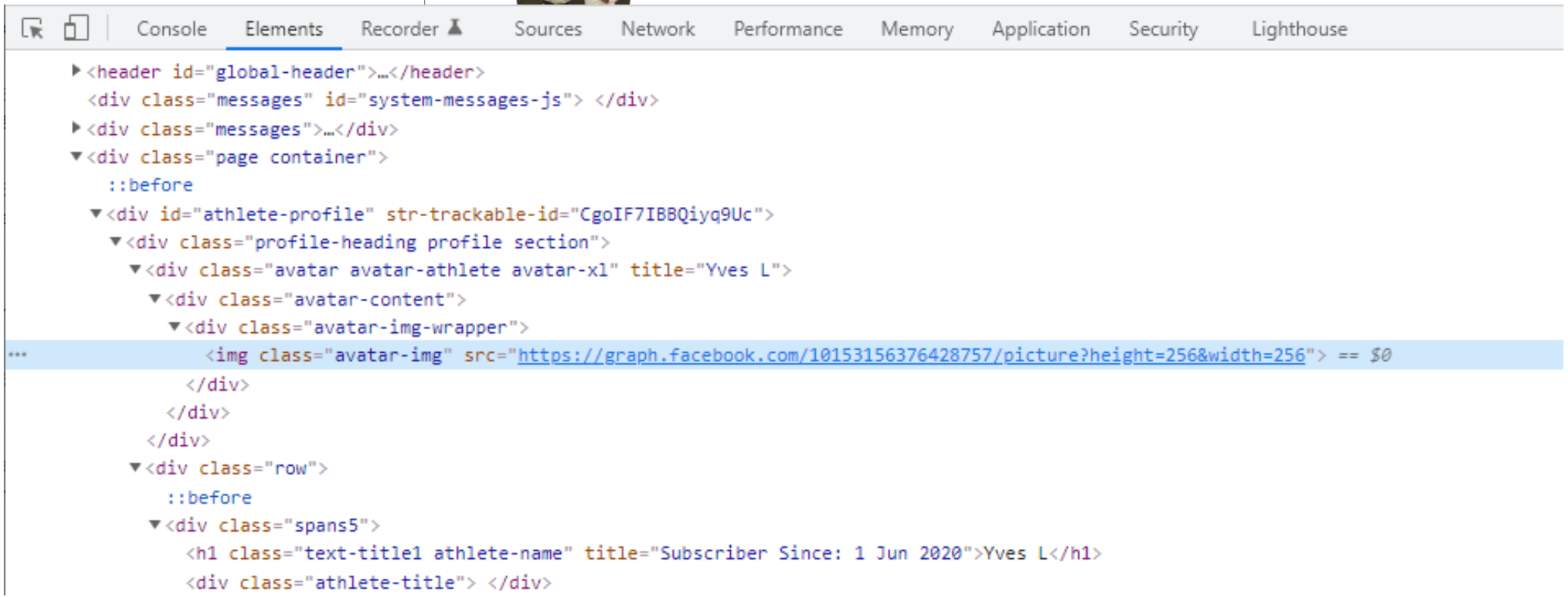
The profile image comes in the form an URL.

https://graph.facebook.com/XYZ/picture

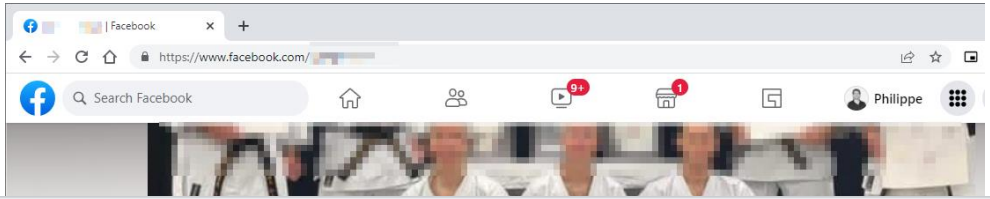
https://graph.facebook.com/XYZ/picture?height=256&width=256

Can image be linked back to a Facebook profile? (1/4)





Can image be linked back to a Facebook profile? (2/4)

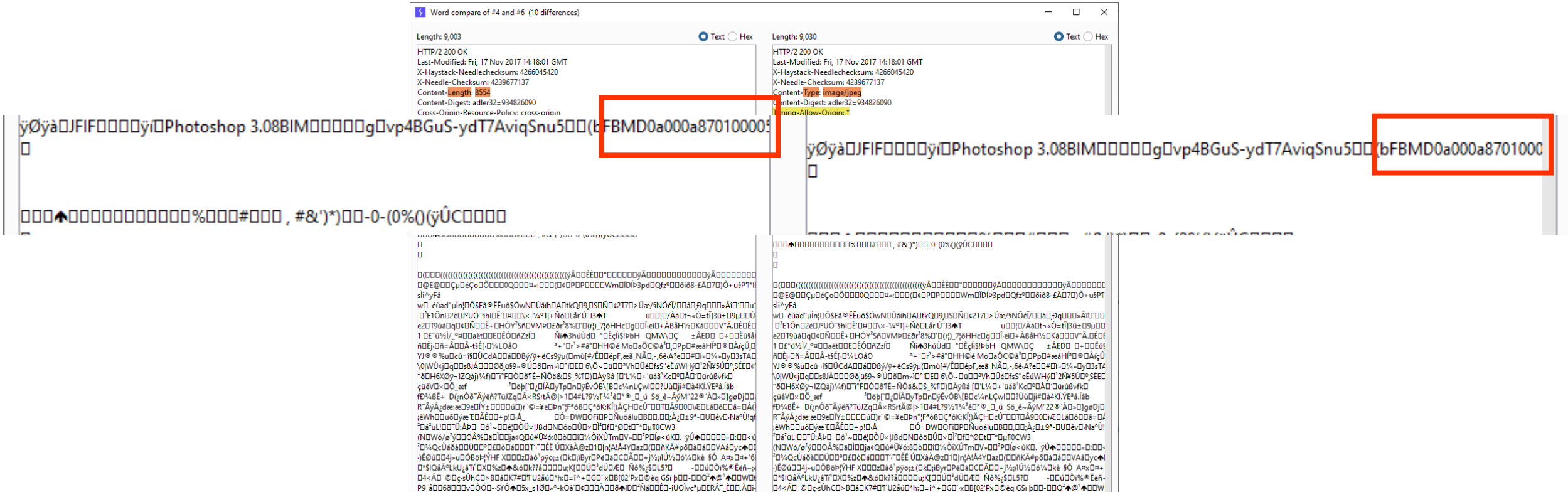


Console | Elements | Recorder | Sources | Network | Performance | Memory | Application | Security | Lighthouse

```
▼<div class="q9uorilb l9j0dhe7 pzggbiyp du4w35lb">
  ▼<svg aria-label="[redacted]" class="pzggbiyp" data-visualcompletion="ignore-dynamic" role="img" style="height: 168px; width: 1
  ">
    ▶<mask id="jsc_c_p">…</mask>
    ▼<g mask="url(#jsc_c_p)">
      <image x="0" y="0" height="100%" preserveAspectRatio="xMidYMid slice" width="100%" xlink:href="https://scontent-yyz1-1.xx.f
      et/v/t31.18172-1/p200x200/23632648…yyz1-1.xx&oh=00_AT_z3EwCecAd0nofE9CfcCfpEhsM2Ot1_ahyaHQB-Py01Q&oe=623AA4F9" style="heigh
      px; width: 168px;"></image> == $0
      <circle class="mlqo0dh0 georvekb s6kb5r3f" cx="84" cy="84" r="84"></circle>
    </g>
  </svg>
  <div class="i09qtzwb n7fi1qx3 b5wmifdl hzruof5a pmk7jnqg j9ispegn kr520xx4 c5ndavph art1omkt ot9fgl3s rnr61an3 s45kf179 emlxlay
  p75w spb7xbtv" data-visualcompletion="ignore"></div>
 </div>
</a>
</div>
</div>
<div class="pfnyh3mw wkznzc2l dycxa26y hlyrhctz"></div>
▶<div class="j83agx80 psu0lv52 mpmpiqla ahl66waf tmq14sqq rux31ns4 gy1kt949 sjcfkmk3 dti9y0u4 nyziof1z">…</div> flex
▶<div class="qublvx3c ew1m30ut dwxd3oue rymlbt9a izvtuzxg eja8ua2n dkw9rofs dti9y0u4 t9rp7xwc nyziof1z h374gqy3">…</div>
</div>
```

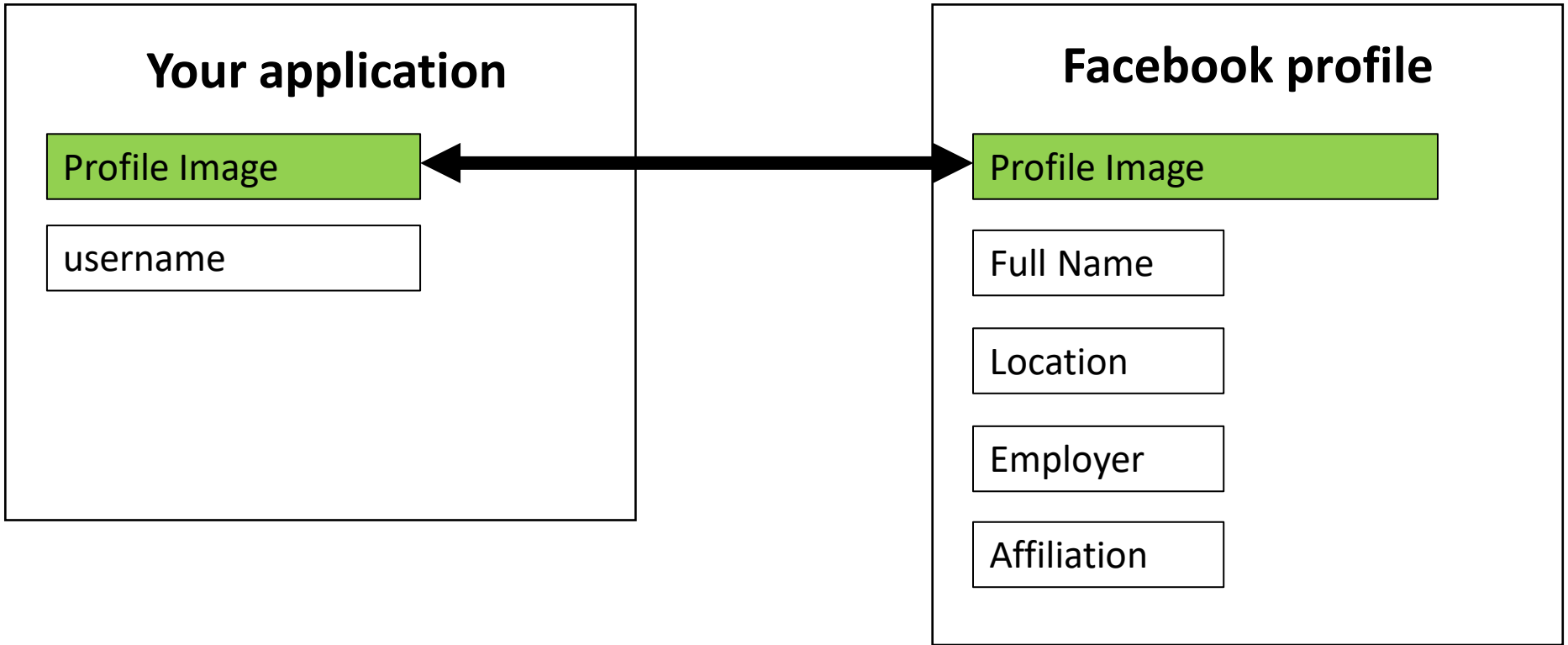
Can image be linked back to a Facebook profile? (3/4)

- Both URL will return the same picture byte for byte
  - Public profile : **http://scontent-yyz1-1.xx.fbcdn.net/v/...**
  - Graph API (FB Login): **https://graph.facebook.com/XYZ/picture**



Can image be linked back to a Facebook profile? (3/4)

In order to correlate the profile picture from the Facebook Graph API to the one from public profile, the attacker needs to **crawl multiples profiles** and keep either a signature from the image or the FBMD code from the image.

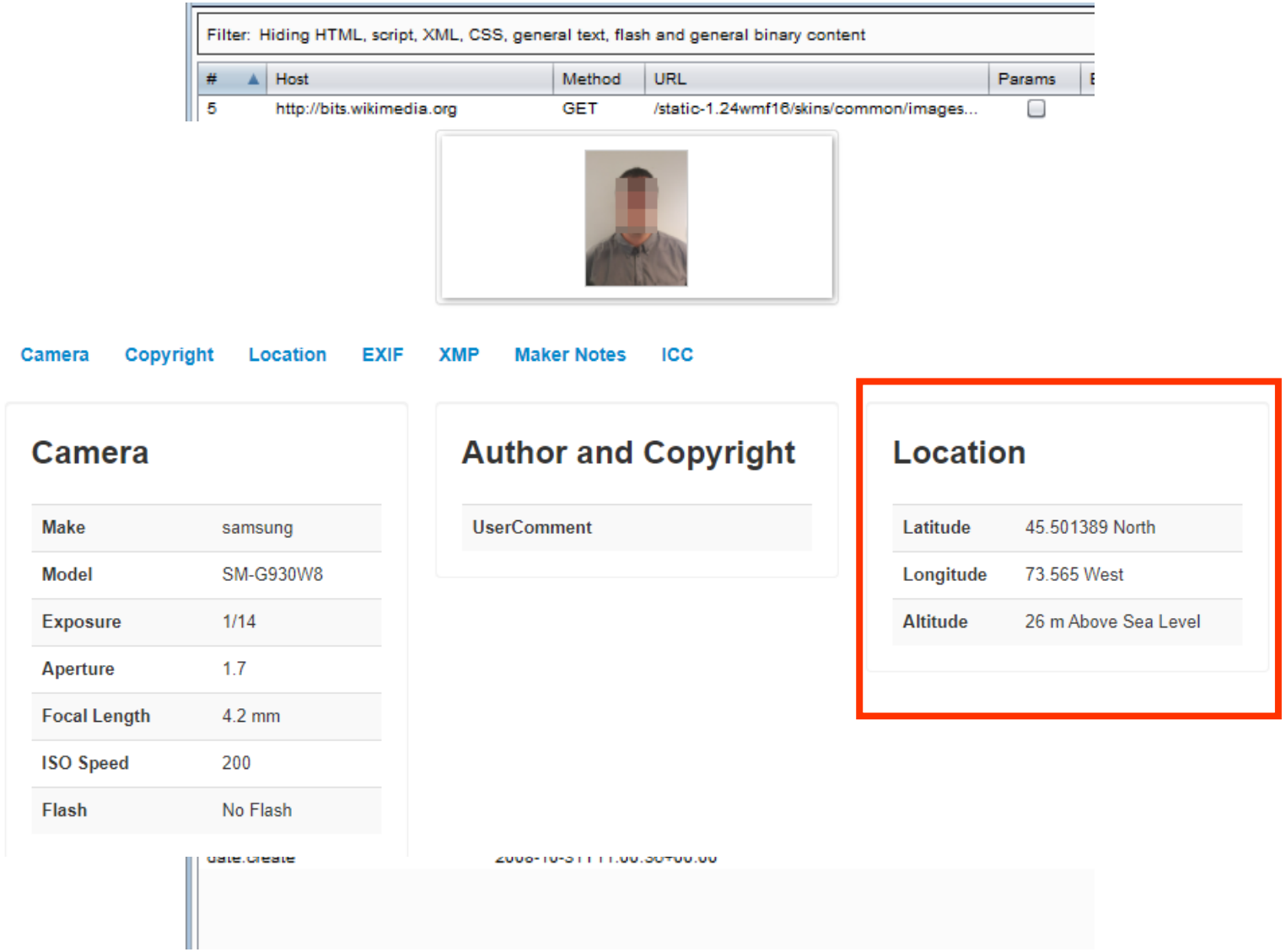


Metadata in documents

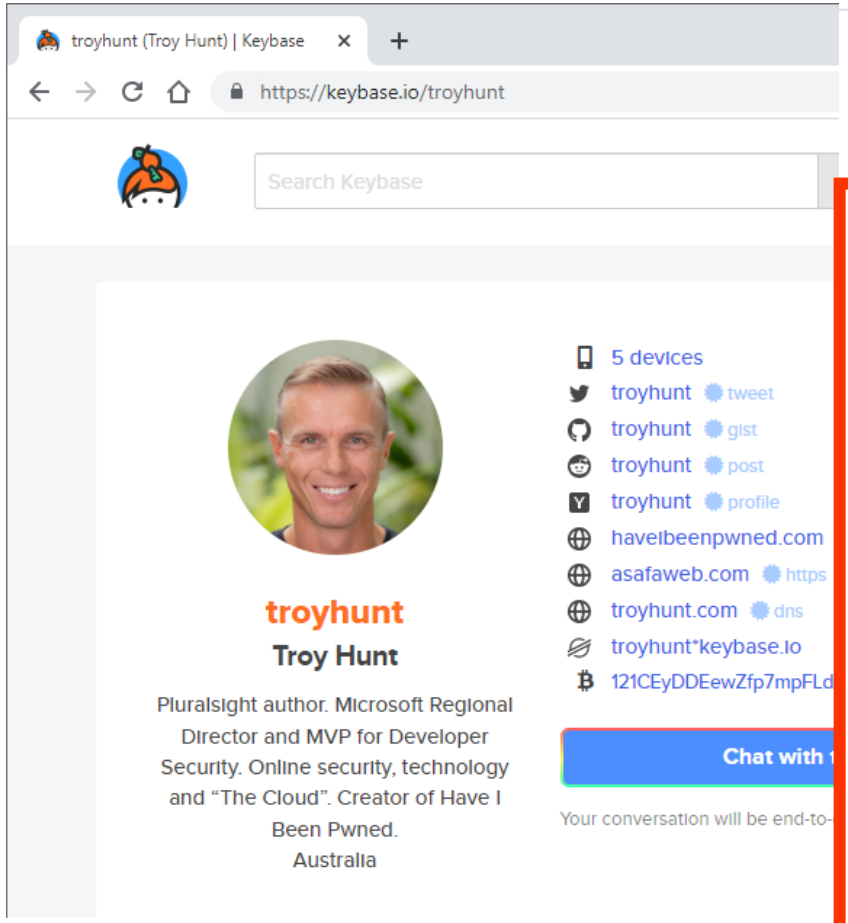
Example #1: Metadata in Image

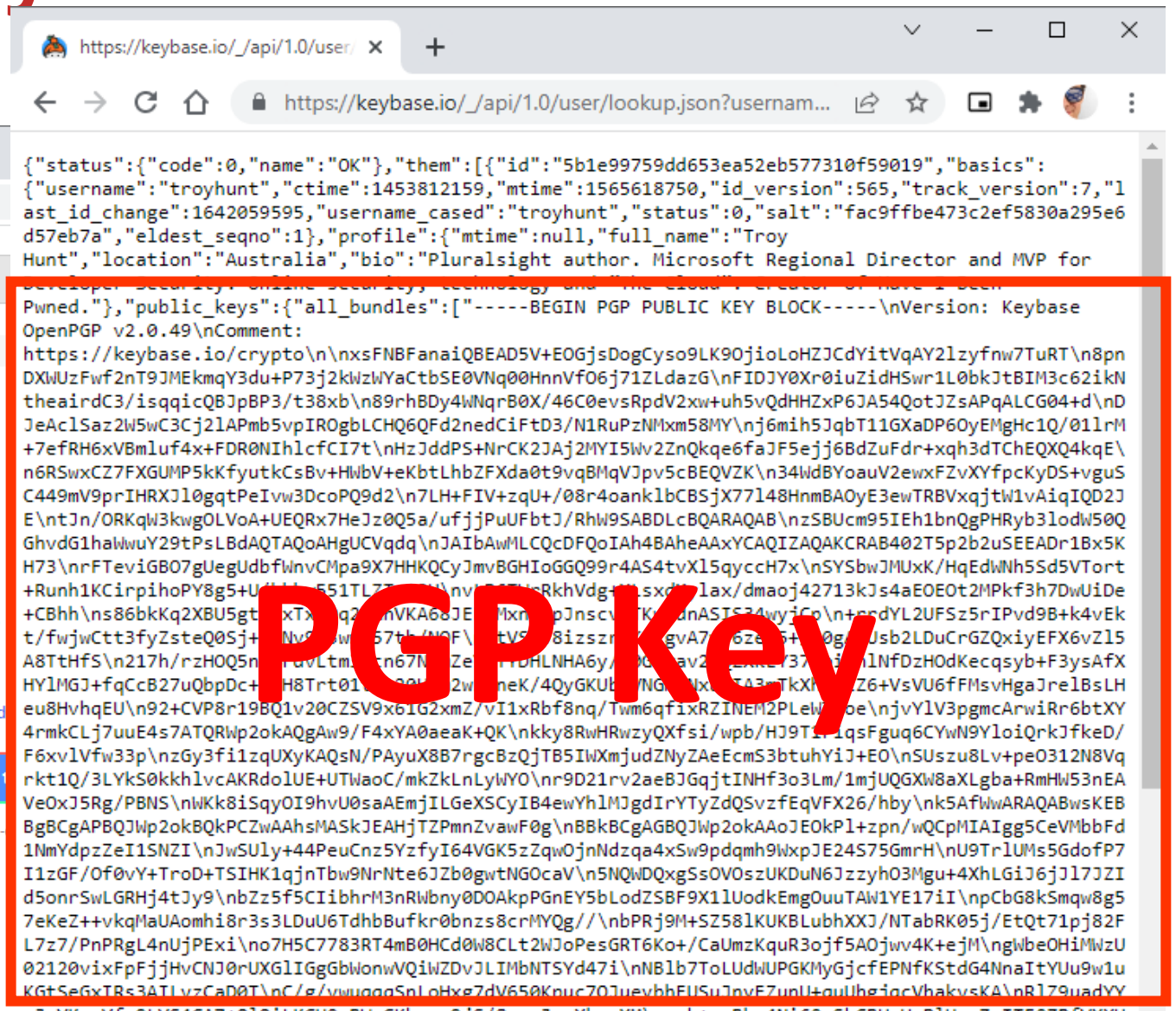
Potential information

- **Geolocation**
- Image path
- Software version



Example #2: Keybase



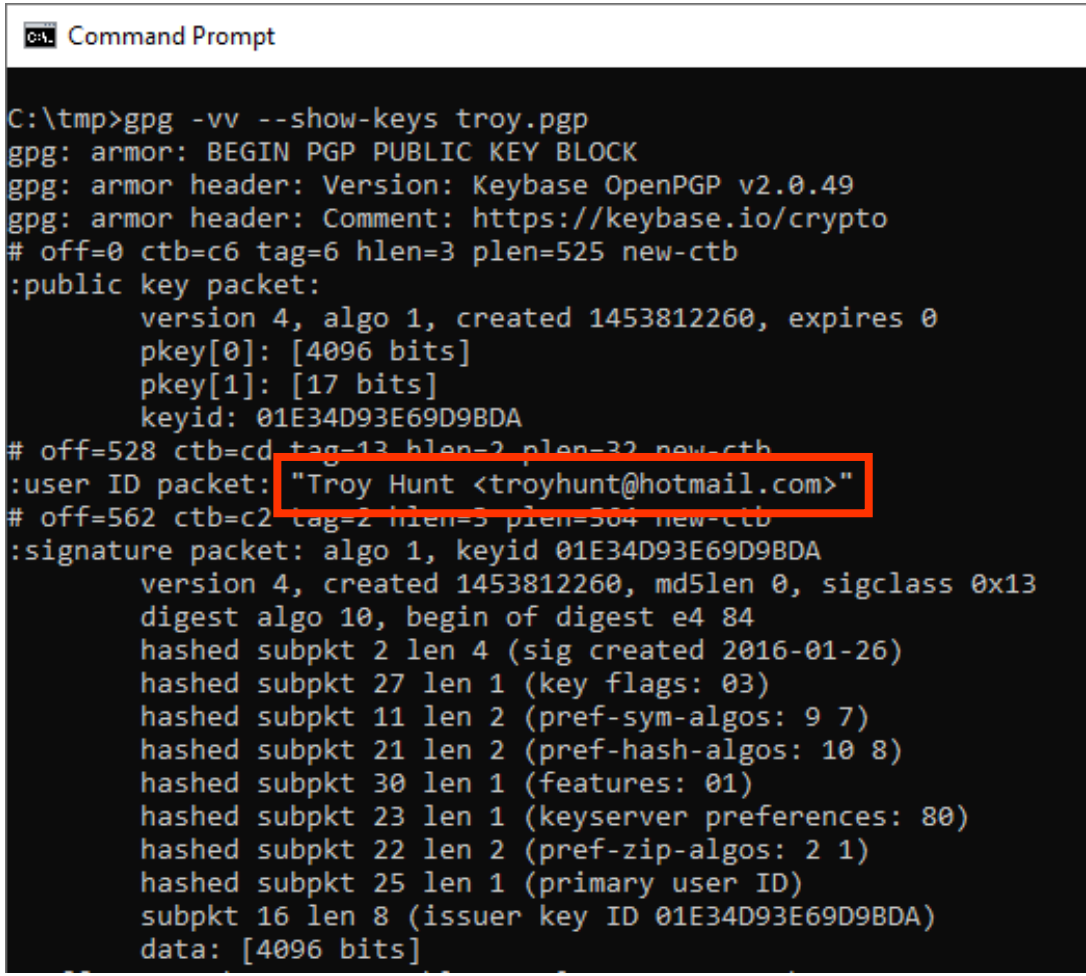


PGPKey

Example #2: Keybase

PGP Key include:

- Header metadata (Versions + Comment)
- Key information
  - Algorithm
  - Issuer
  - Key value
  - …
- User ID (Usually the user email)



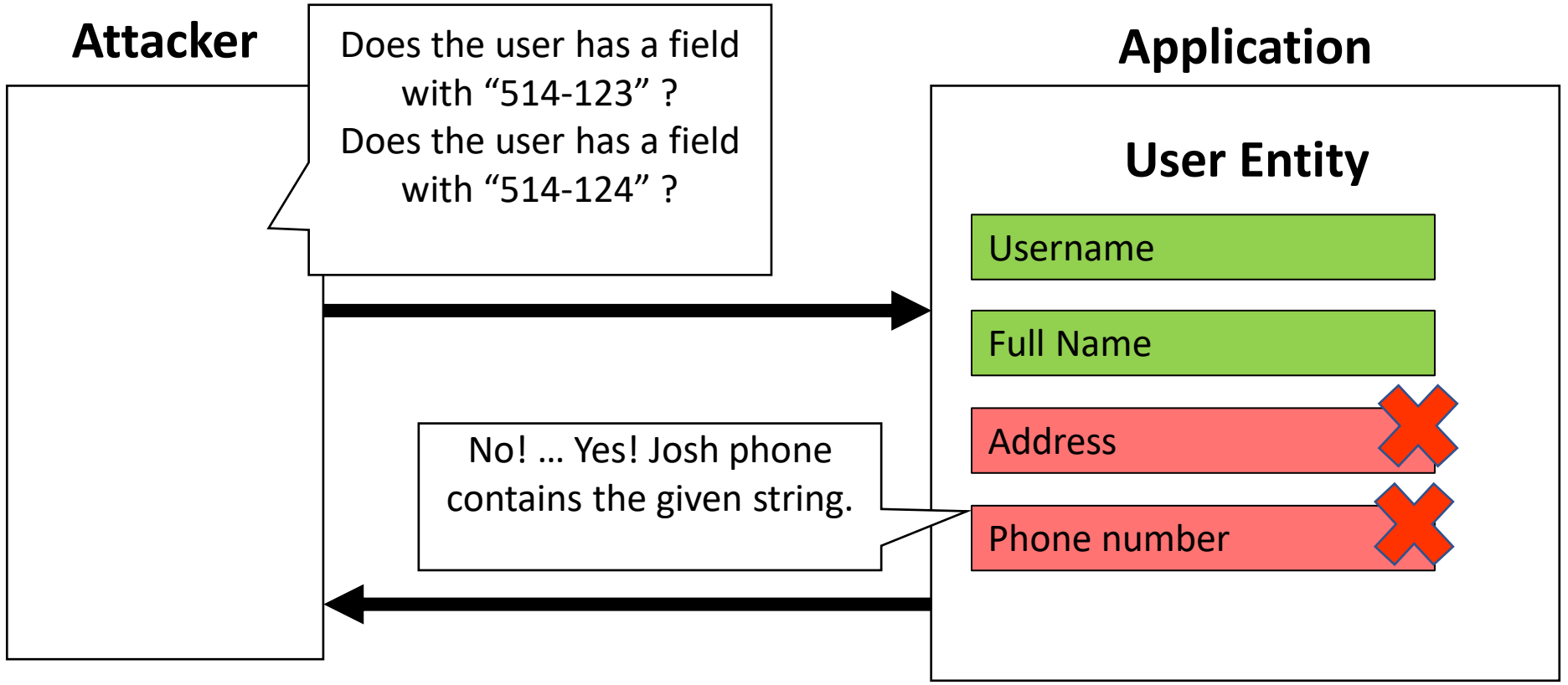
Recommendations

- Identify documents or files that are likely to contain metadata
- Strip metadata from images upon upload
  - Unless your service is a cloud hosting provider

Search features

Search features

Private information **might not be shown** on your application but, could **be queried** through search features.

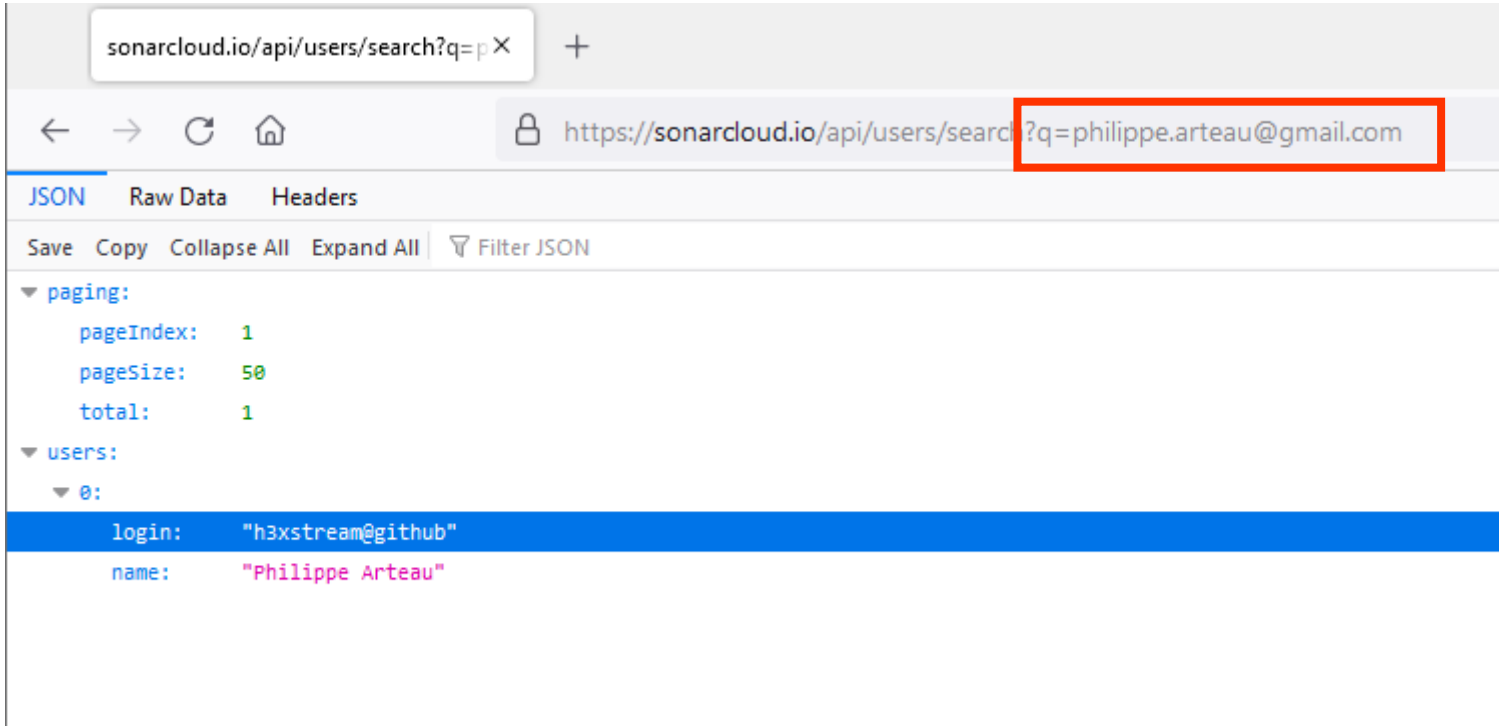


Example #1 SonarQube / SonarCloud

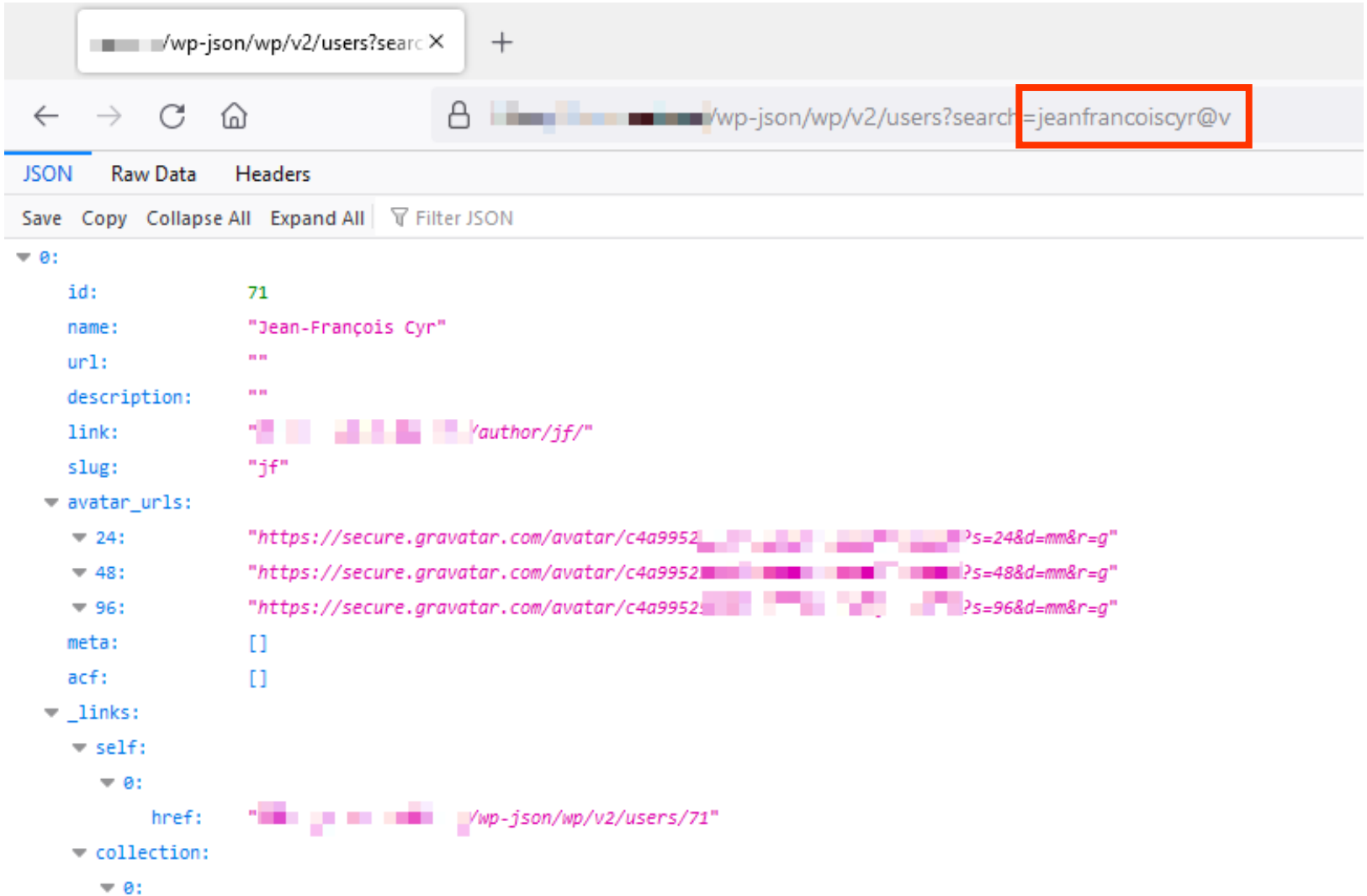
GET api/users/search since 3.6

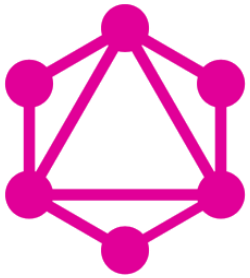
Get a list of active users.
The following fields are only returned when user has Administer System permission or for logged-in in user :

- 'email'
- 'externalIdentity'
- 'externalProvider'
- 'groups'
- 'lastConnectionDate'
- 'tokensCount'



Example #2 : WordPress users endpoint





GraphQL

(and REST endpoints)

REST Data binding

```
@Entity
public class UserEntity {

   @Id
   private Long id;
   private String username;
   private String fullName;
   private String address;
   private String ip;

[…]
```

```
@Controller
public class UsersController {

        @RequestMapping("/users")
        public UserEntity getU(@RequestParam(“id") String id) {
                return …
        }

[…]
```

GraphQL

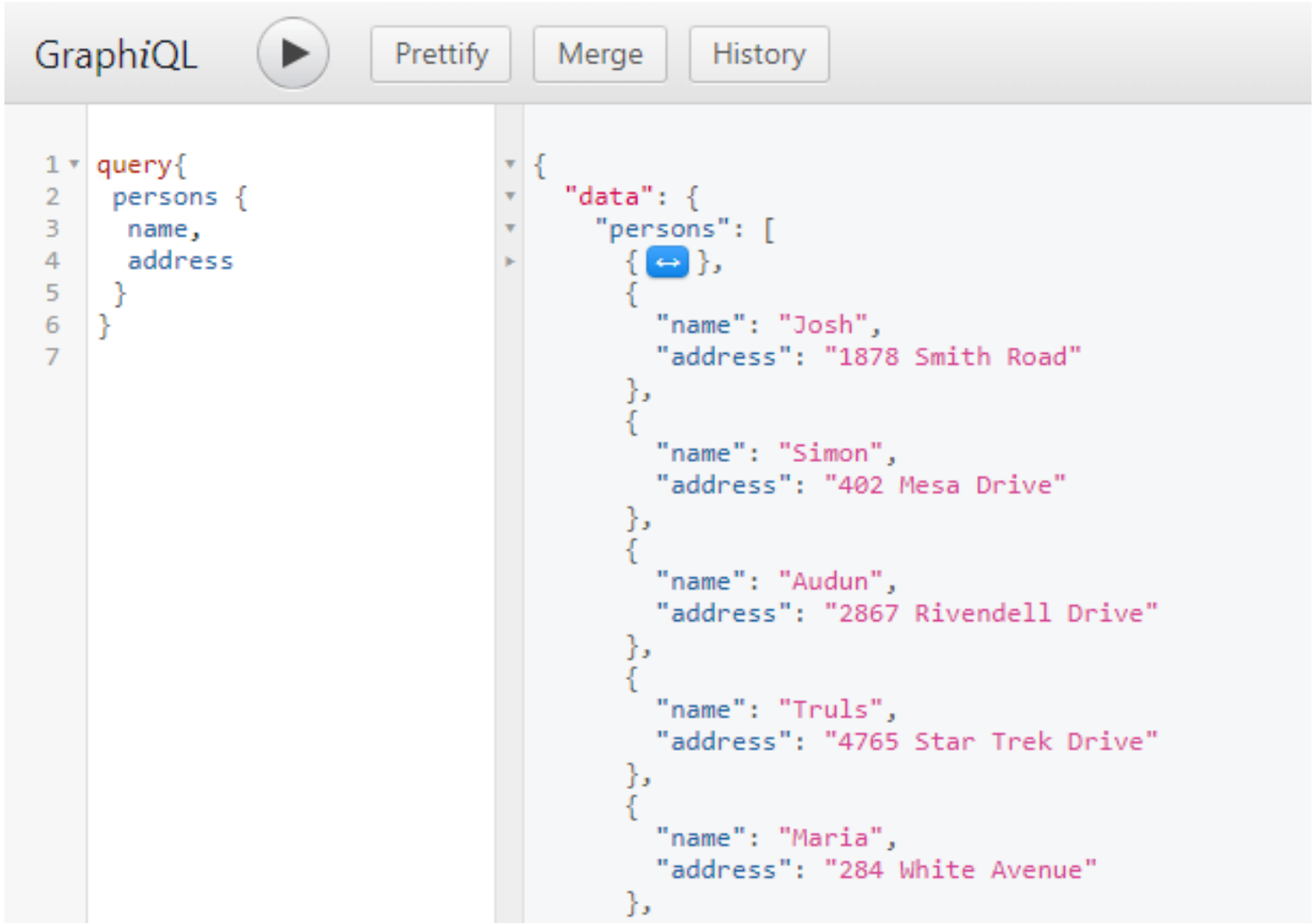
GraphiQL ▶ Prettify Merge History

```
{
  __type(name: "Person") {
    name
    fields {
      name
      type {
        name
        kind
      }
    }
  }
}
```

```
{
  "data": {
    "__type": {
      "name": "Person",
      "fields": [
        {
          "name": "id",
          "type": { ↔ }
        },
        {
          "name": "name",
          "type": {
            "name": null,
            "kind": "NON_NULL"
          }
        },
        {
          "name": "address",
          "type": {
            "name": "String",
            "kind": "SCALAR"
          }
        },
        {
          "name": "talks",
          "type": {
            "name": null,
            "kind": "LIST"
          }
        }
      ]
    }
  }
}
```

More info: https://graphql.org/learn/introspection/

GraphQL


GraphiQL
Prettify
Merge
History
query{
persons {
name,
address
}
}
{
"data": {
"persons": [
{ ↔ },
{
"name": "Josh",
"address": "1878 Smith Road"
},
{
"name": "Simon",
"address": "402 Mesa Drive"
},
{
"name": "Audun",
"address": "2867 Rivendell Drive"
},
{
"name": "Truls",
"address": "4765 Star Trek Drive"
},
{
"name": "Maria",
"address": "284 White Avenue"
},

Recommendations

- Avoid binding of persistence class directly to endpoints
  - Prefer separate data class
- Audit public classes for unexpected private or sensitive information
  - Address, IP, password, third-party account disclosure (FB, Google, Github), …
- Look for search API that could be queried with private information
- Reconsider complete users enumeration as REST/GraphQL endpoint?

Side-channel attacks

Side-channel attacks

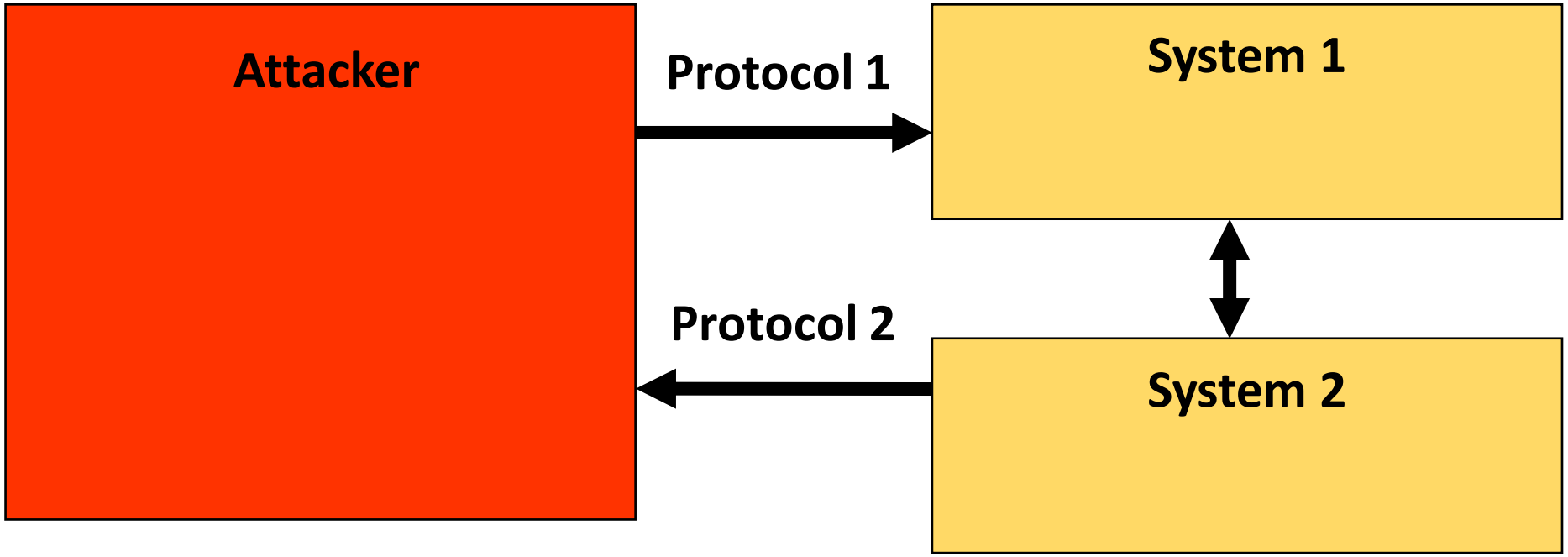
- A side-channel attack is any attack based on **information gained from the implementation**.
- **Another channel** can provide an **extra source of information**.

- Channel usually refer to
  - Timing information
  - Power consumption
  - Electromagnetic leaks
  - Sound
- Here I'll be referring to different **systems** or **protocols**.

« Side-channel » leaks

Get information on specific user using a private PII.

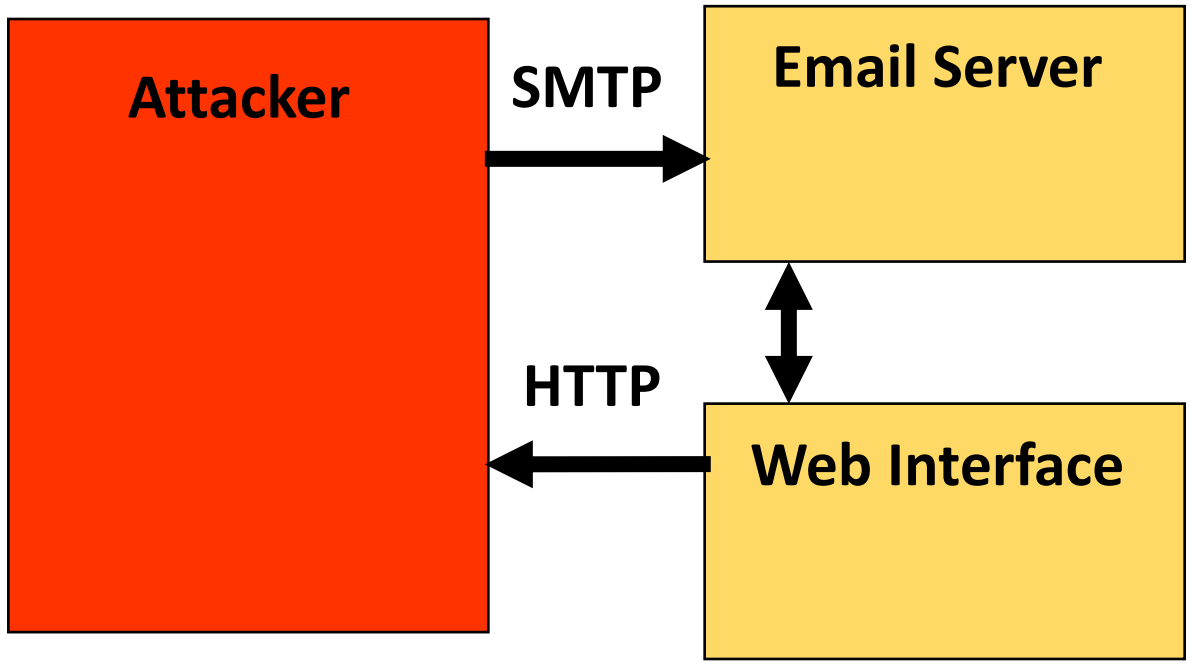
The attacker will succeed by **sending a user identifier** and **receiving details on another channel** about this user.



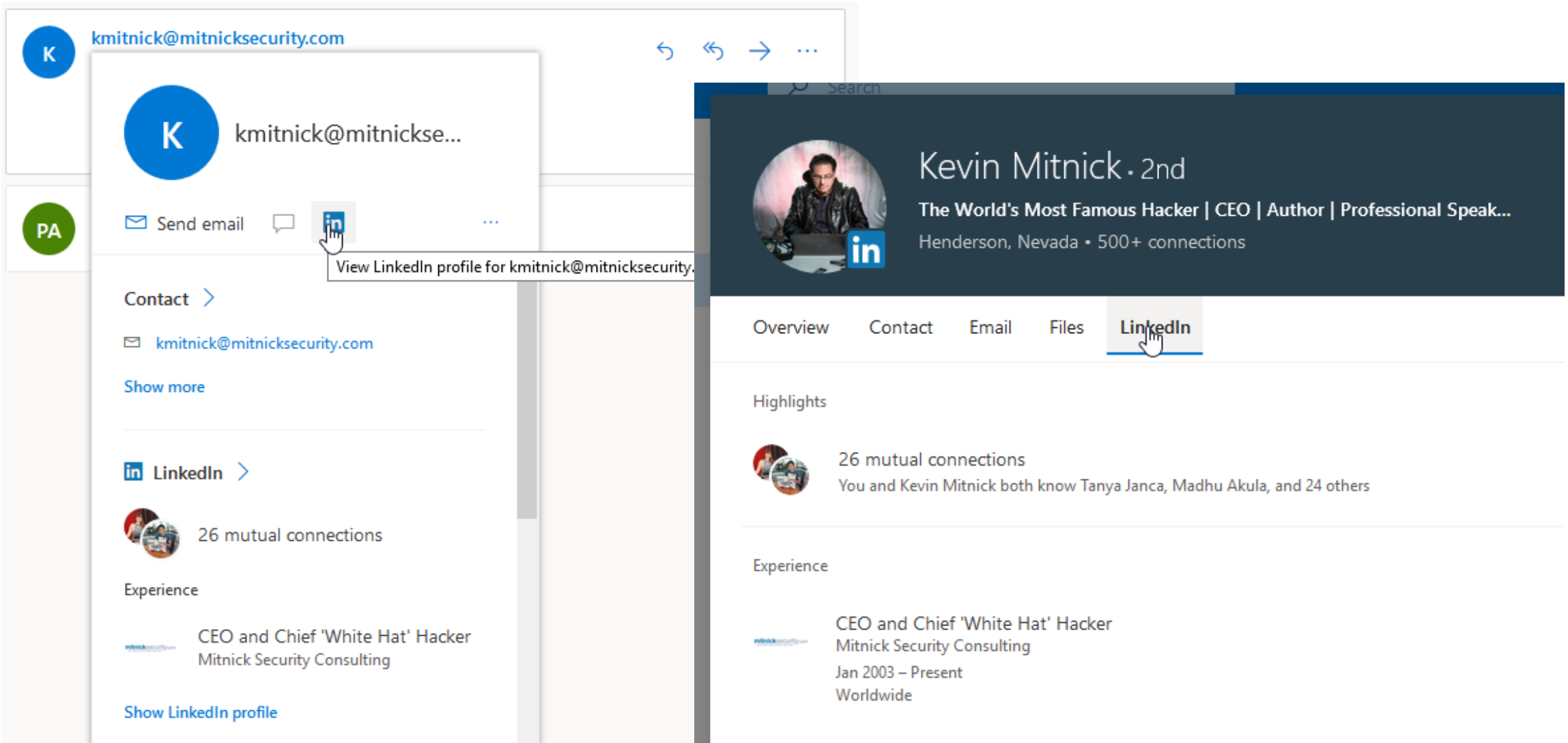
Example #1: Web Mail Client

1. The attacker imports spoofed emails through either SMTP. (Some provider allow pull from external mailbox in POP3 or IMAP)

2. The attacker view the crafted email through the web interface to see if the identify were enriched with additional information. (Full name, organisation, …)

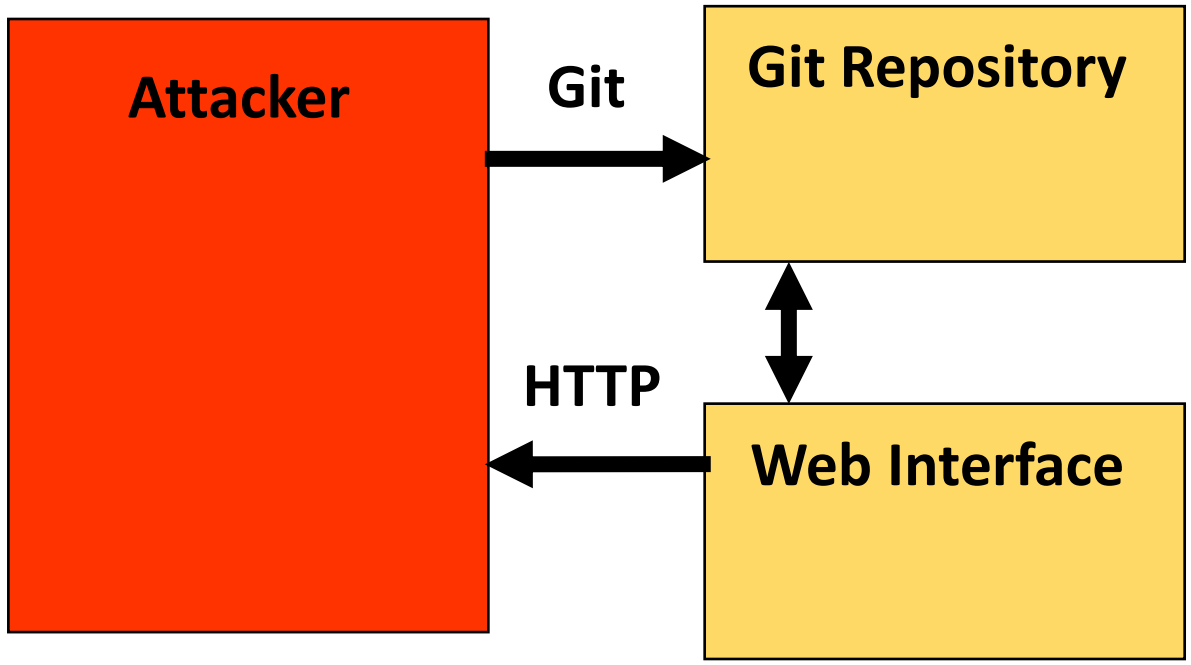


Example #1: Web Mail Client (Outlook)



Example #2: Git repository

1. The attacker imports commit with spoof emails.
2. The attacker view the commits through the web interface to see if the identify were enriched with additional information. (username, full name, organisation)



Conclusion

Recommendations

On your next threat modelling…

- Identify API that could **return** sensitive information.
- Identify API that could be **queried** with sensitive information
- Identify third party integrations (ie: Facebook Login, Google Account, Gravatar, etc.)
- See if the risks matter for your application and context

In your code…

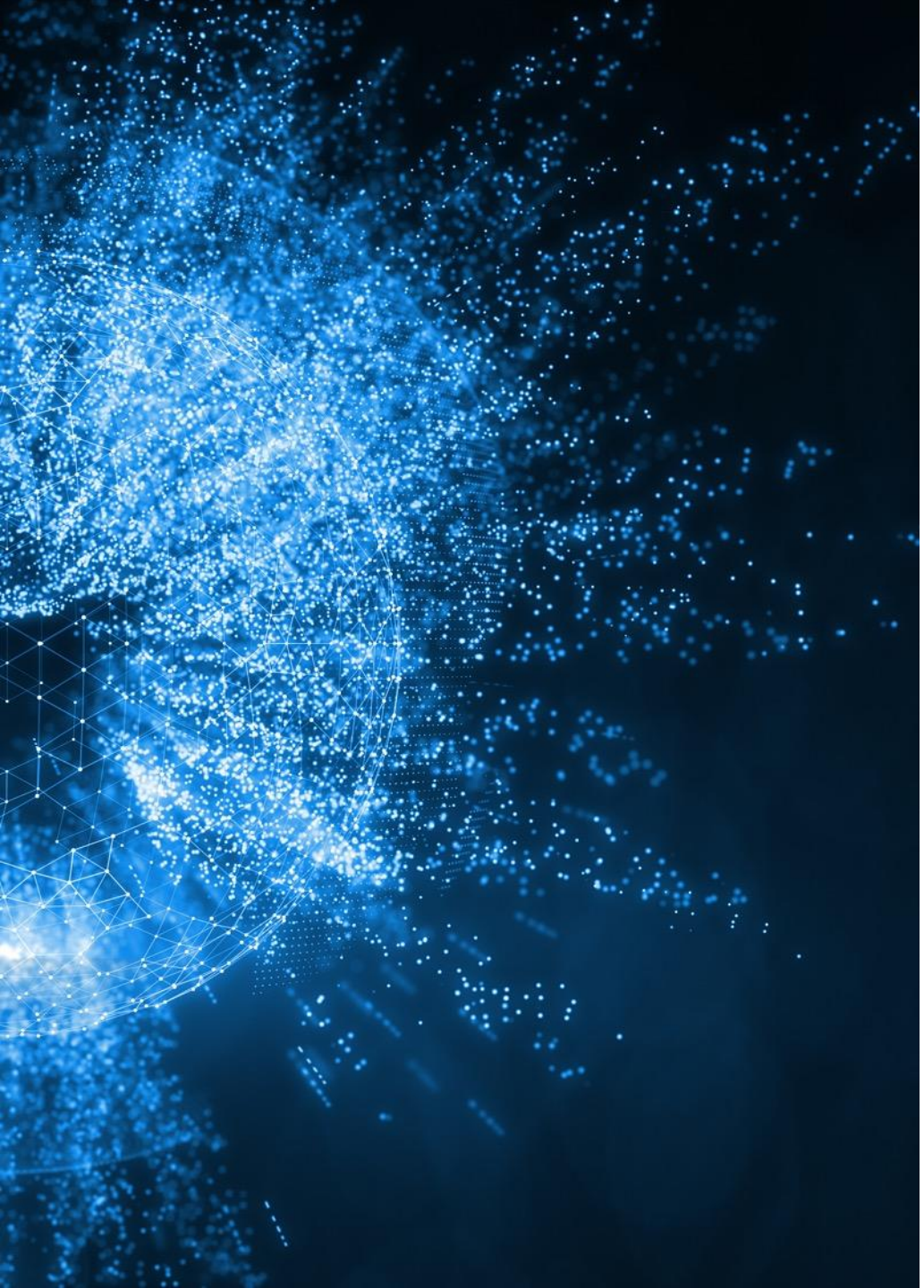
- Avoid security by obscurity (base64, unsalted hash)

Forewarned is forearmed

Users that are aware of those leakage are more likely to take precautions.

Examples:

- Knowing that your FB account will be linked to multiple websites, avoid publishing sentive information such as your address, affiliations or even full name.
- Create separate account to avoid identity correlation. (Creating multiples Gravatar/Github/Bitbucket account rather than one.)



Questions?

- parteau@gosecure.ca
- @GoSecure_Inc
- @h3xStream

https://www.gosecure.net/blog

References

Blogs/tools on similar topics

- Gravatar risk assessment on Wordpress.com/Gravatar.com
  - https://blog.h3xstream.com/2021/03/emails-disclosure-on-wordpress.html
- Hardening tests for Wordpress related to Gravatar
  - https://blog.h3xstream.com/2021/03/6-ways-to-enumerate-wordpress-users.html
- LinkedIn deanonymization example
  - https://blog.h3xstream.com/2021/04/deanonymizing-linkedin-users.html
- Ghunt: Google account investigation tool
  - https://github.com/mxrch/GHunt

Privacy acts around the world

- GDPR (EU)
  - https://gdpr-info.eu/art-5-gdpr/
- CCPA (California)
  - https://oag.ca.gov/privacy/ccpa
- Loi sur la Protection des Renseignements Peronnels (Québec)
  - https://www.legisquebec.gouv.qc.ca/fr/document/lc/P-39.1