



OWASP

Open Web Application
Security Project

Why AppSec Matters?

Aldo Salas

aldo.salas@owasp.org

Agenda

- Intro.
- Current status of AppSec in the industry.
- Case study.
- Why OWASP matters.



About me

- 10+ years of experience in AppSec.
- Currently working for Fortune 500 Company.
- Independent researcher in free time (bug bounty).
- Chapter Leader for Aguascalientes.
- Favorite vulnerability: SQL Injection.
- Proud U.A.A. alumnus.



I'm not here to scare you...

- Or
maybe
I am



OWASP
Open Web Application
Security Project

Another week, another hack

Data breach

- 3,000 Tidewater Community College workers victimized in W-2 scam
- Attacker compromises information of 250K in Bailey's data breach
- Karmanos Cancer Center patient info at risk from lost flash drive
- Stolen laptop exposes PII of over 200K Premier Healthcare patients
- Crooks Steal, Sell Verizon Enterprise Customer Data
- Thieves Phish Moneytree Employee Tax Data
- Seagate Phish Exposes All Employee W-2's
- Hacker breaches USA Cycling, personal information at risk
- 1-800 FLOWERS warns that hacker may have stolen customers' personal info
- Data breach puts southern expats' personal details online
- Hackers reportedly access OpSec employee data
- 18 million stolen IDs discovered on server / Criminals in China got illegal access
- JASACare notifies 1,154 patients of breach

And the list goes on:

<http://www.itgovernance.co.uk/blog/list-of-data-breaches-and-cyber-attacks-in-march-2016/>



OWASP
Open Web Application
Security Project

We are not doing a great job



OWASP
Open Web Application
Security Project

CONNECT.

LEARN.

GROW.

Real-life case study



OWASP
Open Web Application
Security Project

Background

- Third party used to collect festivals and new-hires information.
- The following email was sent to artists/managers/assistants:

review and make sure all of the information is up to date:

[public_key=27deeb3af140](#)

[37784098d02a](#)

['402672/edit?](#)

- First thought on “public_key”: Maybe it’s an authentication token, not ideal but still provides some level of authentication.



CONNECT.

LEARN.

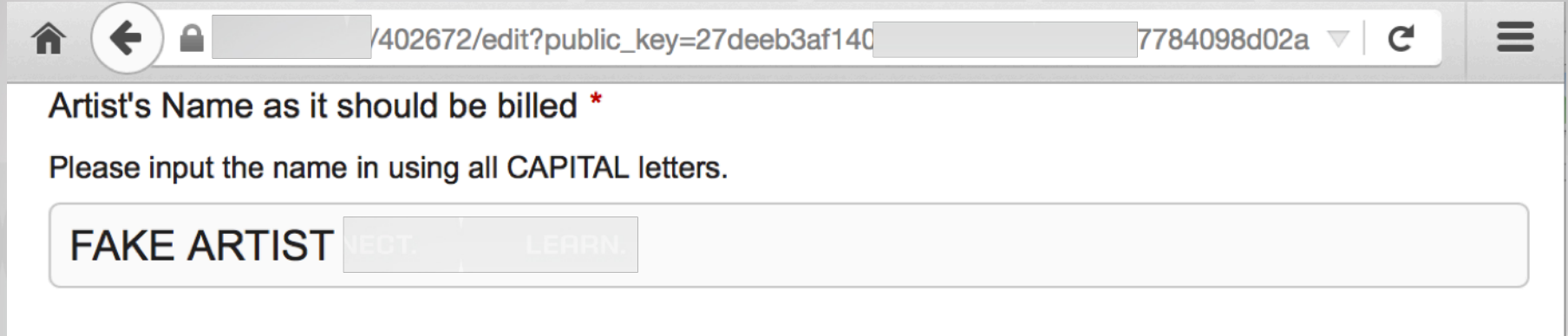
GROW.

Phase 1: Discovery



OWASP
Open Web Application
Security Project

Public key parameter:



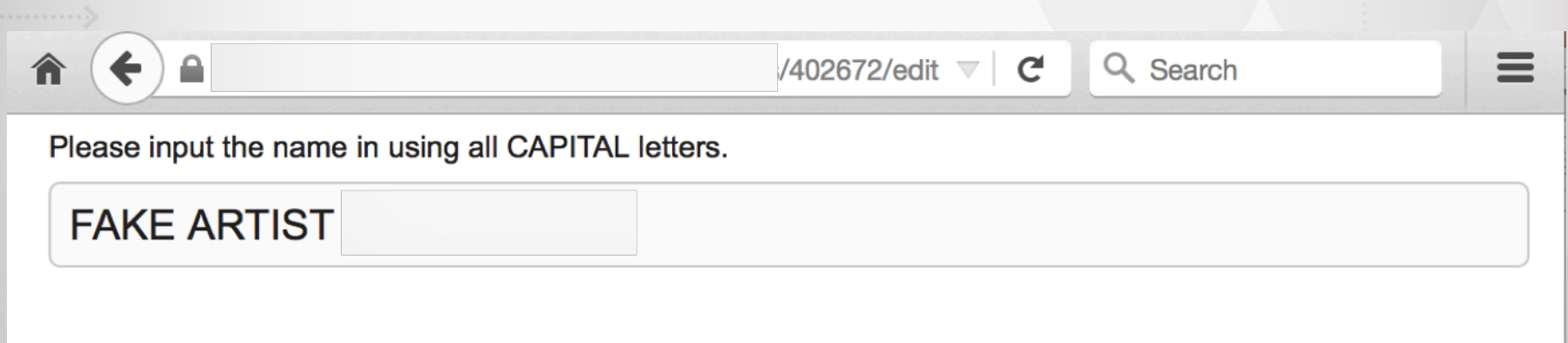
Artist's Name as it should be billed *

Please input the name in using all CAPITAL letters.

FAKE ARTIST

NEOT. LEARN.

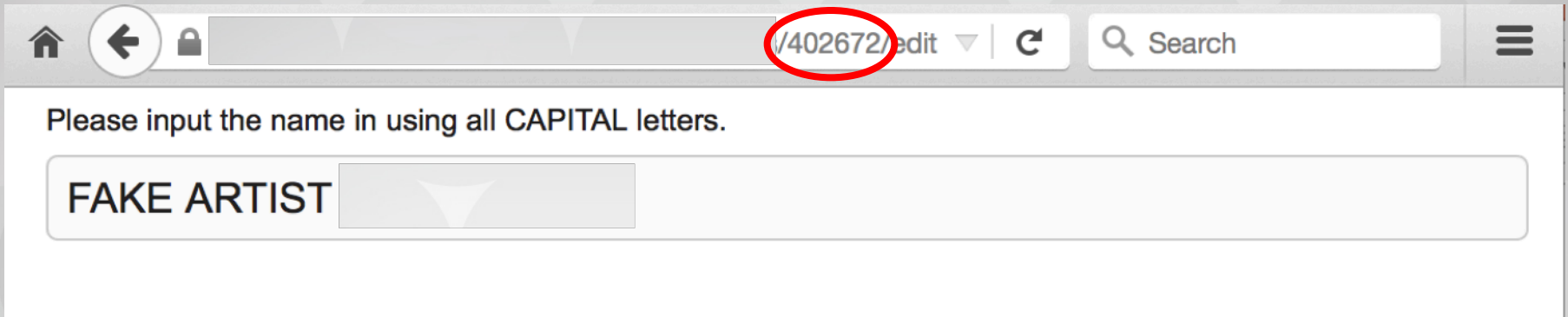
- Removing public_key = Unauthenticated Access To Data



Please input the name in using all CAPITAL letters.

FAKE ARTIST

Analyzing URL:



Home Back Lock /402672/edit Search

Please input the name in using all CAPITAL letters.

FAKE ARTIST

- Changing ID in URL = Insecure Direct Object Reference (Still unauthenticated)

Artist's Name as it should be billed *

Please input the name in using all CAPITAL letters.

Taylor Swift

Mobile

407

Analyzing page:

Please upload official materials for festival use. Uploaded files should be in raster format only **jpg, jpeg, gif or png.**

Artist Hi Res Photo

Upload File

Artist Hi Res Photo 

Artist Prod. Rider

Upload File

Riders 

Upload File

Artist Hi Res Photo 

Artist Prod. Rider

Upload File

Riders 

Official Band Logo

Upload File

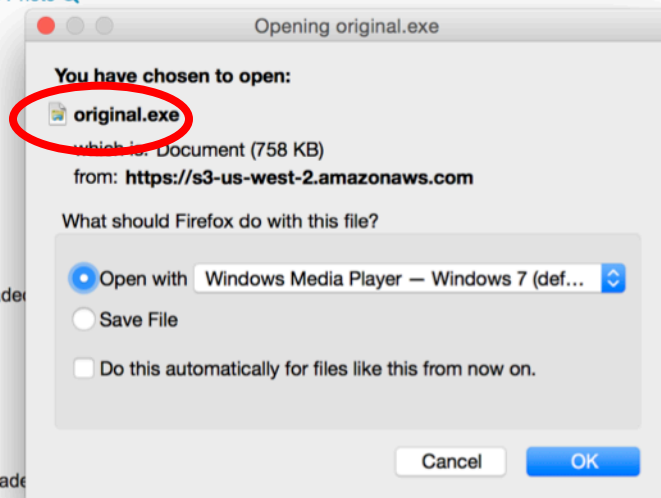
No file uploaded

Images of Live performance

Upload File

No file uploaded

Artist Bio



OWASP
Open Web Application
Security Project

Analyzing page:

```
ue='0' />  
id]' value='' />  
<a href="https://s3-us-west-2.amazonaws.com/[redacted]uploads_files/000/781/957/original.exe"
```

- Files are stored in AWS S3
- File is always renamed to original.ext
- Unauthenticated access to uploaded files as well.
- Bruteforcing of files is possible but not really needed.



Summary so far:

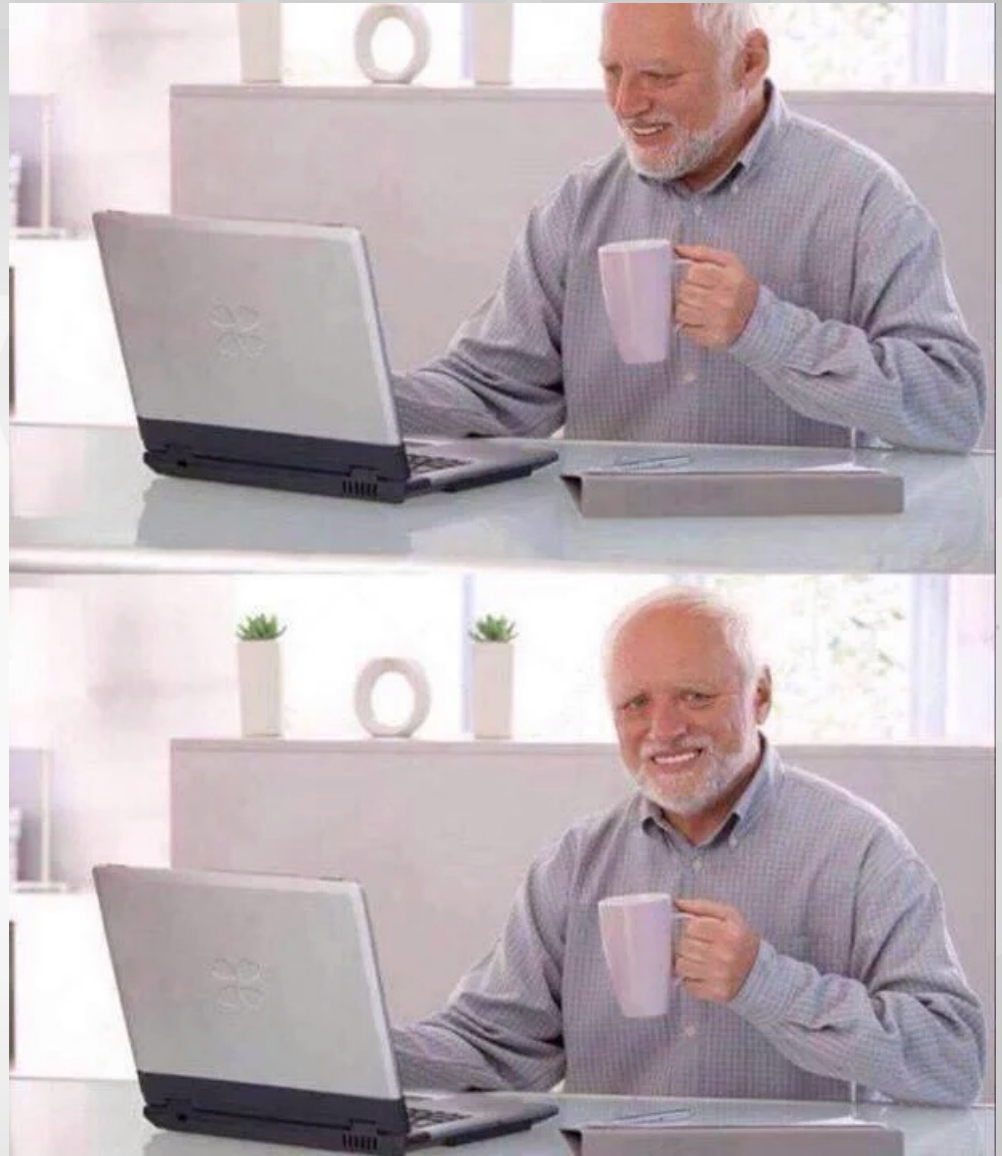
- Unauthenticated access to artist profile.
- Access to ANY profile is possible using Insecure Direct Object References.
- Unrestricted File Upload is possible.
- Unauthenticated Access to uploaded files is possible.



CONNECT.

LEARN.

Reminder:
this is a
single page.



OWASP
Open Web Application
Security Project

CONNECT.

LEARN.

GROW.

Phase 2: Automation



OWASP
Open Web Application
Security Project

Automating data retrieval to demonstrate risk

```
try:
    for num in range(403618,230000,-1):
        url = "
        /"+str(num)+"/edit"
        q.put(url)
    q.join()
except KeyboardInterrupt:
    sys.exit(1)
```

- Initial Results:
 - More than 80 thousand records found.
- Notes:
 - More than 170 thousand requests were sent.
 - More than 6GBs were downloaded.
 - I was never stopped nor even detected.



CONNECT.

LEARN.

GROW.

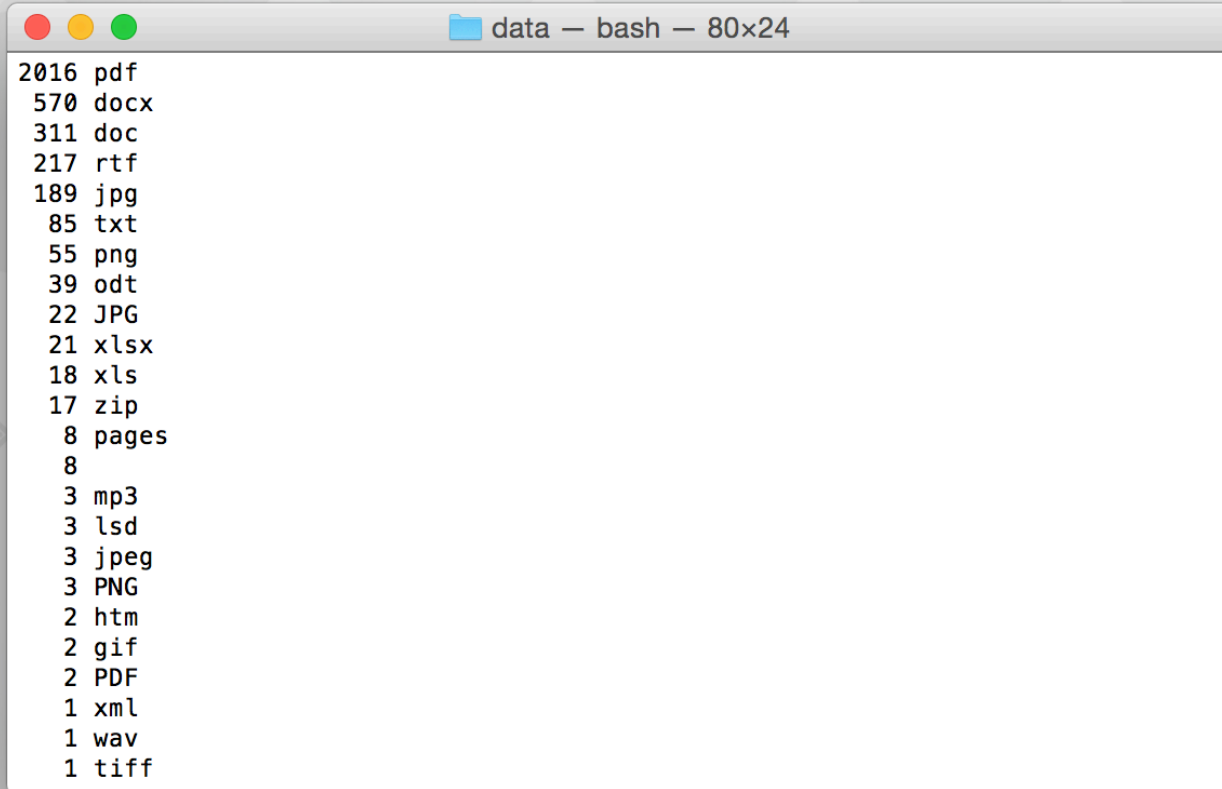
Phase 3: Parsing data



OWASP
Open Web Application
Security Project

Parsing data:

- Number Of Direct URLs To Download Files obtained:



A terminal window titled "data — bash — 80x24" displays a list of file extensions and their corresponding counts. The data is as follows:

Count	Extension
2016	pdf
570	docx
311	doc
217	rtf
189	jpg
85	txt
55	png
39	odt
22	JPG
21	xlsx
18	xls
17	zip
8	pages
8	
3	mp3
3	lsd
3	jpeg
3	PNG
2	htm
2	gif
2	PDF
1	xml
1	wav
1	tiff

Parsing data:

- Artists PII including emails and phones

CONNECT.

LEARN.

GROW.



OWASP
Open Web Application
Security Project

Parsing data:

- And much, much more:

CONNECT.

LEARN.

GROW.



OWASP

Open Web Application
Security Project

Possible outcomes if exploited by attackers:

- Headlines in the news:
 - “HUNDRES OF THOUSANDS ARTISTS DETAILS LEAKED BY COMPANY”
 - “WANT TAYLOR SWIFT’S NUMBER? WE’VE GOT IT”
- Attackers selling or leaking artists information (stalkers, curious people, etc.)
- Fraud and potential legal consequences (SSNs involved).
- Phishing campaigns against retrieved emails.
- Etc., etc., etc.



What could've been done better:

- Prevent unauthenticated access to the page.
- Once authentication has been implemented, perform authorization checks.
- Validate at server-side the uploaded files.
- Also add authentication checks to the files.
- Logging and IPS/IDS configuration to detect unusual activity.



Tools used for discovery and exploit:

- Python programming language to code small download script.
- Standard Unix tools to parse data (find, cat, cut, grep, sort, sed, ls, wget).



CONNECT.

LEARN.

GROW.

Questions?



OWASP

Open Web Application
Security Project

Why OWASP Matters?

- All the vulnerabilities shown in this presentation could've been avoided by following OWASP recommendations.



OWASP
Open Web Application
Security Project

OWASP TOP 10 misses:

- A1 – Injection
- A2 – Broken Authentication and Session Management
- A4 – Insecure Direct Object References

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project



CONNECT.

LEARN.

GROW.

Questions?



OWASP
Open Web Application
Security Project

CONNECT.

LEARN.

GROW.

Thanks!

aldo.salas@owasp.org



OWASP
Open Web Application
Security Project