Cheatsheets / **Common Attacks on Web Applications**

Preventing Cross-Site Scripting (XSS) Attacks

Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS) is a vulnerability that occurs when a web application returns unsanitized input to the front end of an application.
Three types of XSS attacks are:

- *Stored XSS*: when a server saves an attacker's input into its datastores.
- *Reflected XSS*: when a user's input is immediately returned back to the user.
- *DOM-Based XSS*: when user input is interpreted by the DOM, an attacker could inject arbitrary code.

The code shows examples of HTML tags that help attackers inject dangerous input.

```
<script>alert(1);</script>

<img src="X" onerror=alert(1);>

<b onmouseover=alert(1)>click me!</b>

<body onload=alert('test1')>
```

Preventing Cross-Site Scripting

XSS can be mitigated by properly sanitizing input, as well as using specialized functions. We can generally succeed in preventing XSS attacks by removing potentially dangerous keywords or potentially dangerous characters such as:

- `<`
- `>`
- `"`
- `=`

Rather than remove characters, we could replace them with the HTML-encoded versions. For example, the `<` character would be converted to the "<" string.

Cross-Site-Scripting (XSS)

Cross-Site Scripting (XSS) is a part of the OWASP Top Ten. XSS is when an application allows untrusted data, potentially user-supplied data, into a web page without proper validation or sanitization.

It’s dangerous because it can allow attackers to execute malicious scripts in a victim’s browser leading to hijacked sessions, or malicious page alterations or redirections.

The code is an example of some code that may be used as part of a XSS attack. It could be inserted into a URL.

```
<script>alert(1);</script>
```