SECURE
CODE
WARRIOR

XML EXTERNAL ENTITY INJECTION (XXE)

WHAT IS XXE INJECTION?

XXE injection can be used on web applications that parse XML input

An attacker able to submit XML, can make use of references to external entities.

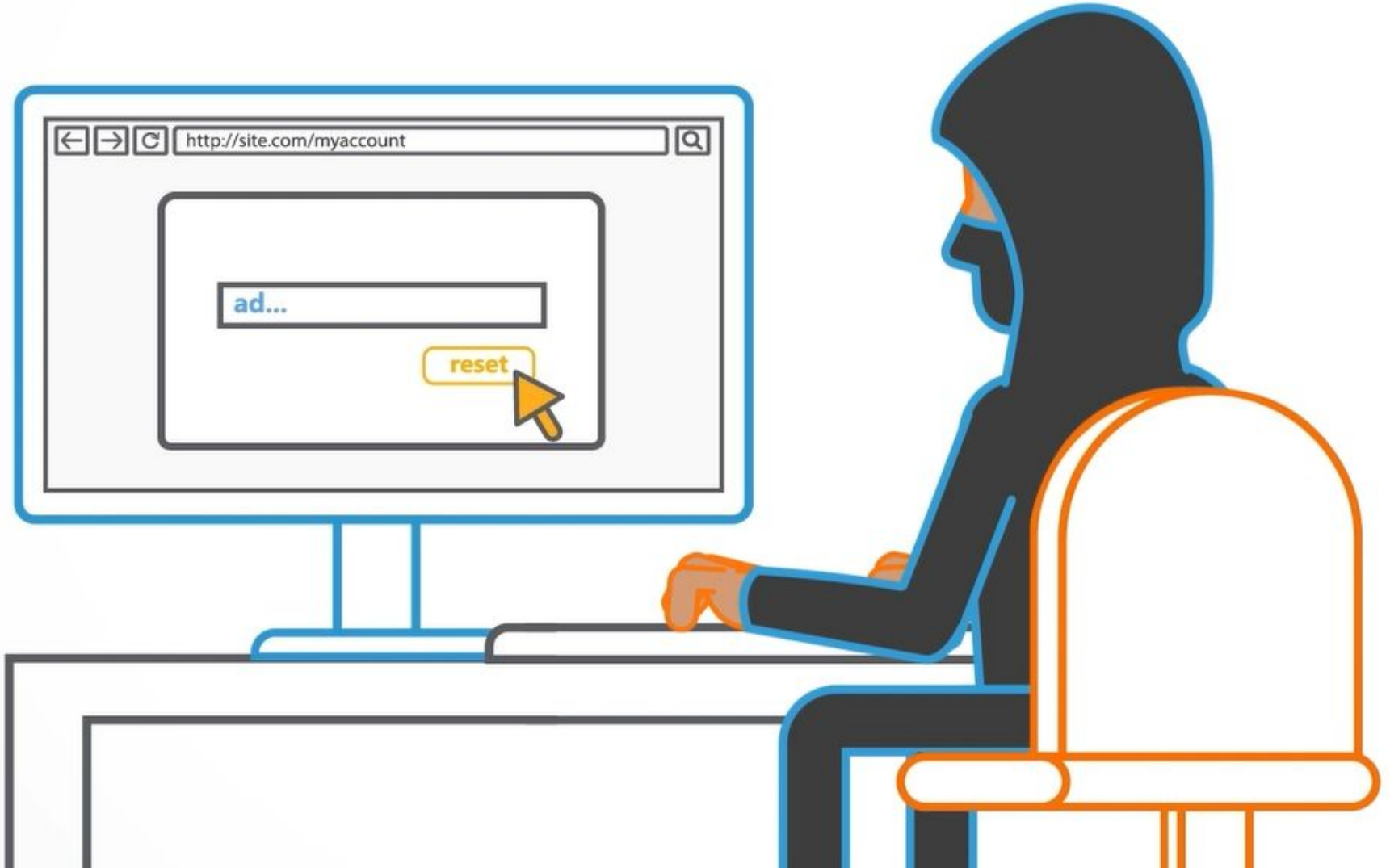
```
POST /forgot_pass HTTP/1.1
Host : Shop.localstore.com
User-Agent:Mozilla/5.0
Accept-Language:en-US
Connection:Keep-alive

<forgot_pass>
<user>admin@localstore.com</user>
</forgot_pass>
```

```
POST /forgot_pass HTTP/1.1
Host : Shop.localstore.com
User-Agent:Mozilla/5.0
Accept-Language:en-US
Connection:Keep-alive

<?xml version="1.0">
<!DOCTYPE forgot_pass[
<!ELEMENT field ANY>
<!ENTITY testxxe SYSTEM "file:///etc/passwd">
]>

<forgot_pass>
<user>
<field name="id">&textxxe;<field>
</user>
</forgot_pass>
```

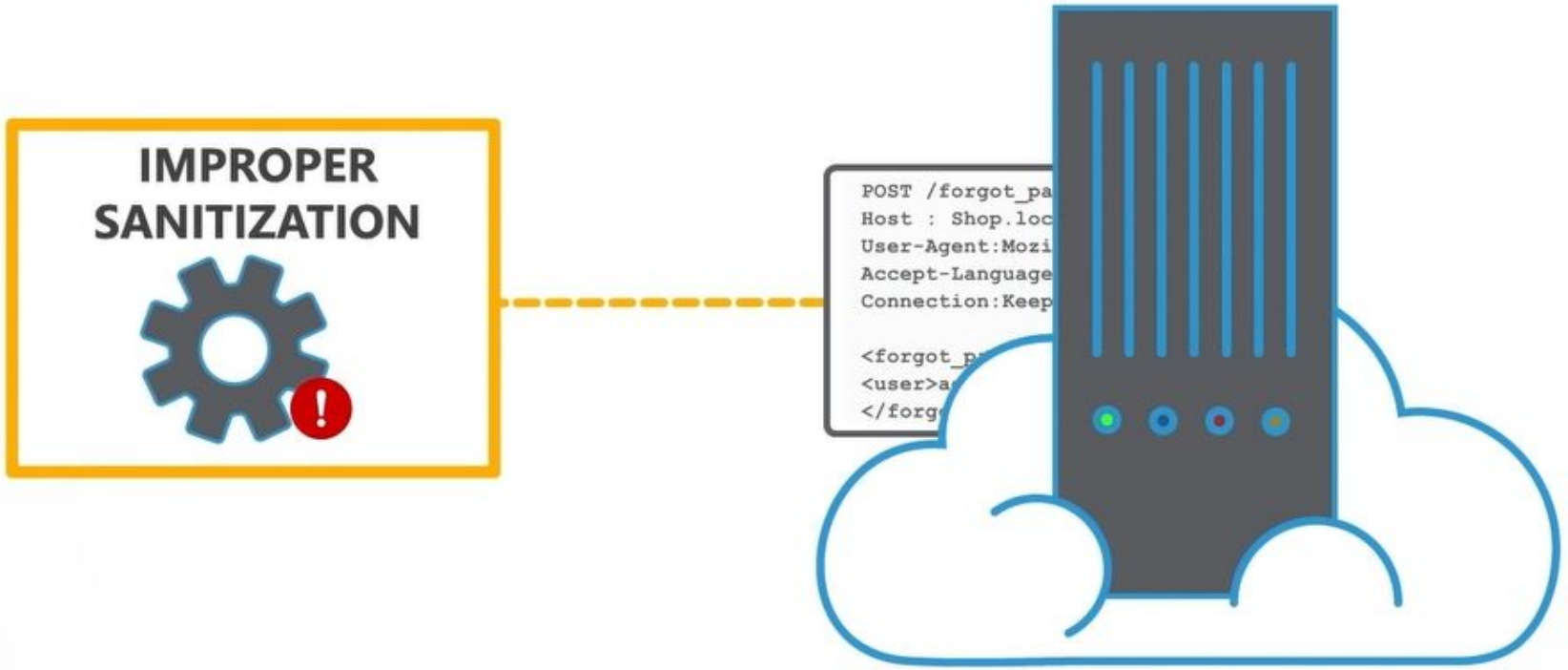
The attack occurs when the XML is processed by the server.

WHAT CAUSES XXE INJECTION?

XXE injections occur when improperly sanitized input is being processed



by an XML parser with default or weakly configured settings.

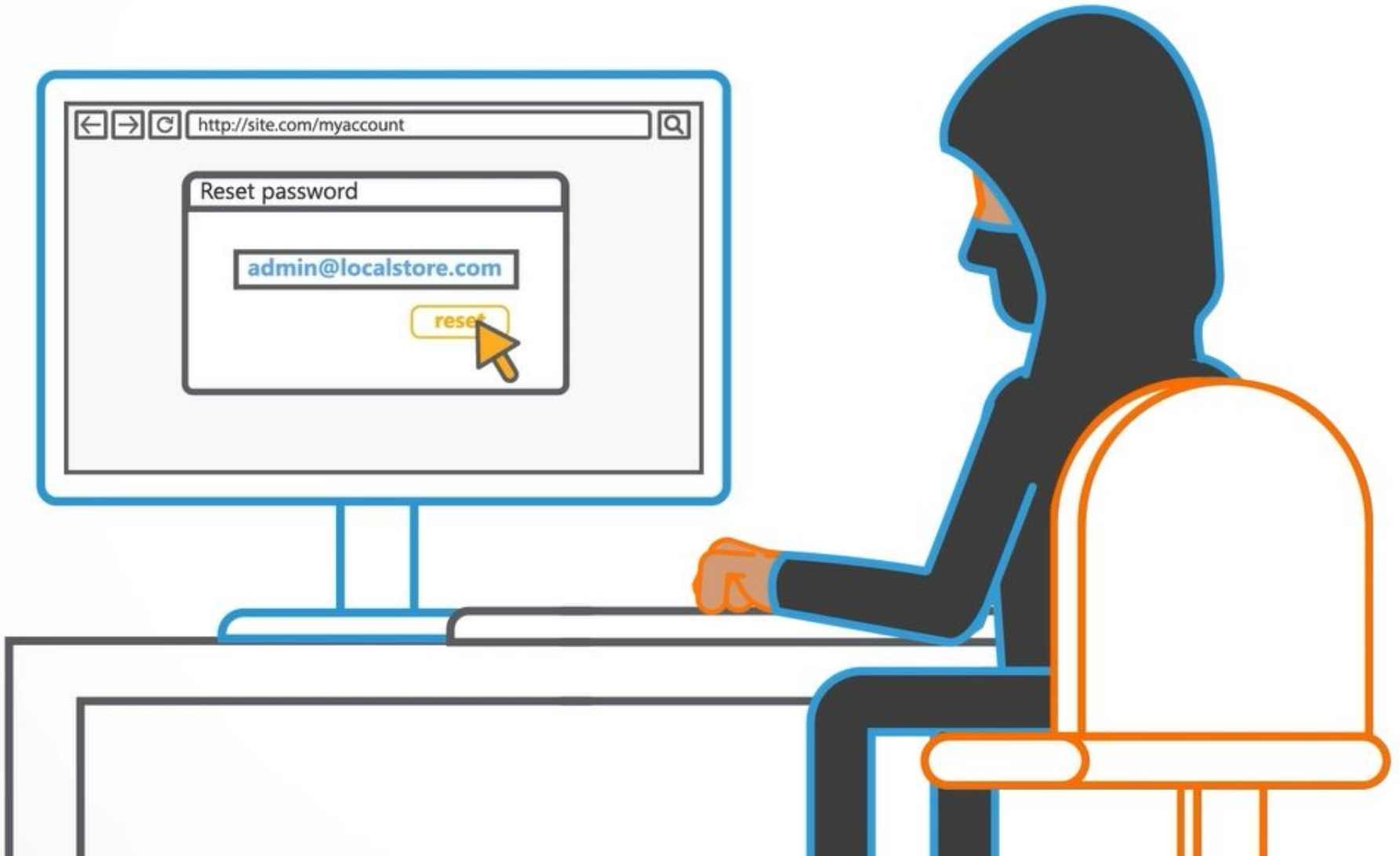


To understand

XXE Injection vulnerabilities,

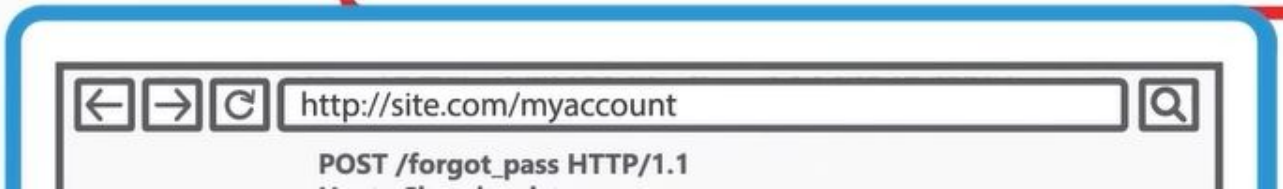
let's look at an example.

A website uses XML to submit data for its reset password functionality.



An attacker intercepts the XML request and changes it in order to access arbitrary files on the server.

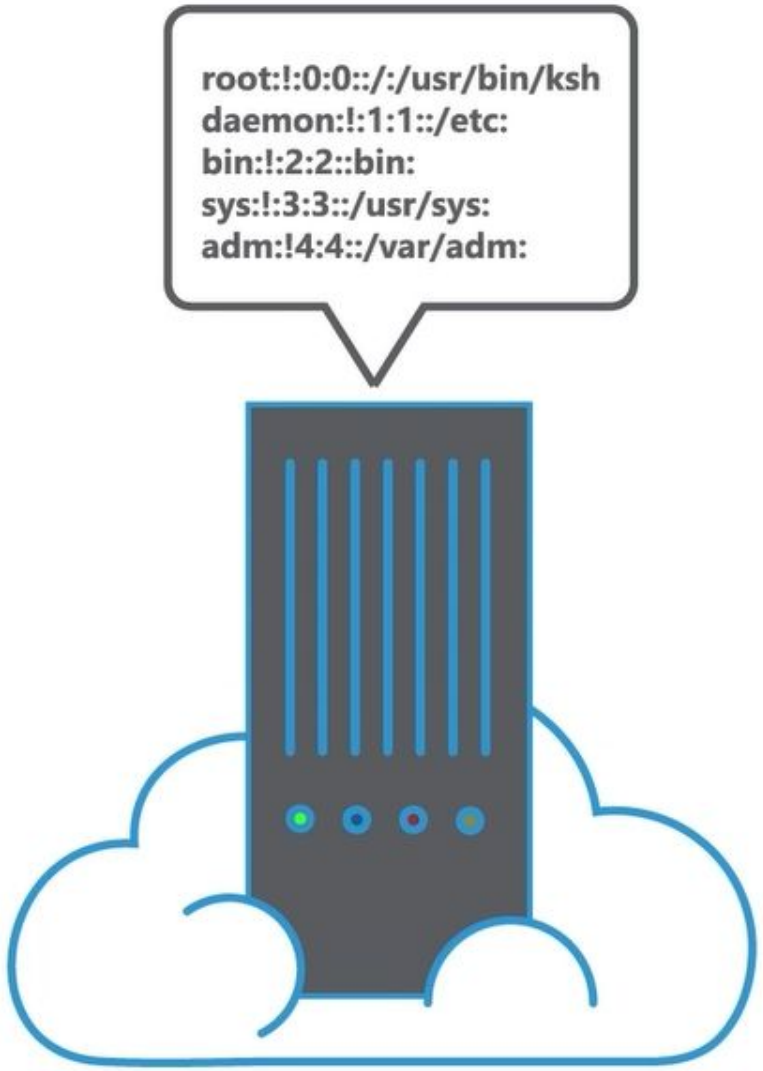
```
POST /forgot_pass HTTP/1.1
Host : Shop.localstore.com
User-Agent:Mozilla/5.0
Accept-Language:en-US
Connection:Keep-alive

<?xml version="1.0"?>
<!DOCTYPE forgot_pass [
<!ELEMENT field ANY>
<!ENTITY testxxe SYSTEM "file:///etc/passwd">
]>

<forgot_pass>
<user>
<field name="id">&textxxe;<field>
</user>
</forgot_pass>
```



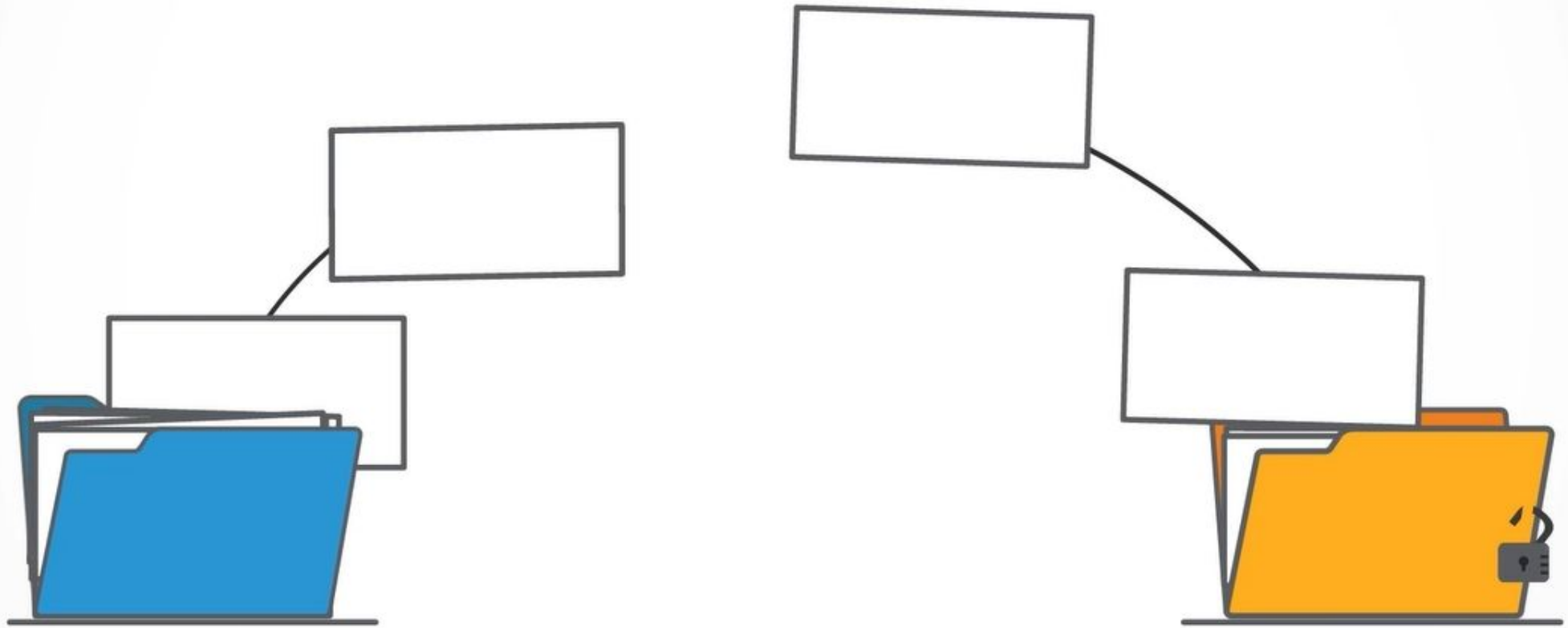
The web server parses the XML and returns the /etc/passwd file.

XXE INJECTION VULNERABILITIES CAN HAVE SIGNIFICANT IMPACT.

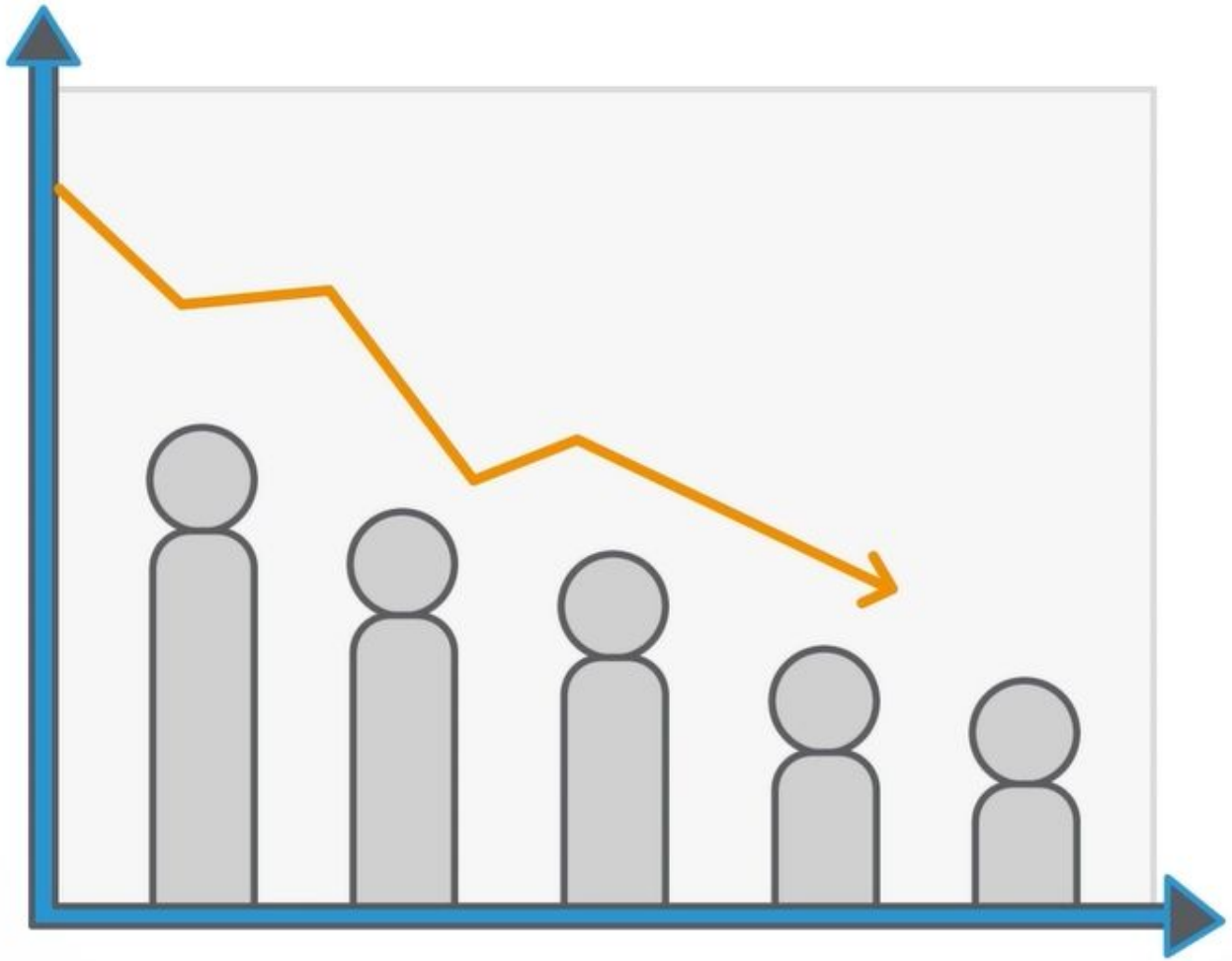
XXE could lead to an attacker performing internal port scanning and mapping your entire network.



An attacker could retrieve documents from the web server, resulting in compromised data.



And system unavailability could cause revenue and reputation loss.



To prevent XXE Injection

- Developers should apply application-wide filters or sanitization on all user-provided input. Consider GET and POST parameters, Cookies and other HTTP headers
- Always apply allowlist input validation.
- And, XML parses should disable support for external entities, or DTDs, completely. Check framework specific settings to do this.

Congratulations,

you have now completed this module, XXE Injection!


SECURE
CODE
WARRIOR