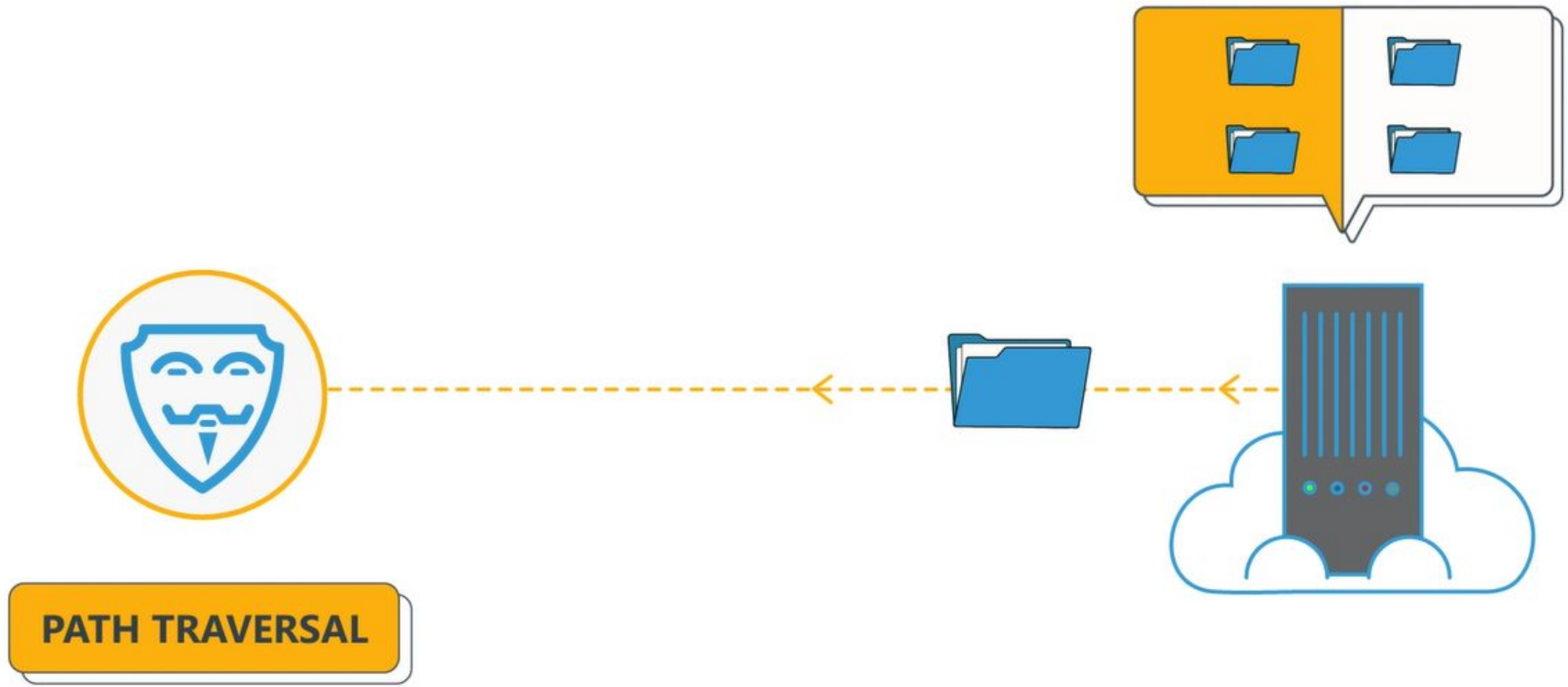
SECURE CODE WARRIOR®

PATH TRAVERSAL

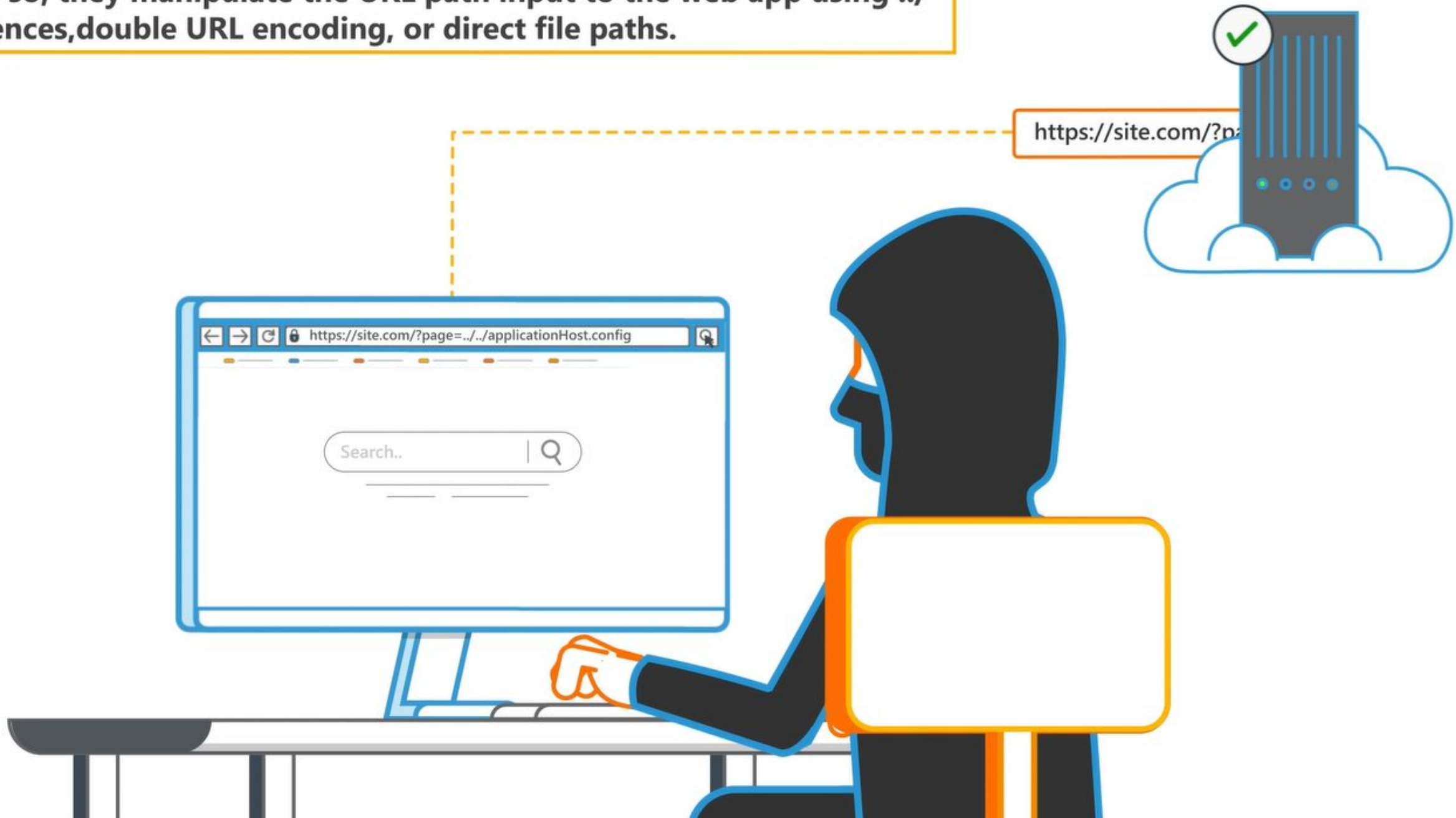
We'll go through

some causes and preventions of vulnerabilities in this category.

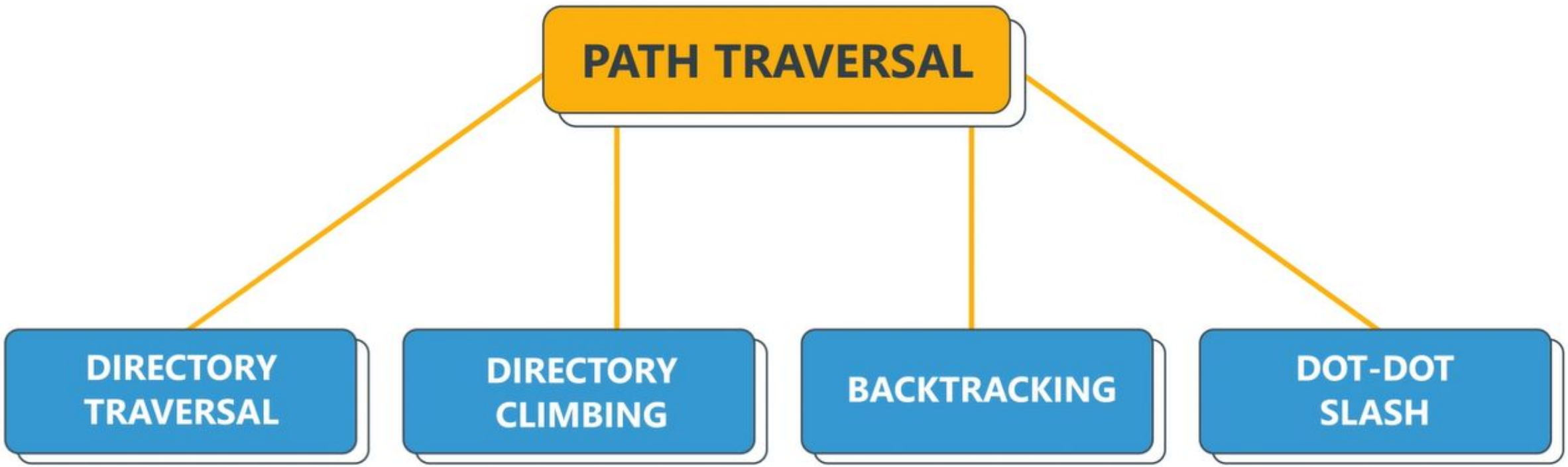
Attackers use path traversal to access files or directories stored in the file system of a web server or application, outside of the web root folder.



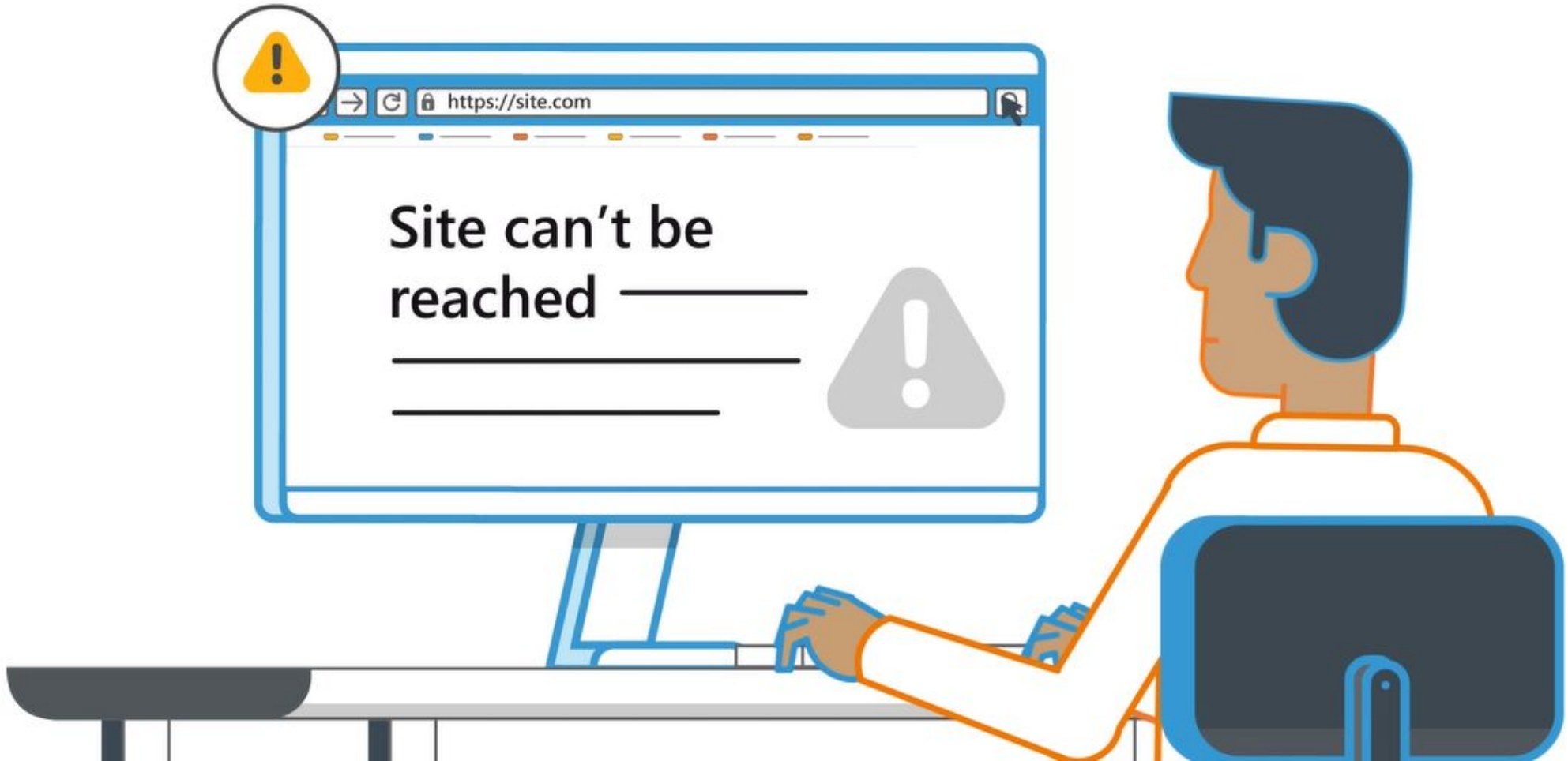
To do so, they manipulate the URL path input to the web app using ../ sequences,double URL encoding, or direct file paths.



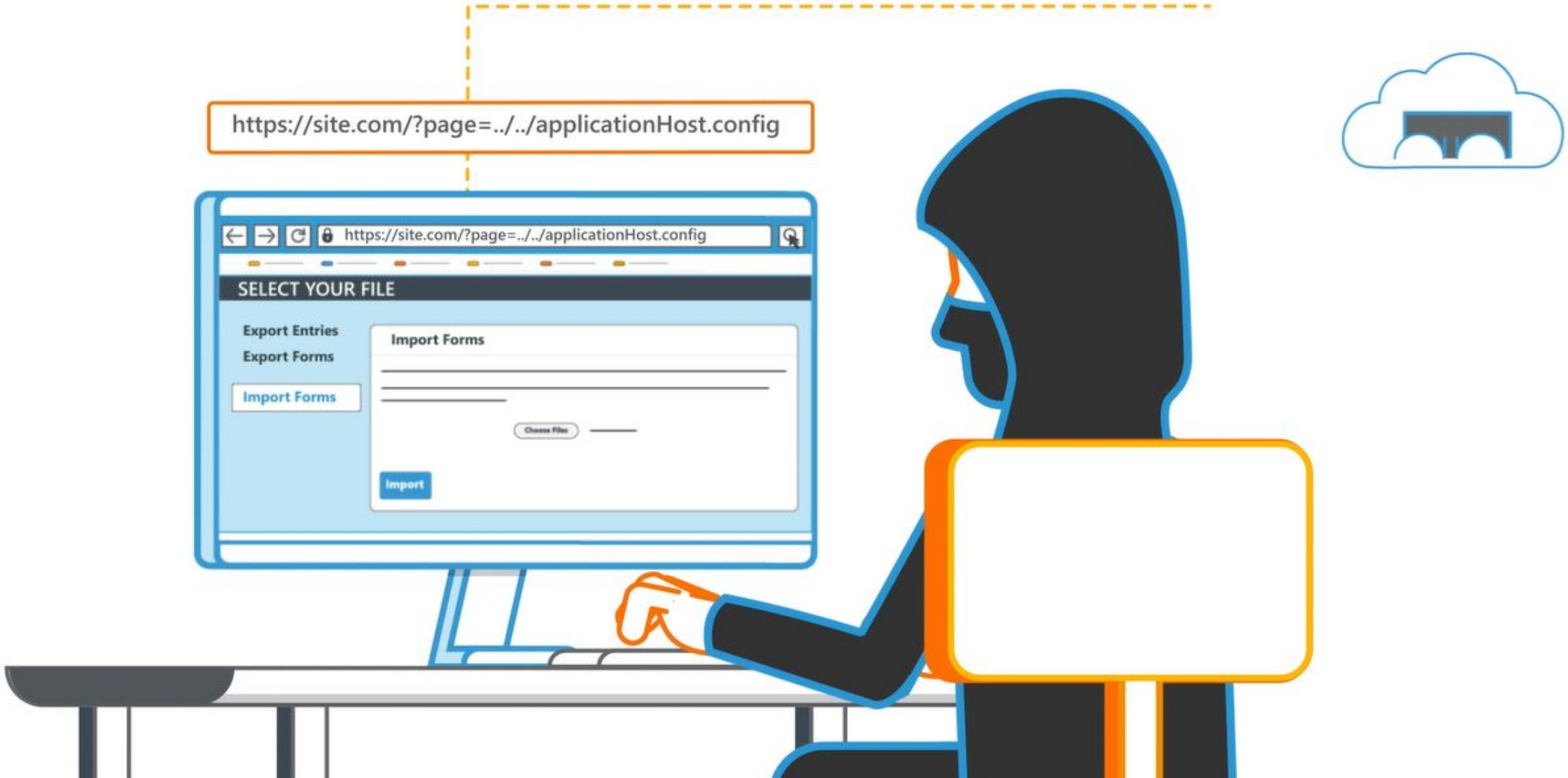
Path traversal is an attack with many names! It's also called directory traversal, directory climbing, backtracking, or, simply, dot-dot-slash.



Though most modern browsers have some level of built-in protection against basic path traversal attacks,



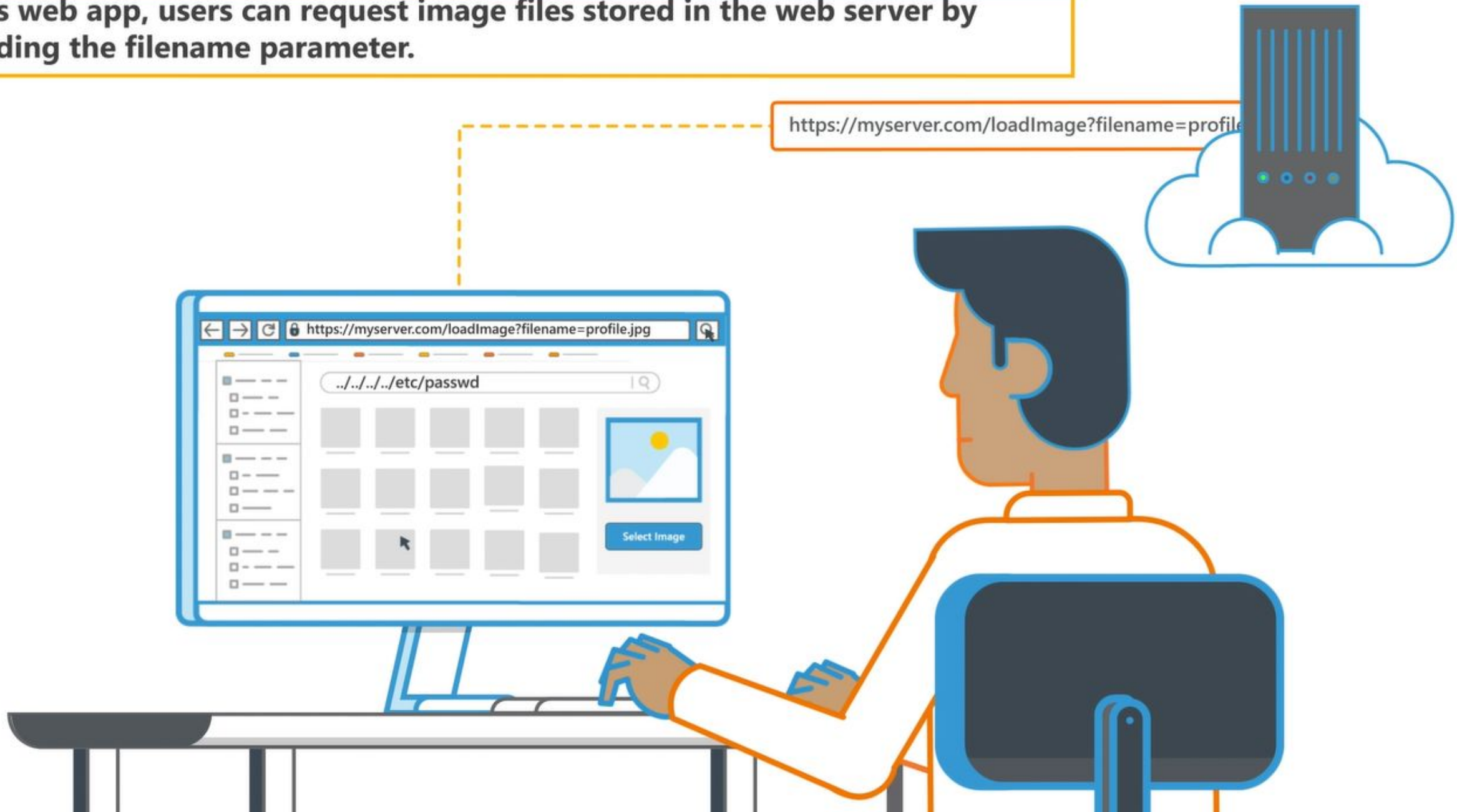
Attackers can still get by using variations of the dot-dot-slash technique.



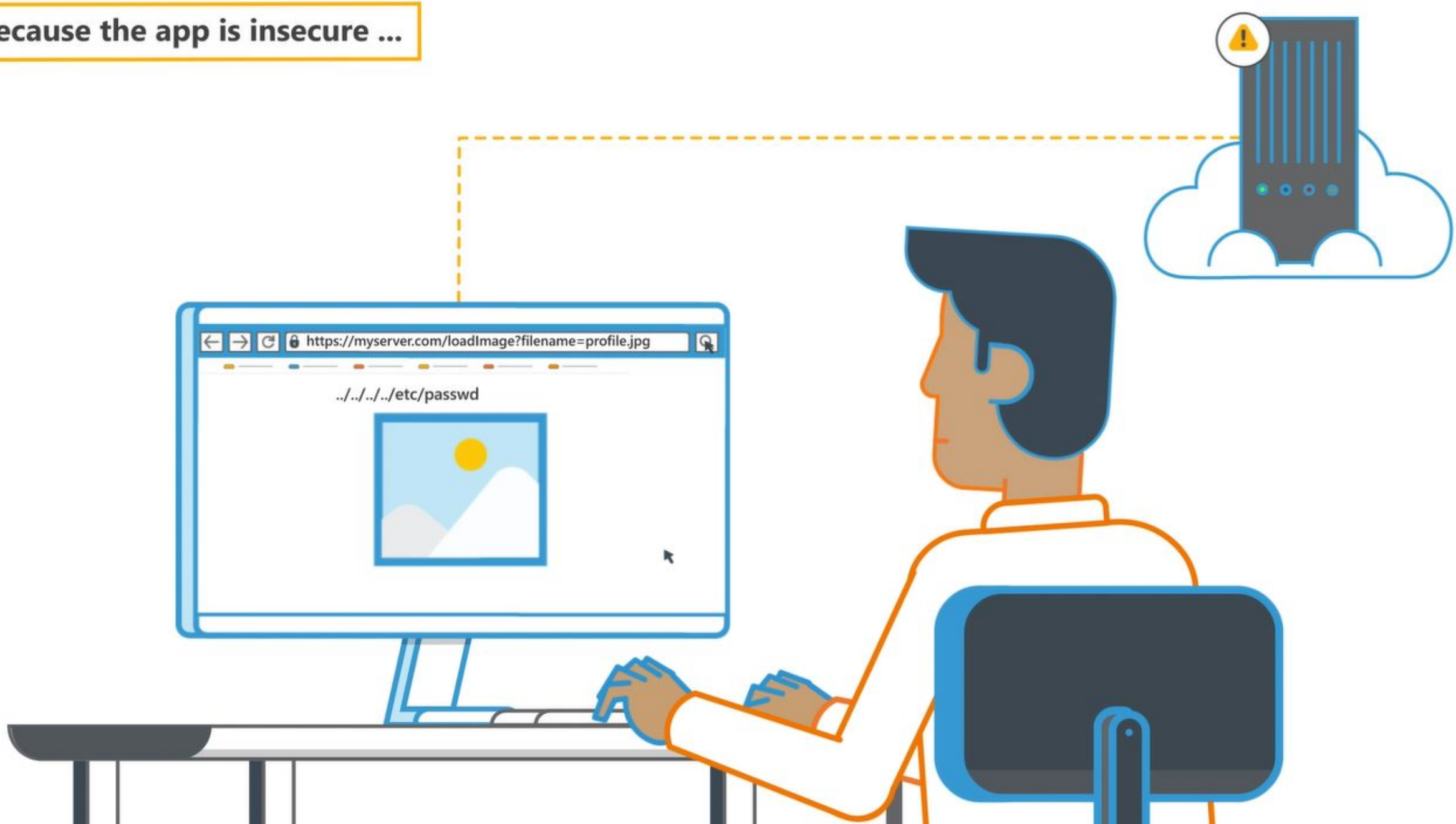
To understand

Path Traversal,
let's look at an example

In this web app, users can request image files stored in the web server by providing the filename parameter.

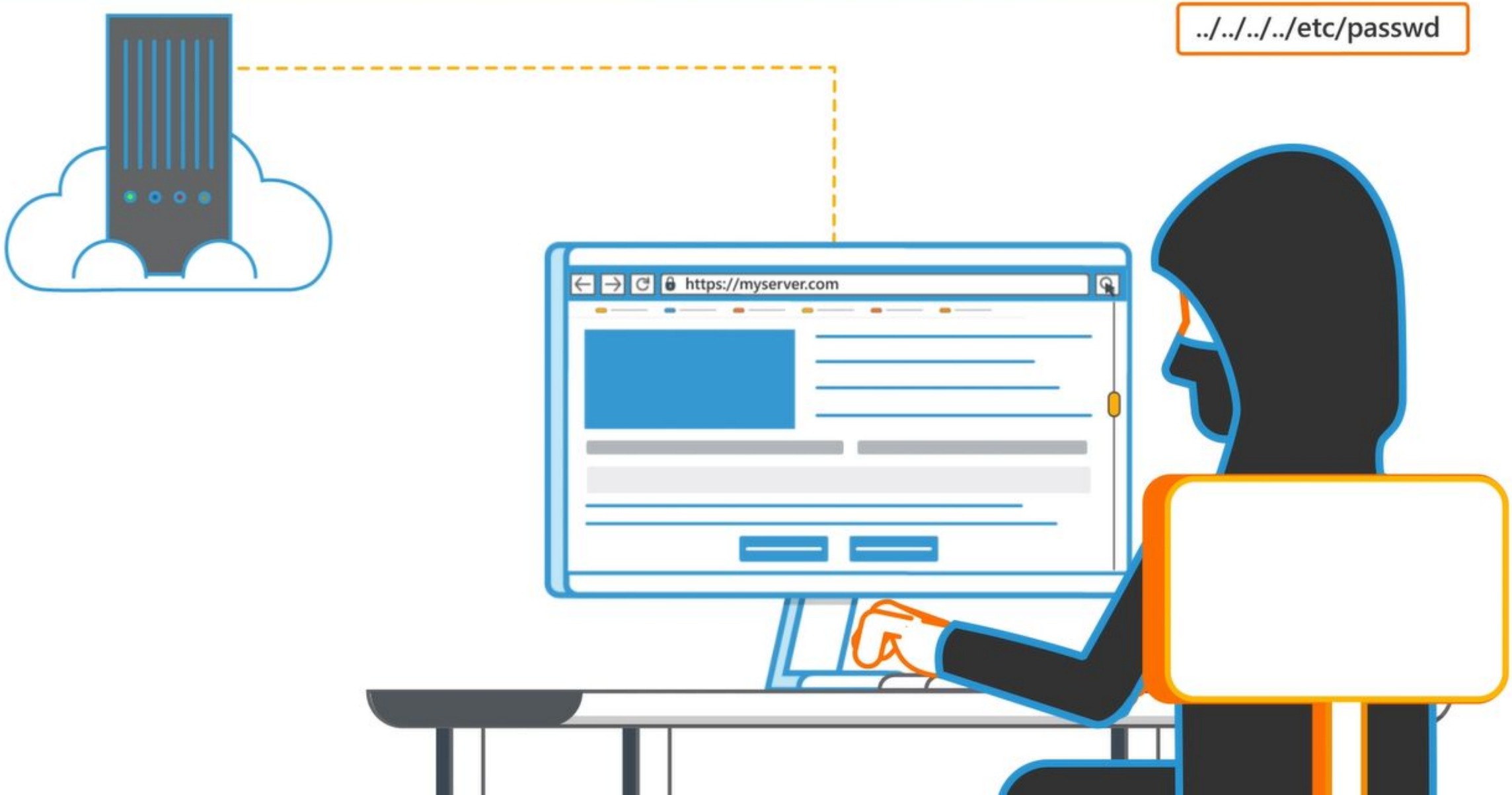


But because the app is insecure ...

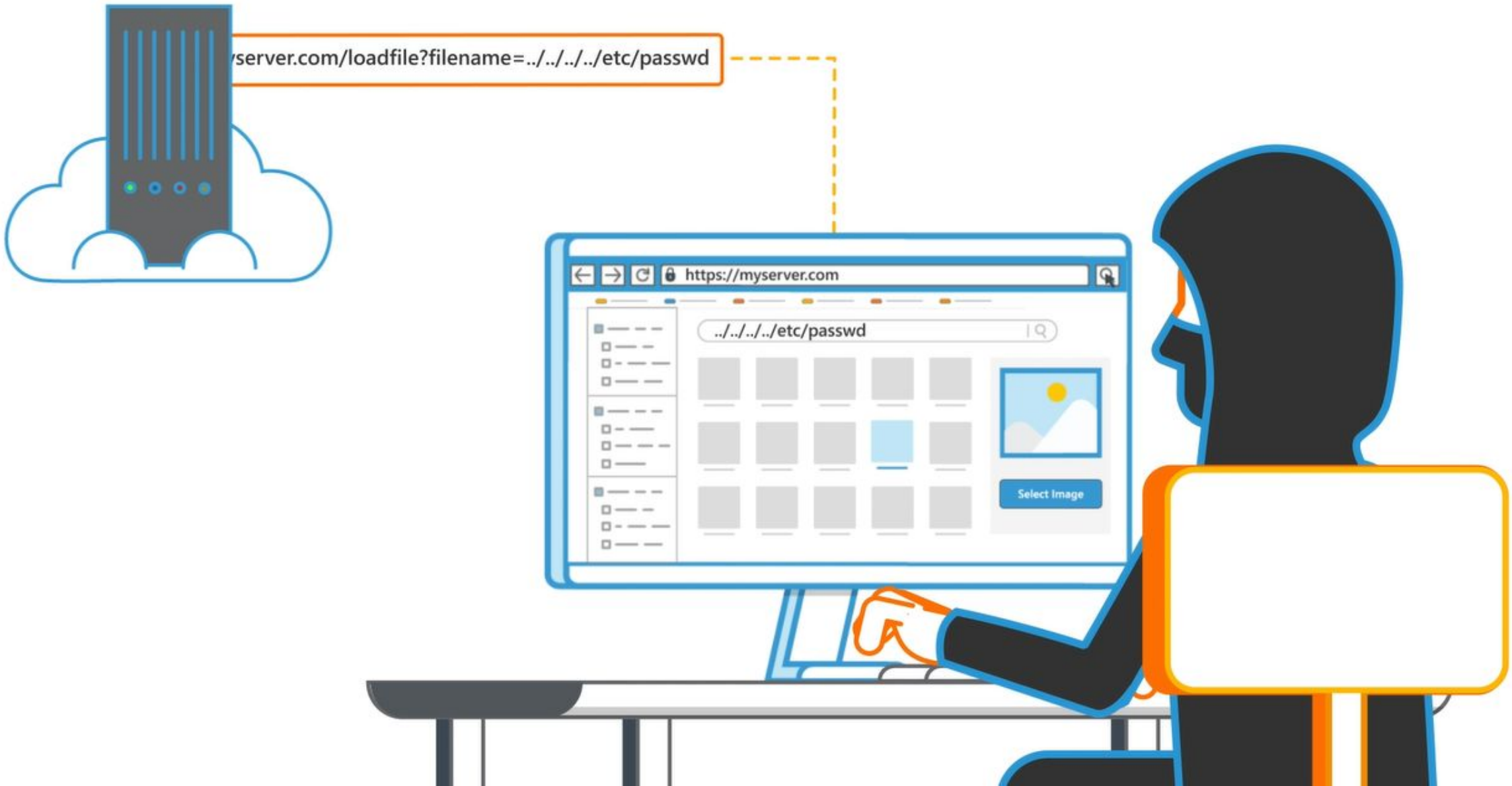


an attacker can enter an arbitrary file from the server's file system to access sensitive data.

../../../../etc/passwd

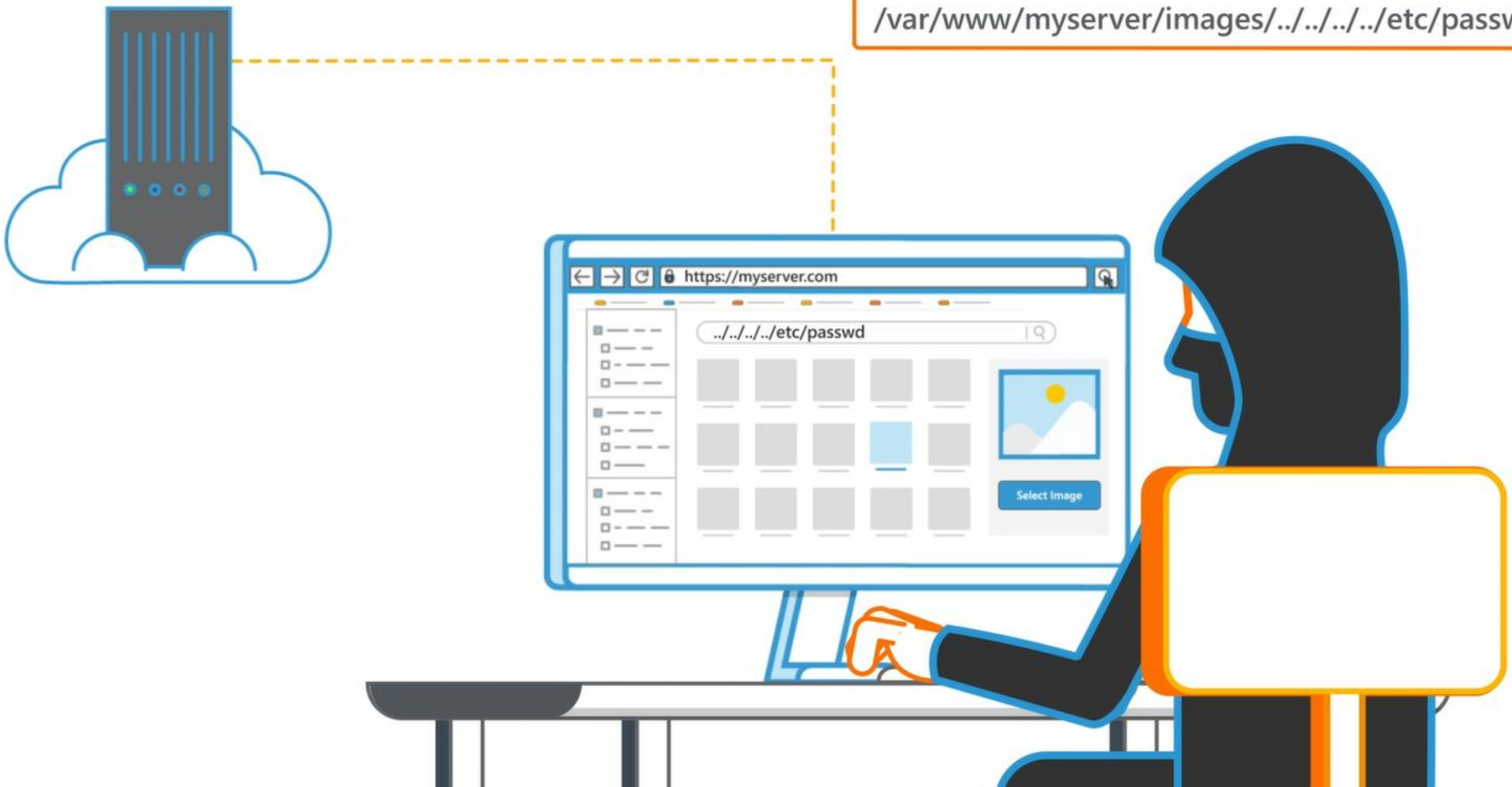


Here, the attacker uses a dot-dot-slash-etc-passwd file.



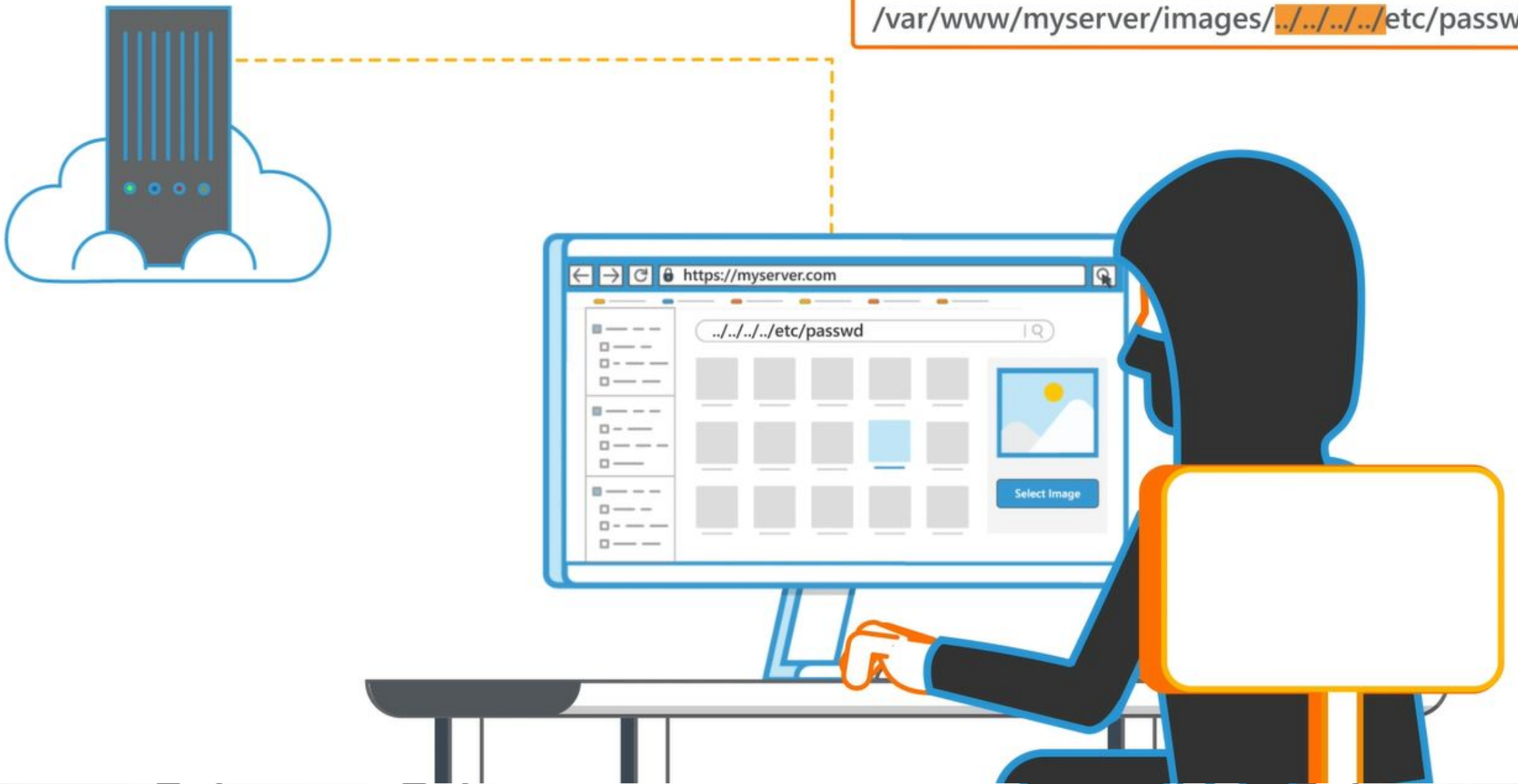
This causes the application to read from the following file path:

/var/www/myserver/images/../../../../etc/passwd

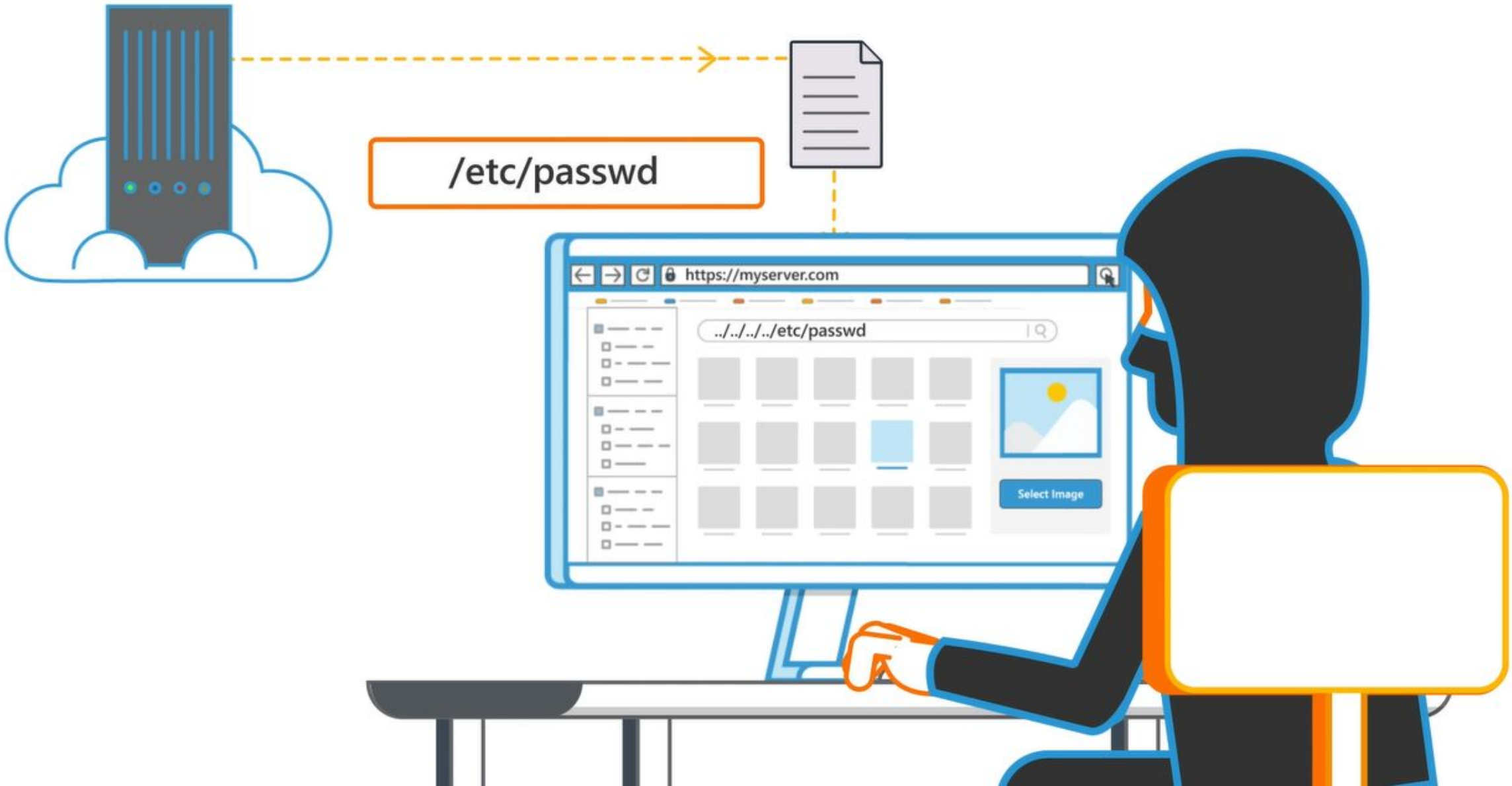


The four consecutive dot-dot-slash sequences moves the attacker up to the root filesystem.

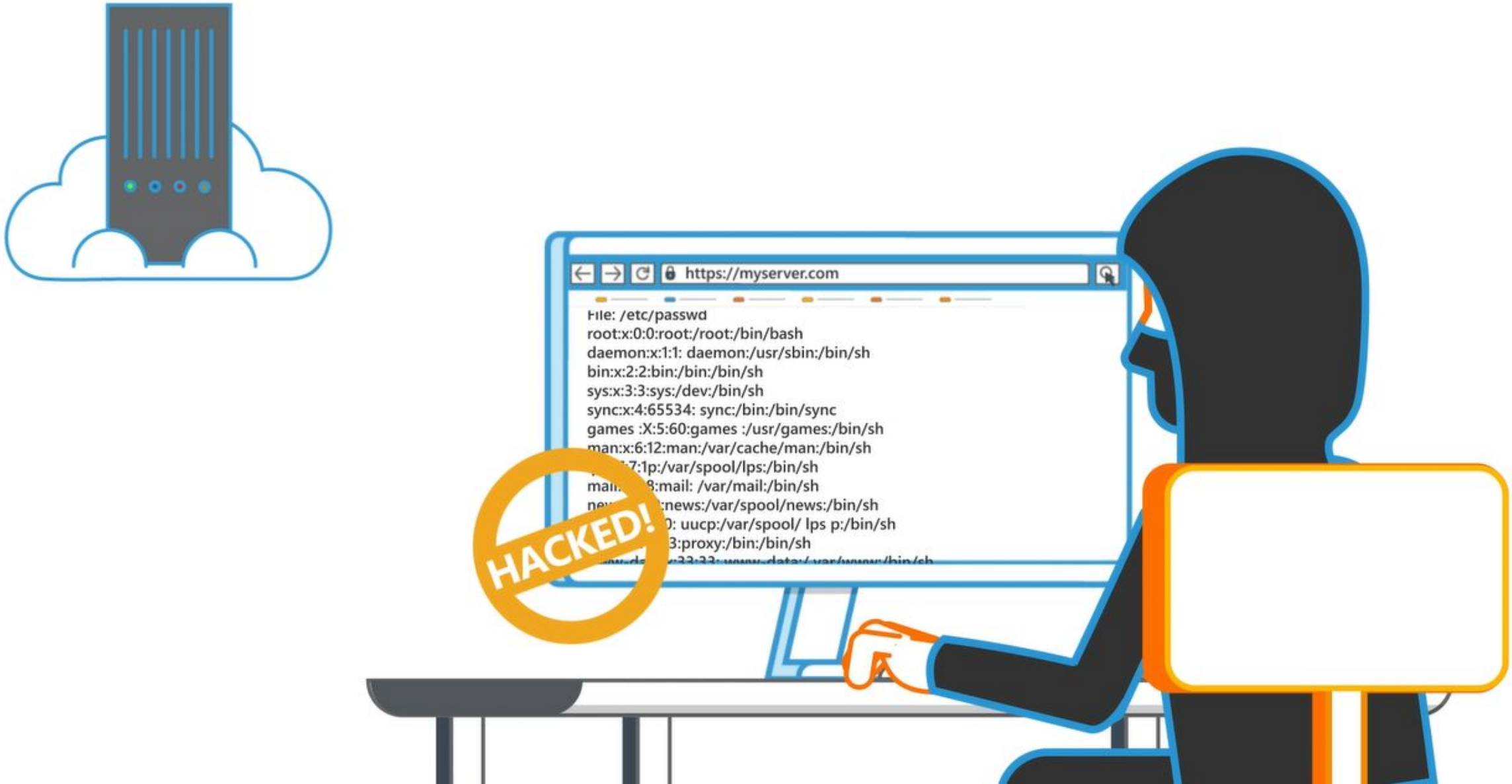
/var/www/myserver/images/../../../../etc/passwd



So the file actually being read is: ...



That's how the attacker is able to see sensitive information.



To prevent Path Traversal, developers should:

- Test for path traversal vulnerabilities
- Work without user input when using file system calls
- Use indexes instead of parts of file names when templating or using language files

To prevent Path Traversal, developers should:

- Be sure that users are unable to supply all parts of the path
- Use chrooted jails and code access policies to restrict unintended access and modification of files
- When user input must be submitted for file operations, normalize the input before using in-file IO APIs

Congratulations, you have now completed this module!

SECURE
CODE
WARRIOR