SECURE CODE WARRIOR

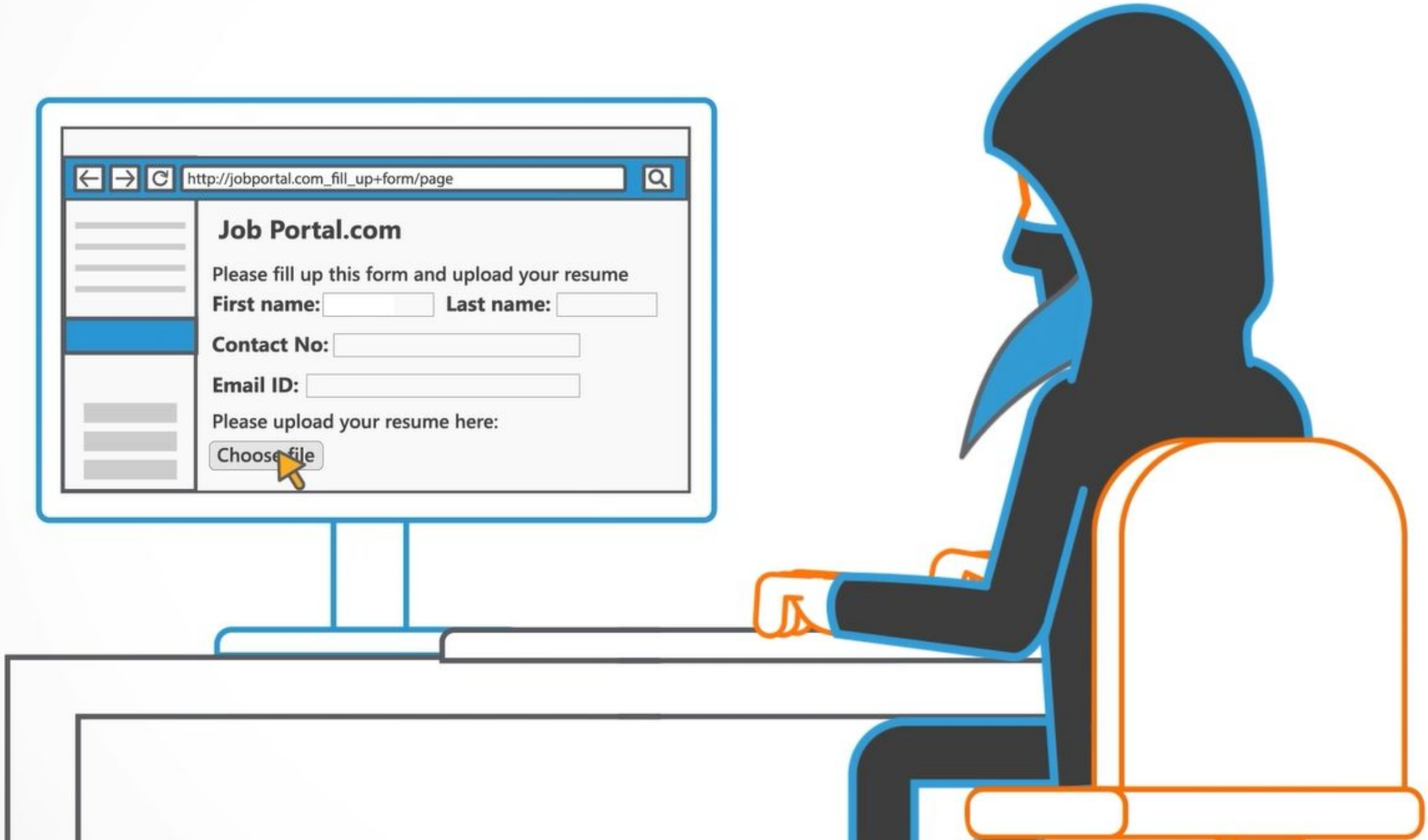
UNRESTRICTED FILE UPLOADS

We will explain

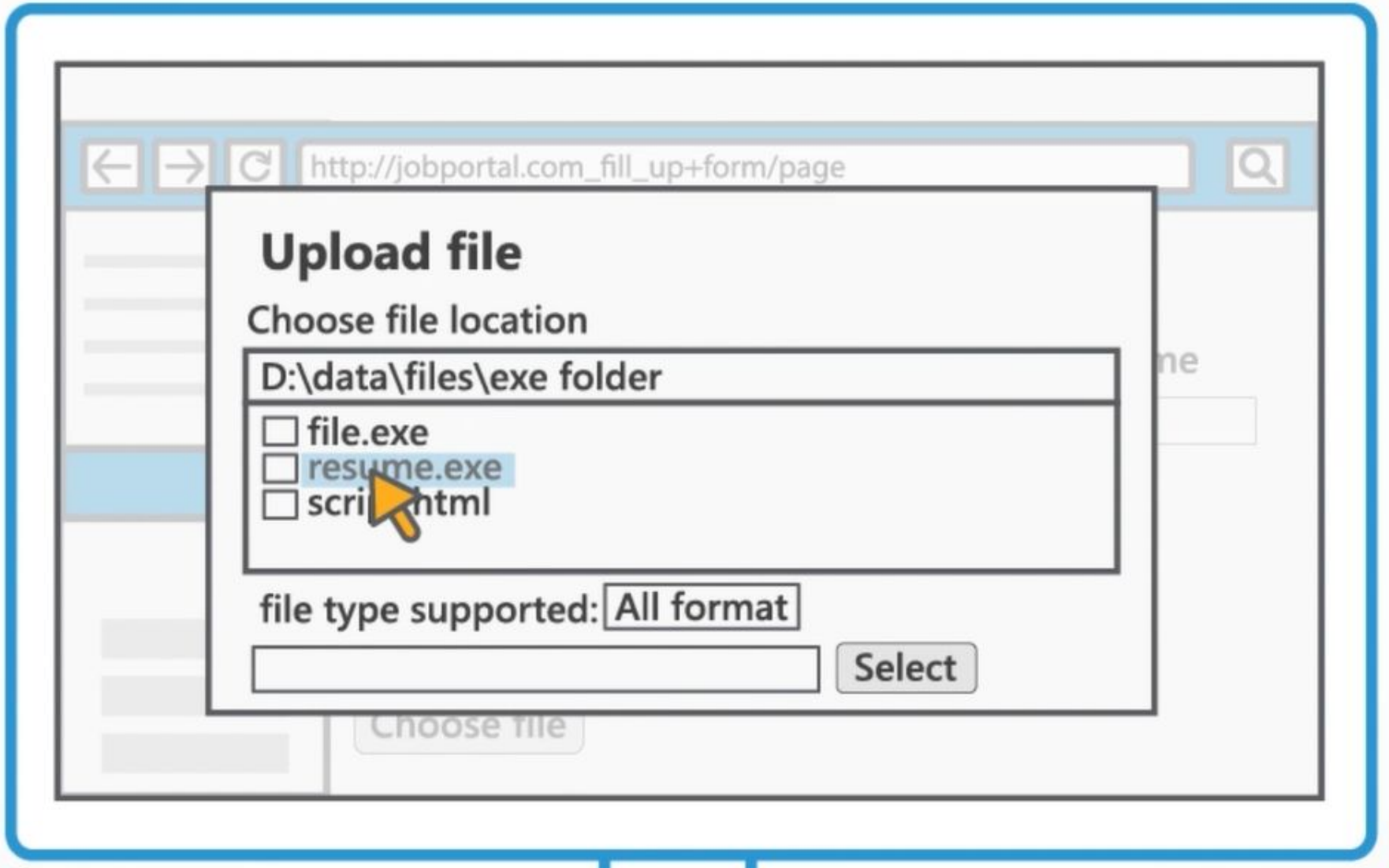
what Unrestricted File Uploads are, their causes and preventions and some potential hazards.

WHAT ARE UNRESTRICTED FILE UPLOADS?

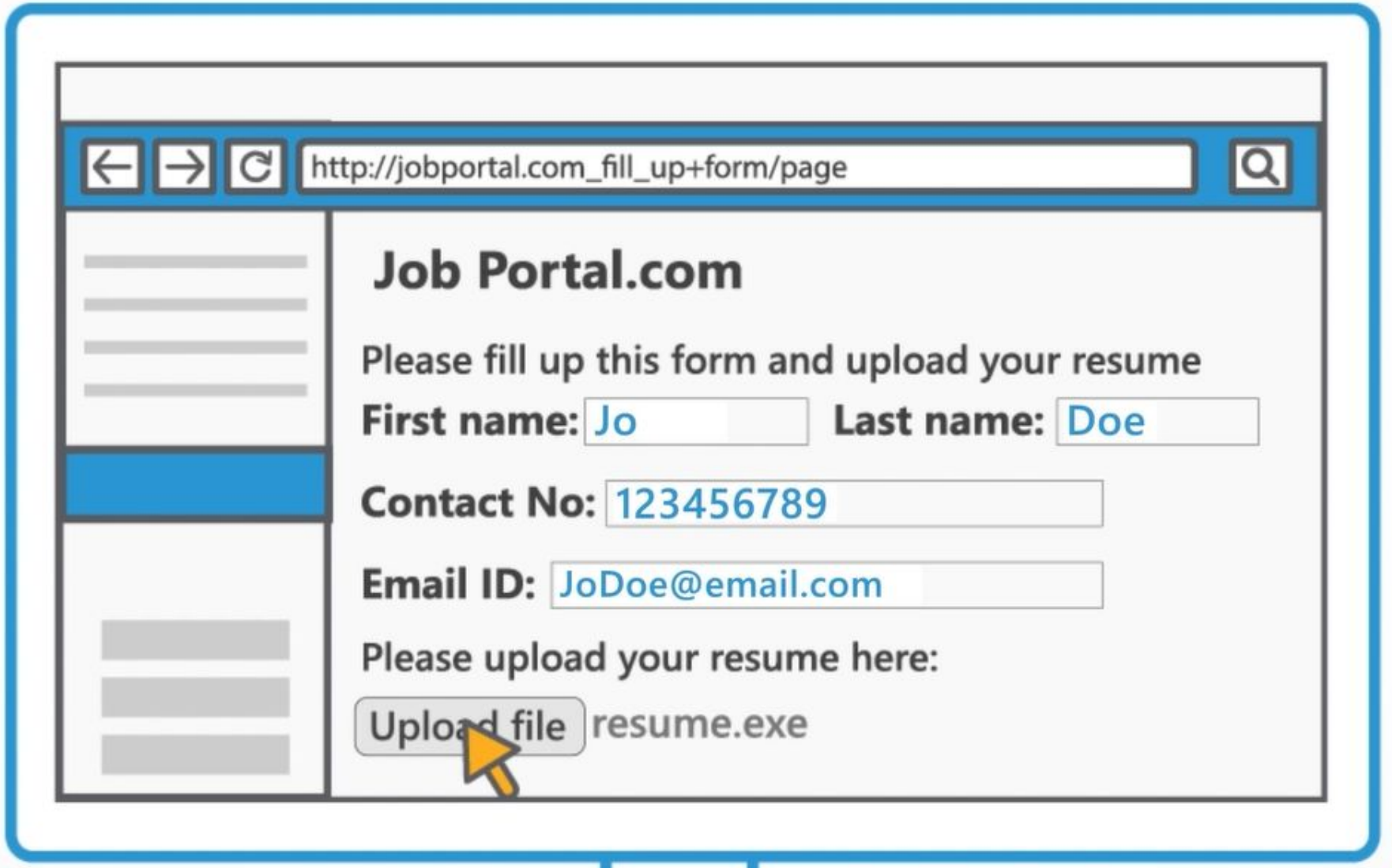
Some applications allow users to upload their own files.



If there are no limitations on the uploaded file name, type or size,

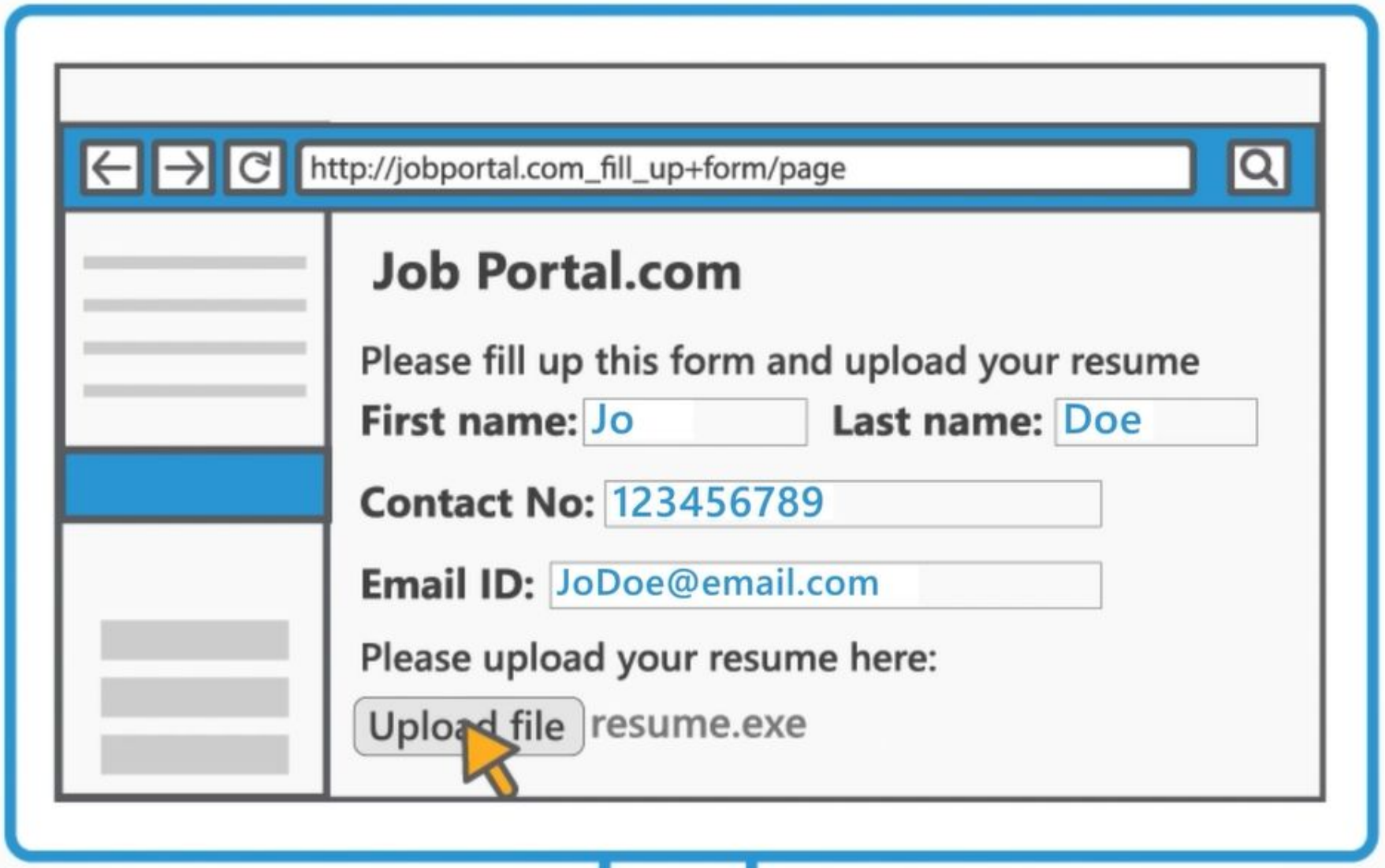


the unrestricted file upload could cause problems.



WHAT CAUSES UNRESTRICTED FILE UPLOADS?

These vulnerabilities can occur when files uploaded to the application are not verified,



or if there are no checks to limit the file size and possibly dangerous extensions or content.

Upload rate limitation could be missing as well.

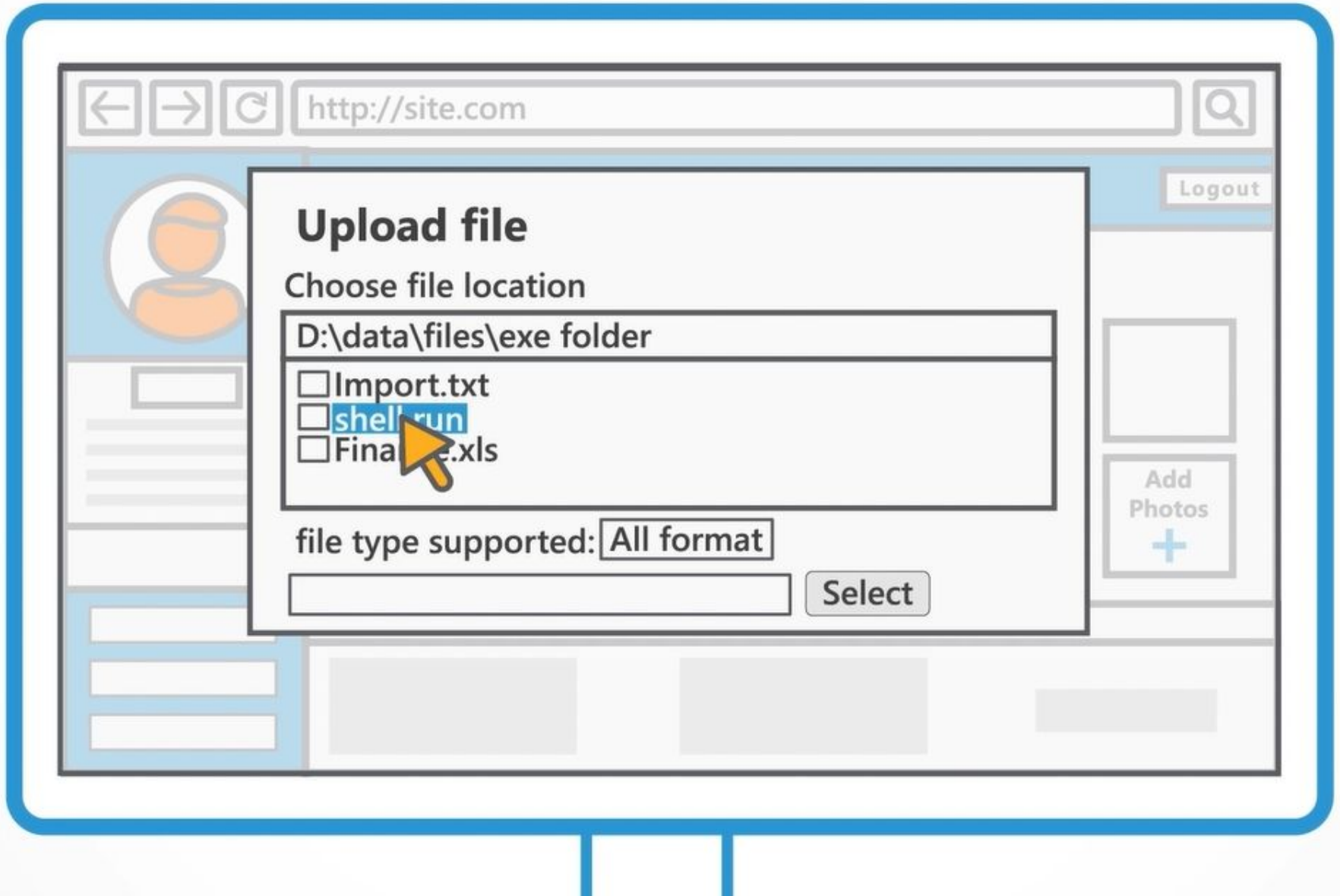
To understand

Unrestricted File Upload vulnerabilities,
let's look at some examples.

Let's say a web application allows users to upload files and view them afterwards in a directory.

A malicious user notices this functionality and decides to upload a web shell.

The attacker can browse to the file through the “Uploads" directory



and can pass commands to their web shell. All commands passed to the shell get executed on the server

04

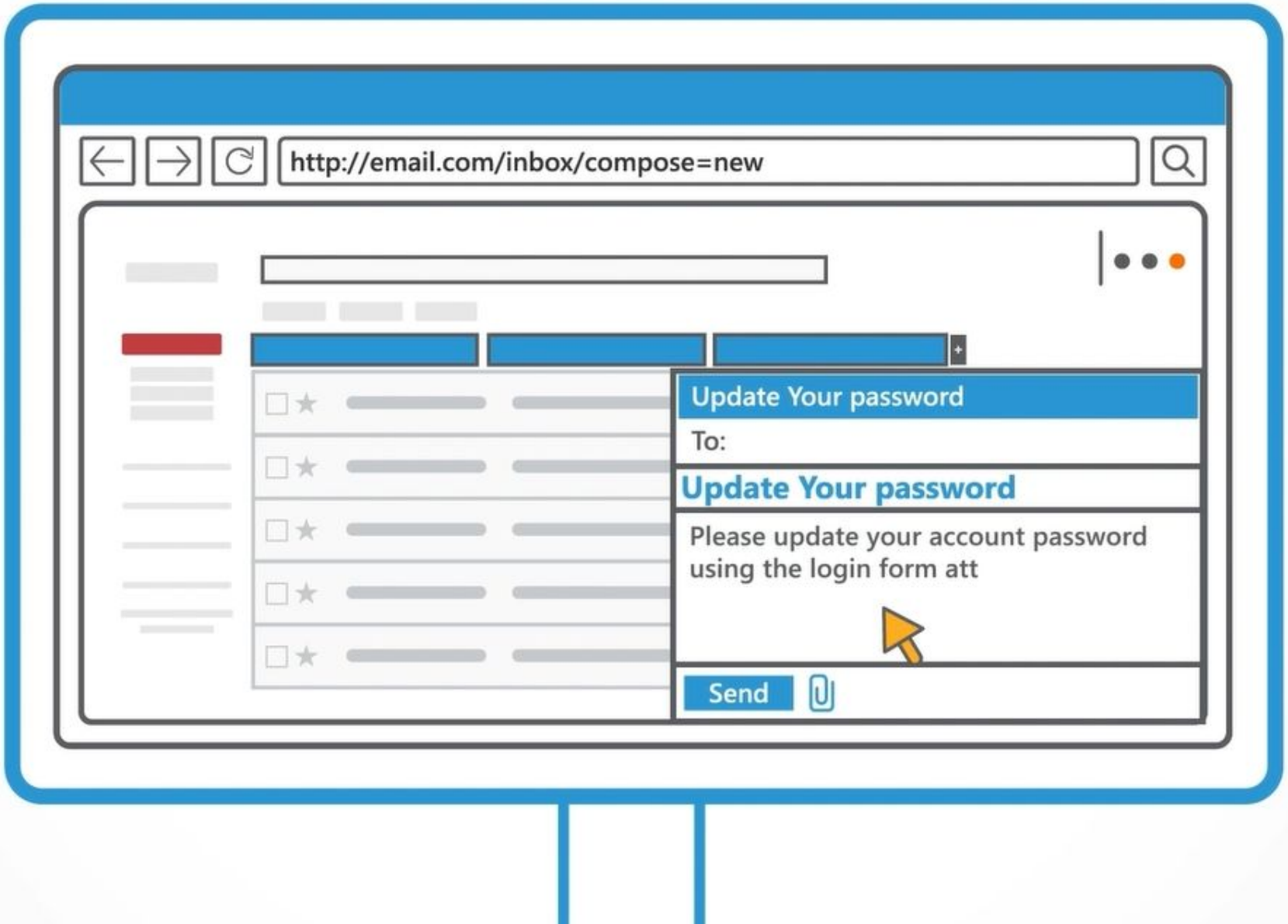
shell.run

```
/uploads/shell.run?cmd=whoami
webuser
```

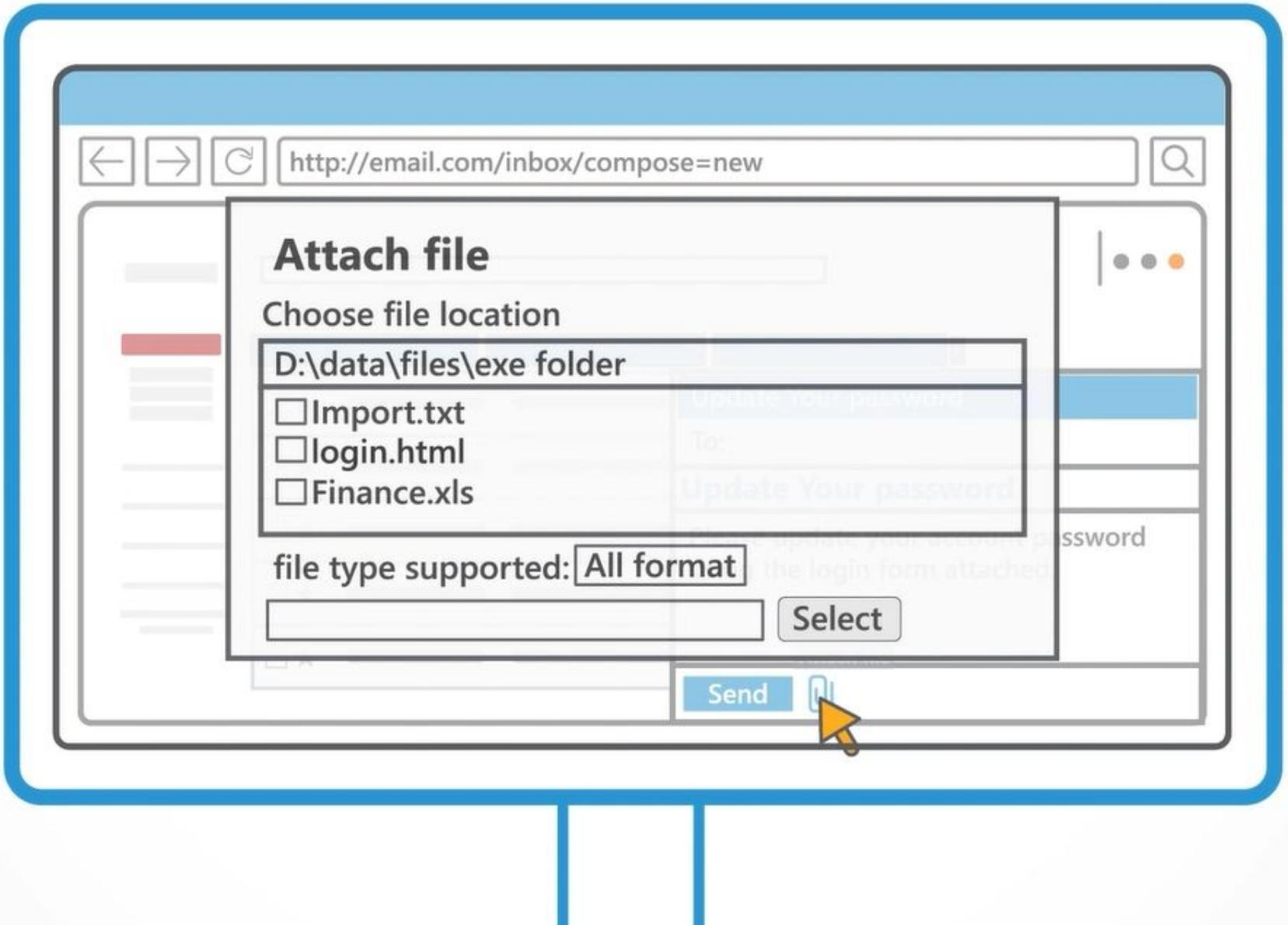
and are run with the same privileges as the web application.

A web application allows users to upload files and send them as an attachment to other users

A malicious user notices this functionality and decides

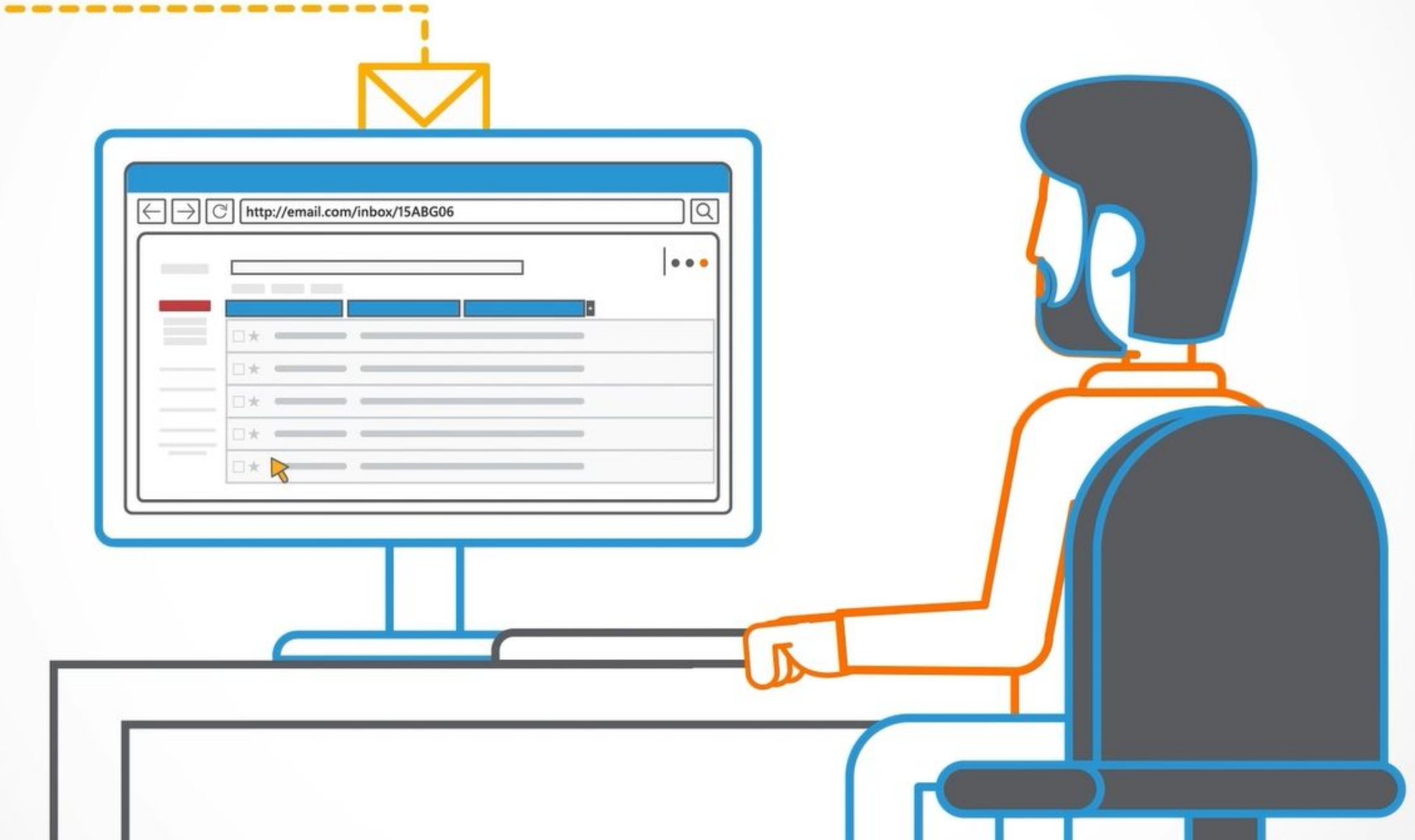


to upload an HTML page containing a fake login.

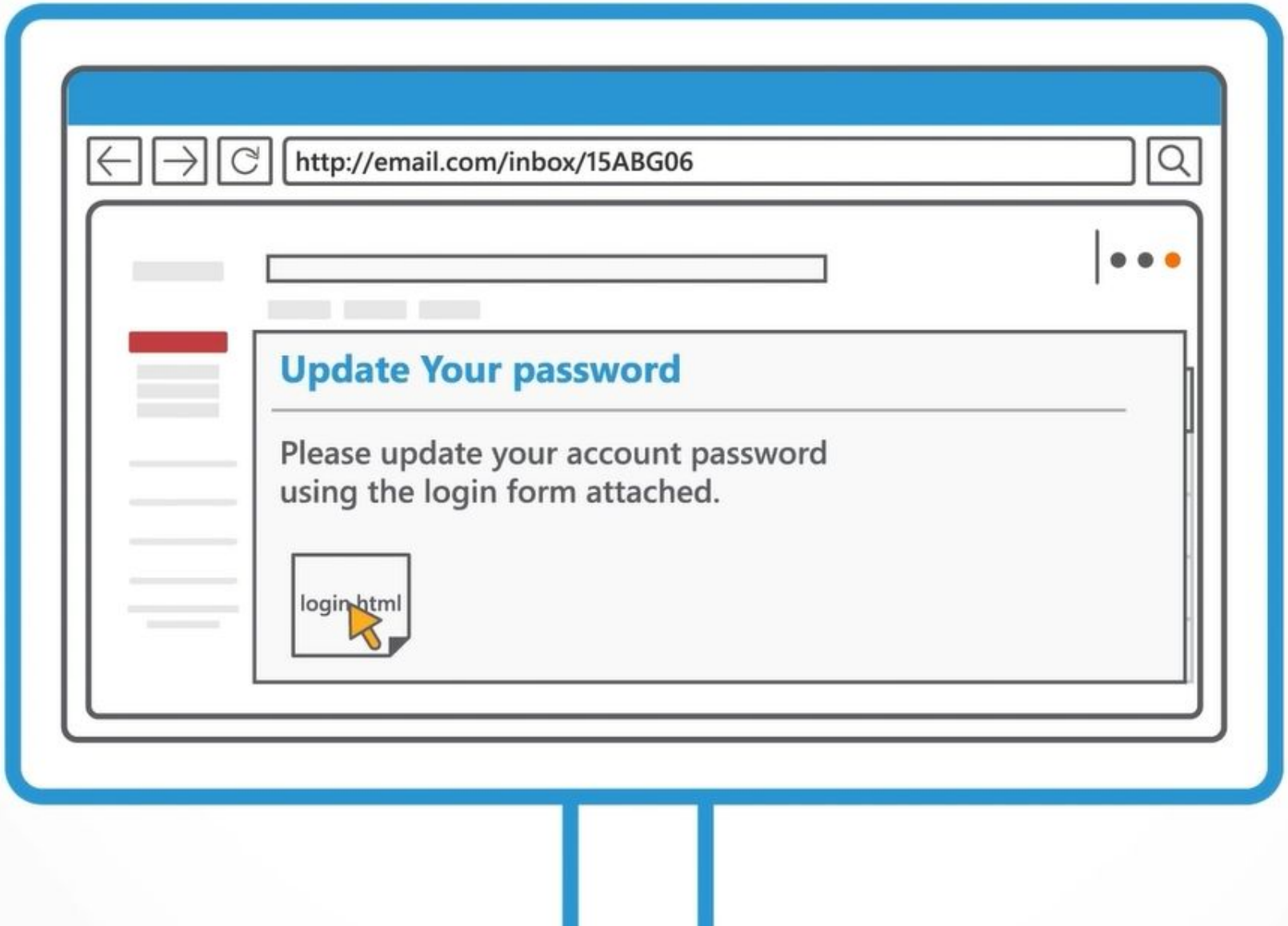


The attacker then sends the fake login page to a number of people,

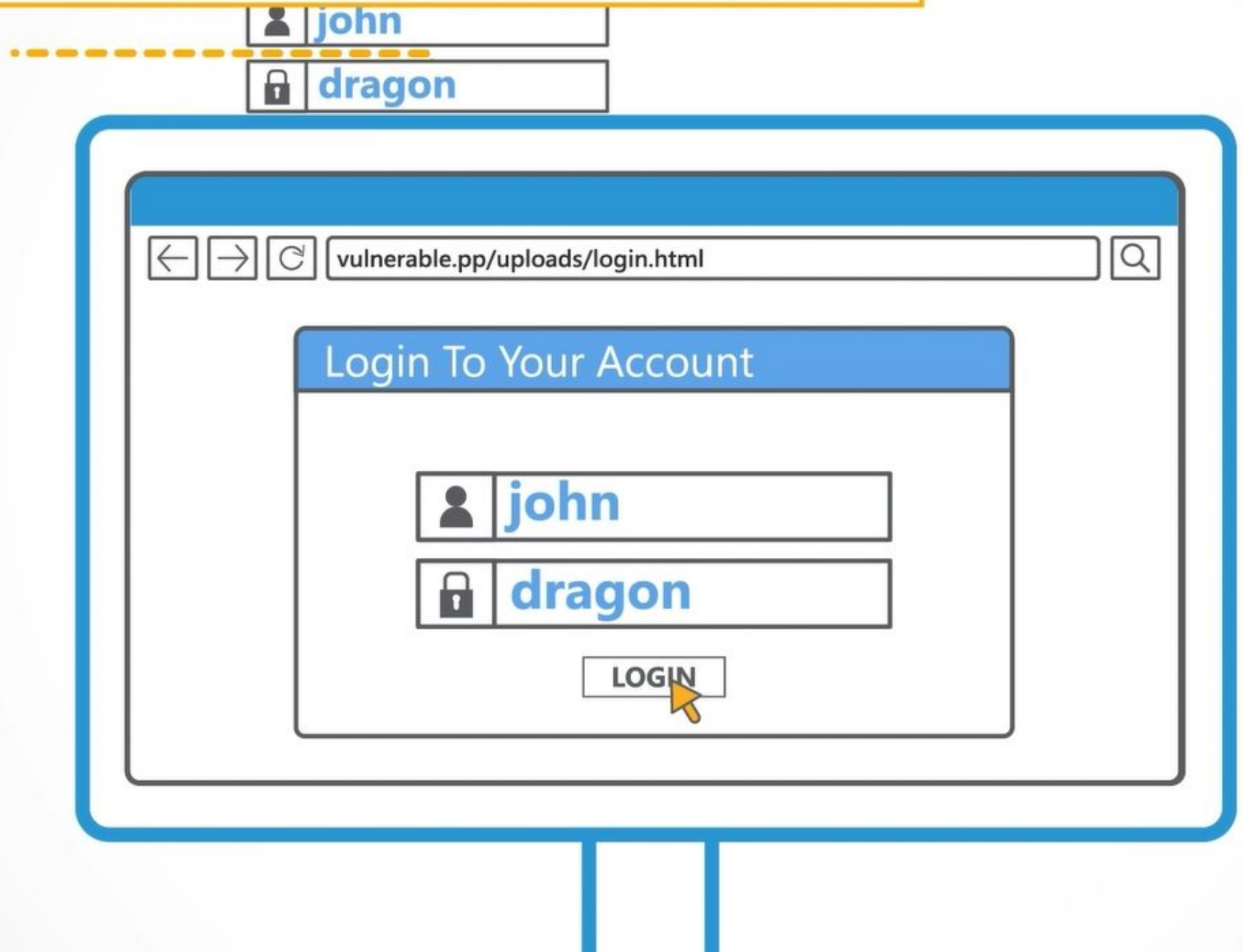
in hopes of someone entering their login credentials.



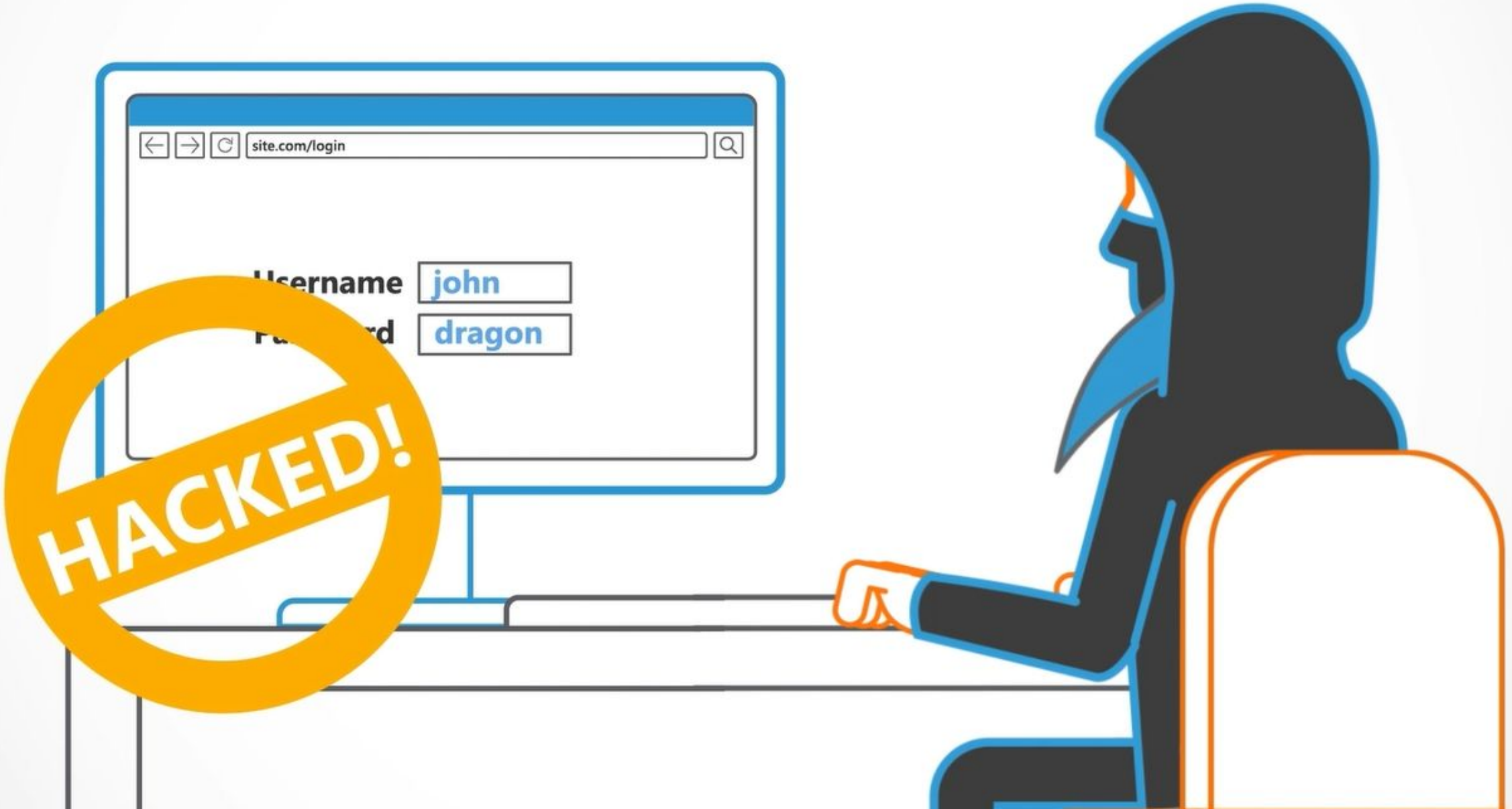
This is commonly called a phishing attack.



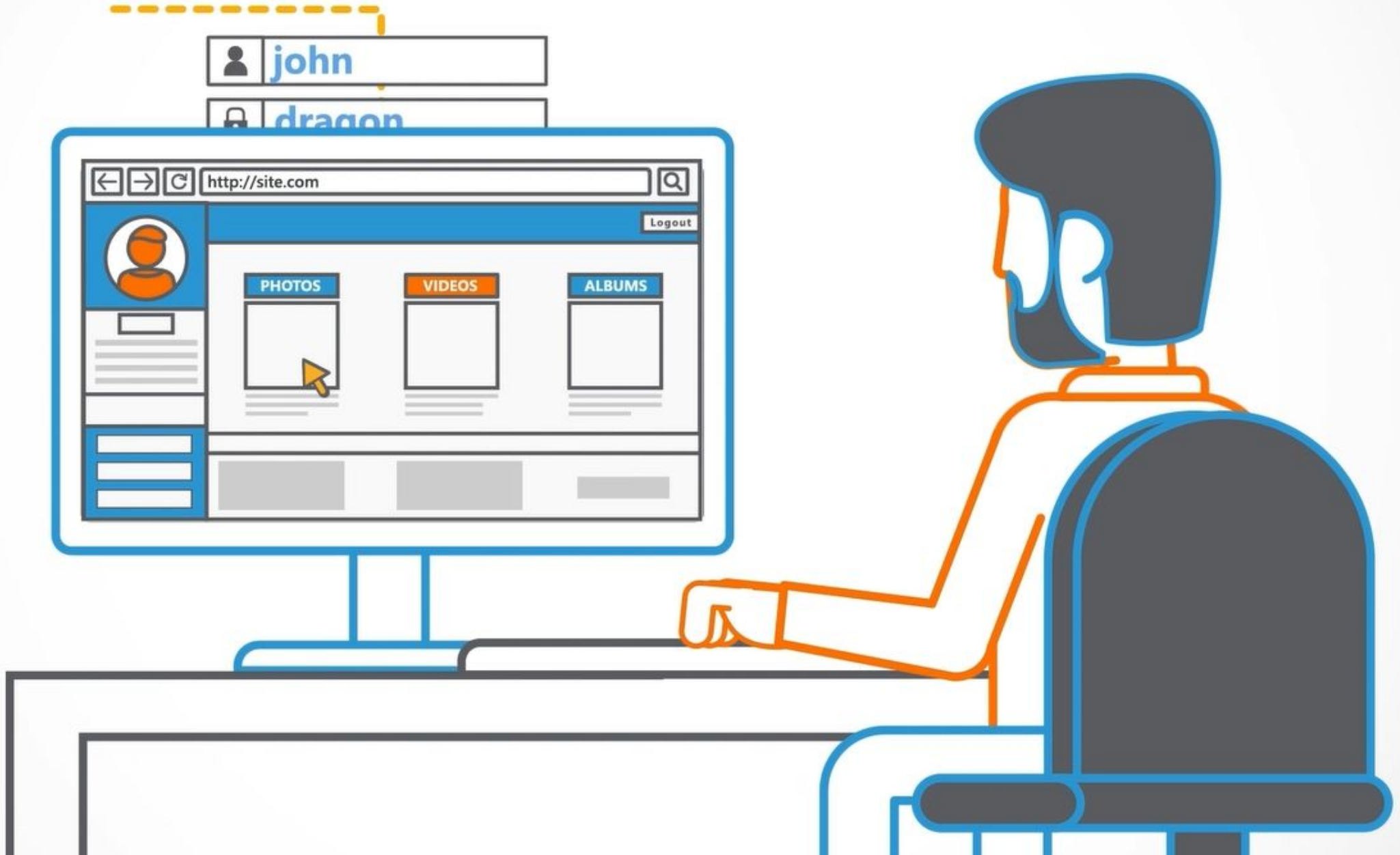
When a user submits credentials through the fake login page,



the information is sent to the attacker. And the user is also logged in normally,

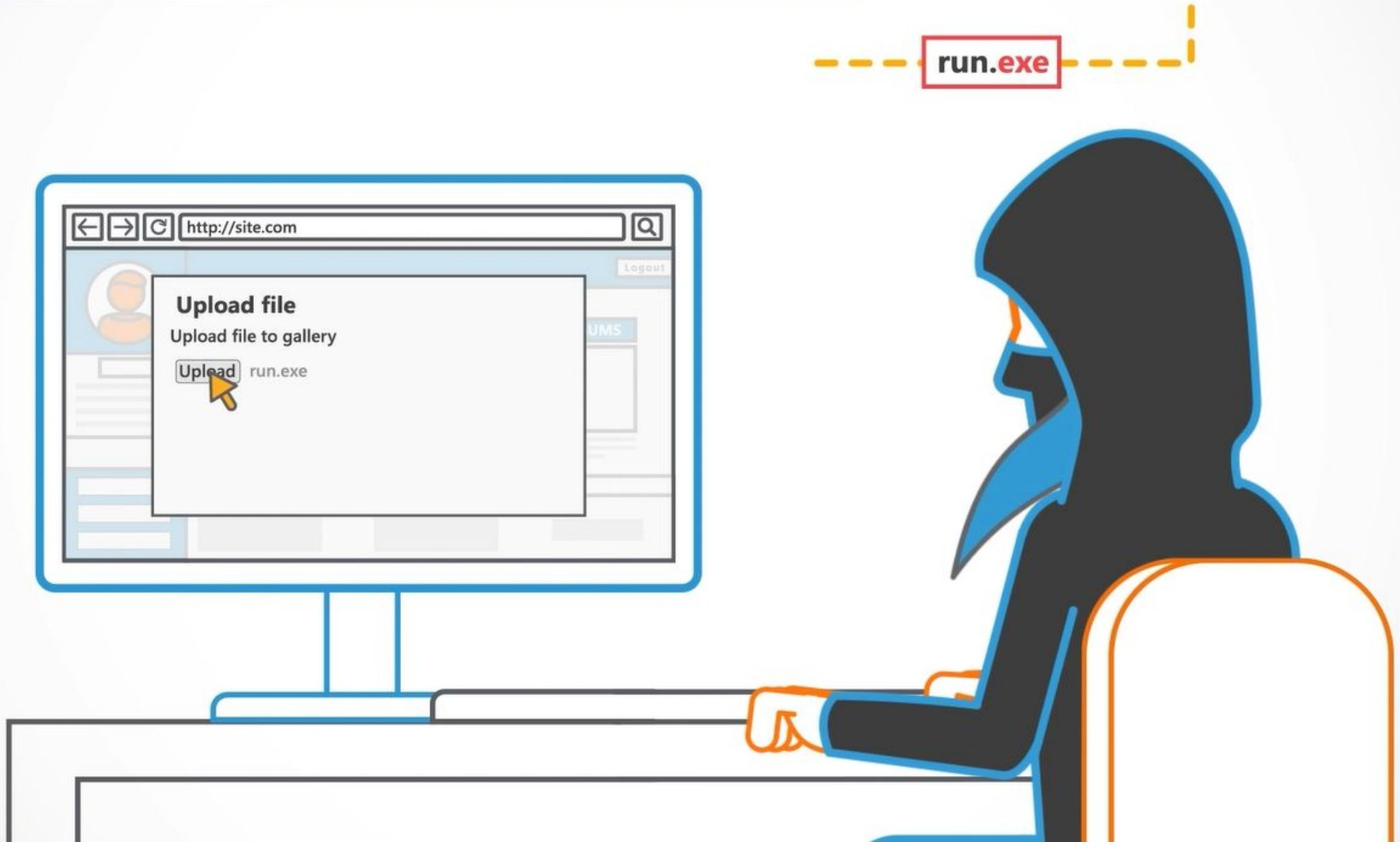


so they aren't even aware, they've just given away their credentials to an attacker.

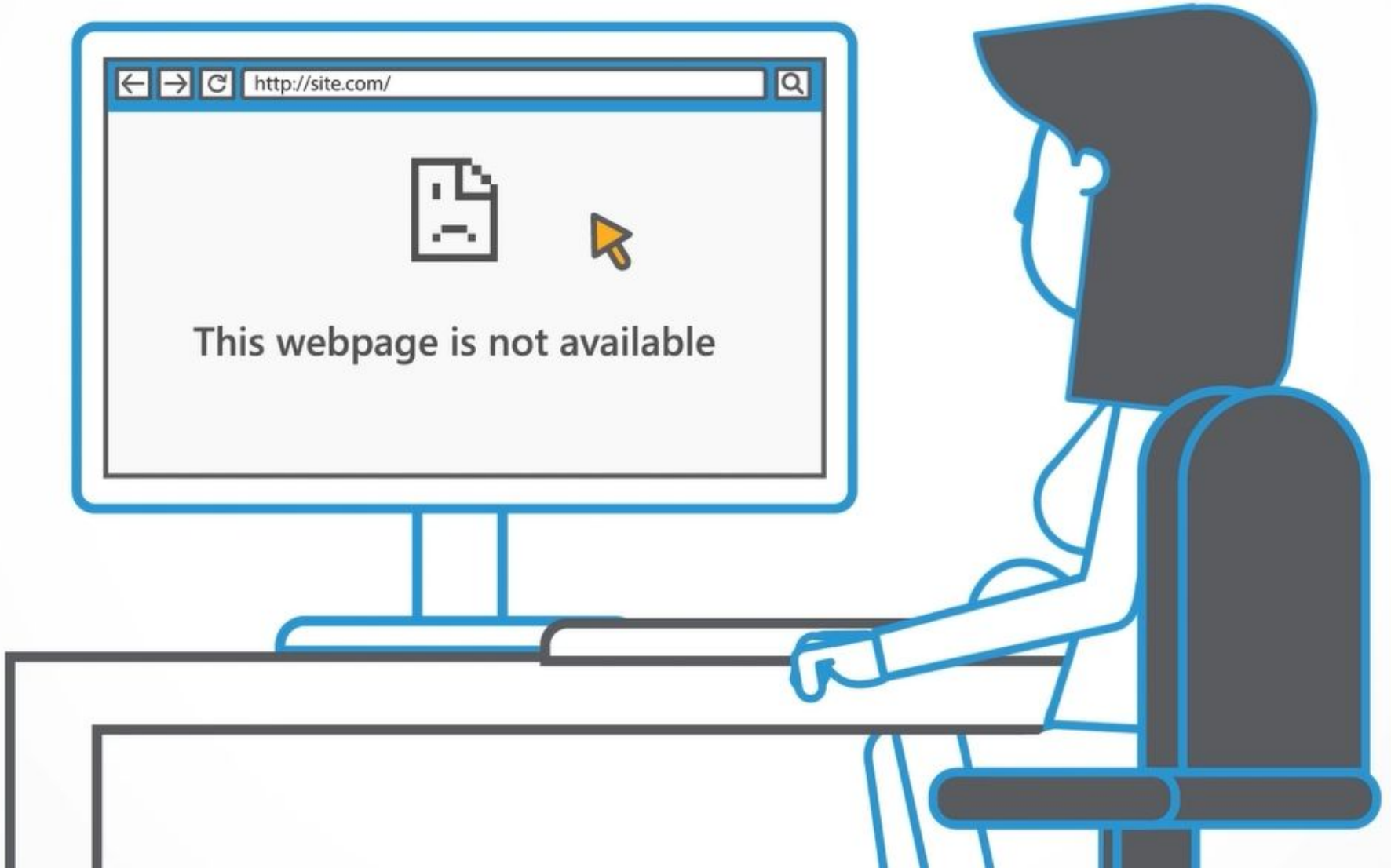


Unrestricted File Uploads can have significant impacts.

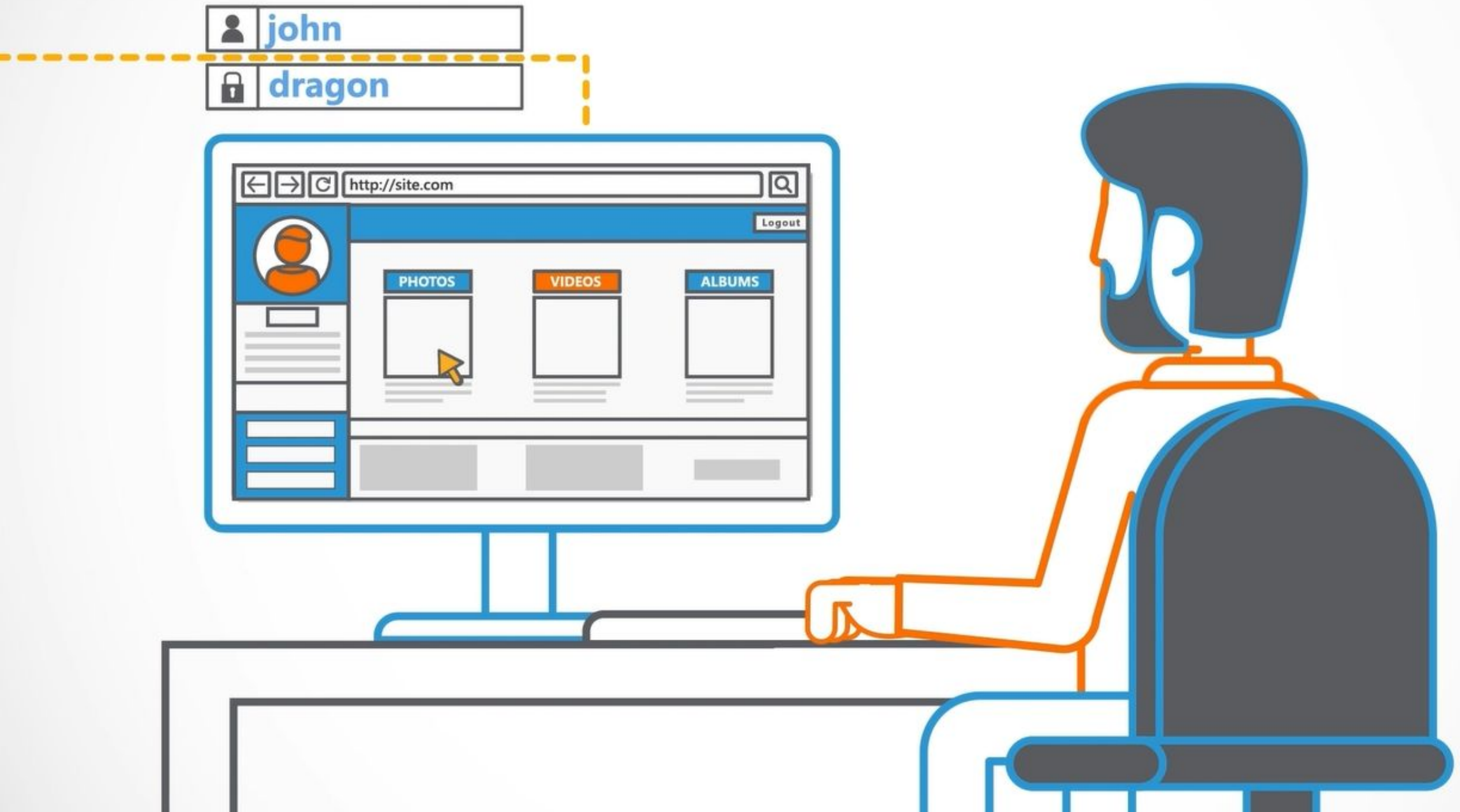
By uploading a large amount of malicious files, or even a zip bomb,



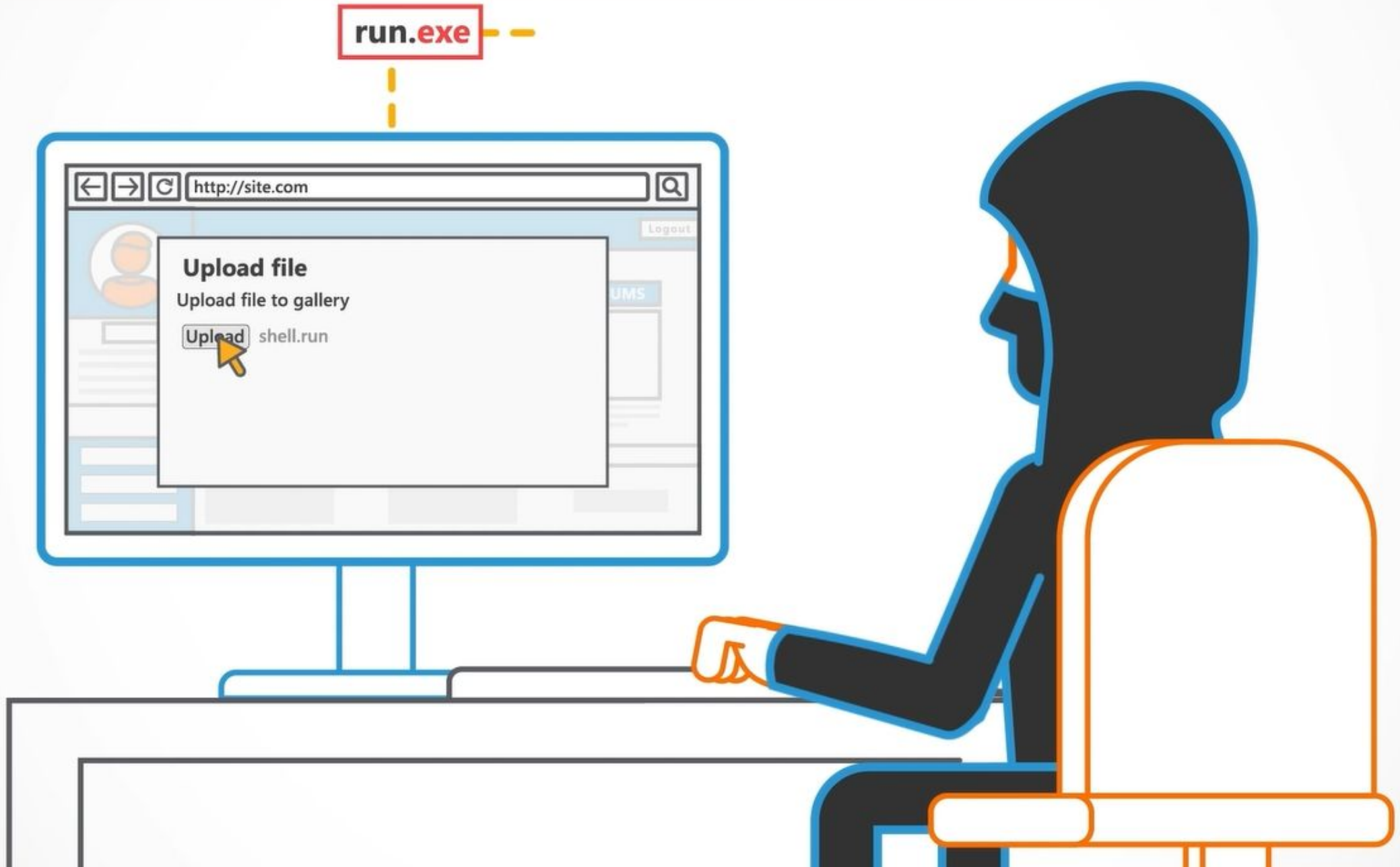
an attacker could launch a Denial-Of-Service attack.



Users could be tricked into providing sensitive information to a phishing page uploaded by an attacker.



Or, an attacker could upload a web shell and gain remote access to the web server.



To prevent Unrestricted File Uploads, developers should:

- Apply allowlist validation on file names and extensions.
- Remove special characters from filenames and limit the length to a fixed amount of characters.
- Only allow particular extensions. Also, save with the detected file extension, not with the original extension.
- Scan uploaded files for malicious content.
- Use Mime type detection.

To prevent Unrestricted File Uploads, developers should:

- Store uploaded files in a private directory without execution privileges.
- Enforce minimum and maximum file size limits.
- Remove EXIF data.
- And finally, store files with randomly generated file names, keeping a table mapping to the original names if necessary.

Congratulations,

you have now completed this module, Unrestricted FIle Uploads!


SECURE
CODE
WARRIOR