SECURE CODE WARRIOR®

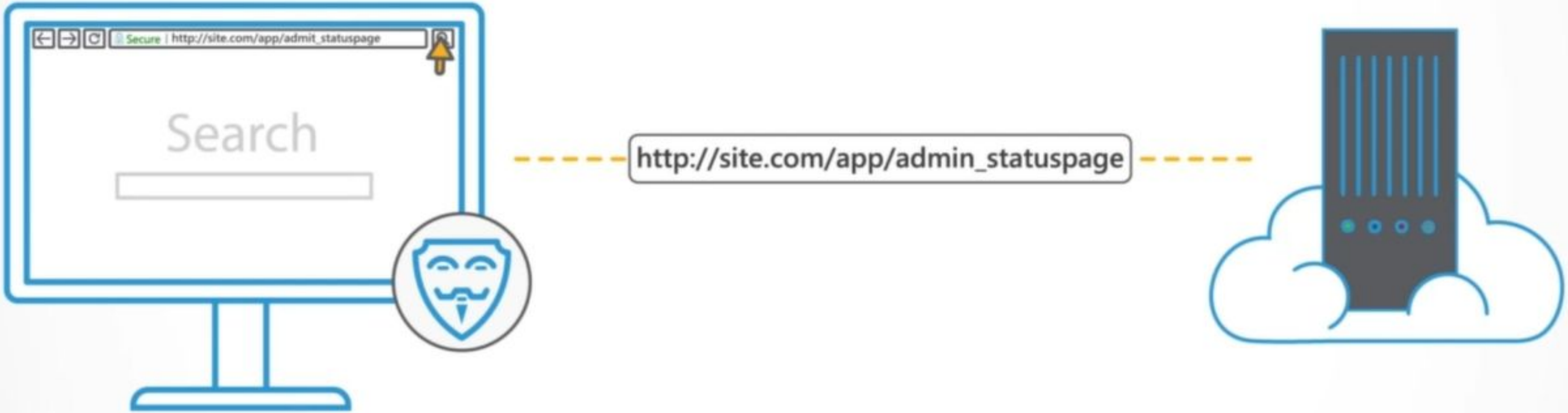
MISSING FUNCTION LEVEL ACCESS CONTROL

We will explain

what Missing Function Level Access Control,
its causes and preventions and some potensial hazards

SO, WHAT IS THE “MISSING FUNCTION LEVEL ACCESS CONTROL” VULNERABILITY?

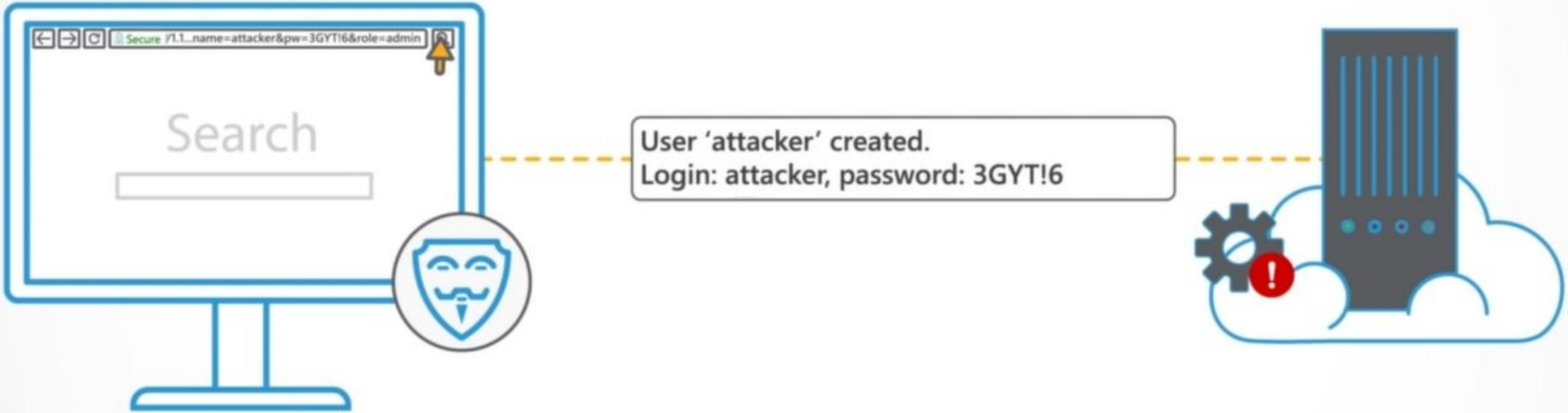
This occurs when users can perform functions that they are not authorized for,


Secure | http://site.com/app/admit_statuspage
Search
http://site.com/app/admin_statuspage

or when resources can be accessed by unauthorized users.

Secure | http://site.com/app/admit_statuspage
Search
elcome to the admin console.
re is what is going on today.

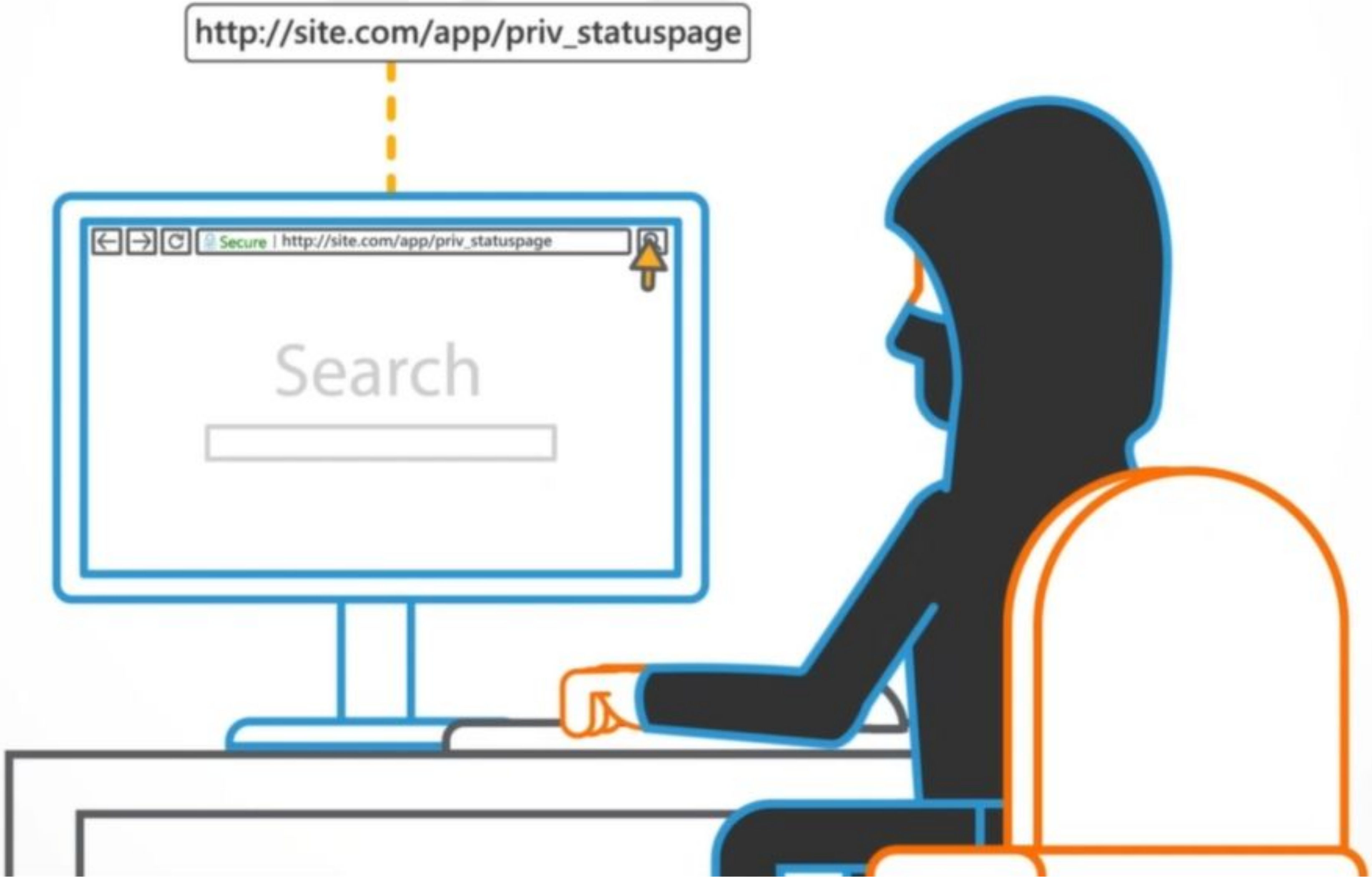
Function level access control is missing when access checks have not been implemented or when a protection mechanism exists but is not properly configured.



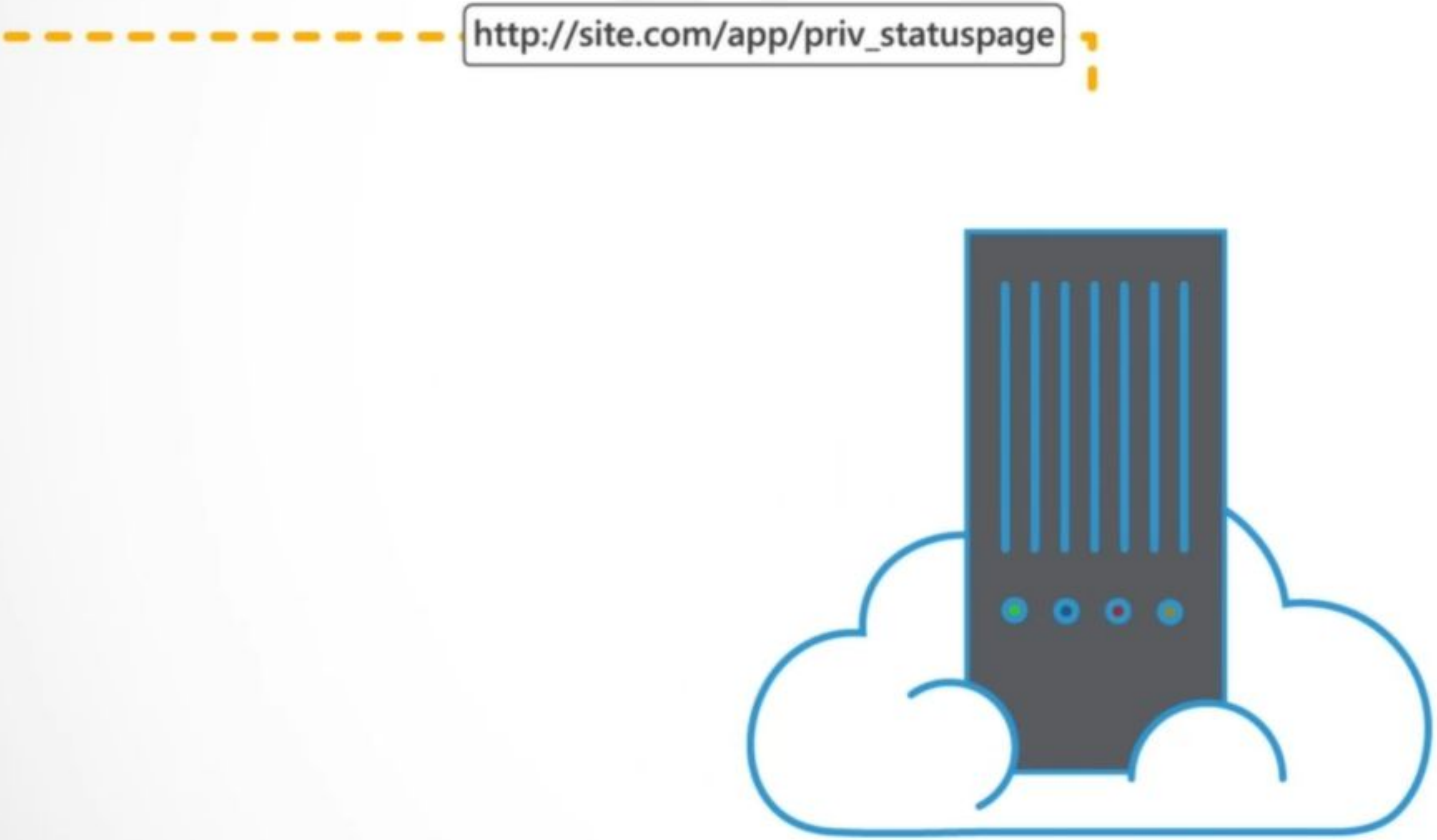
User 'attacker' created.
Login: attacker, password: 3GYT!6

FORCED BROWSING URLS

In this case, an attacker is an authenticated user of a site.

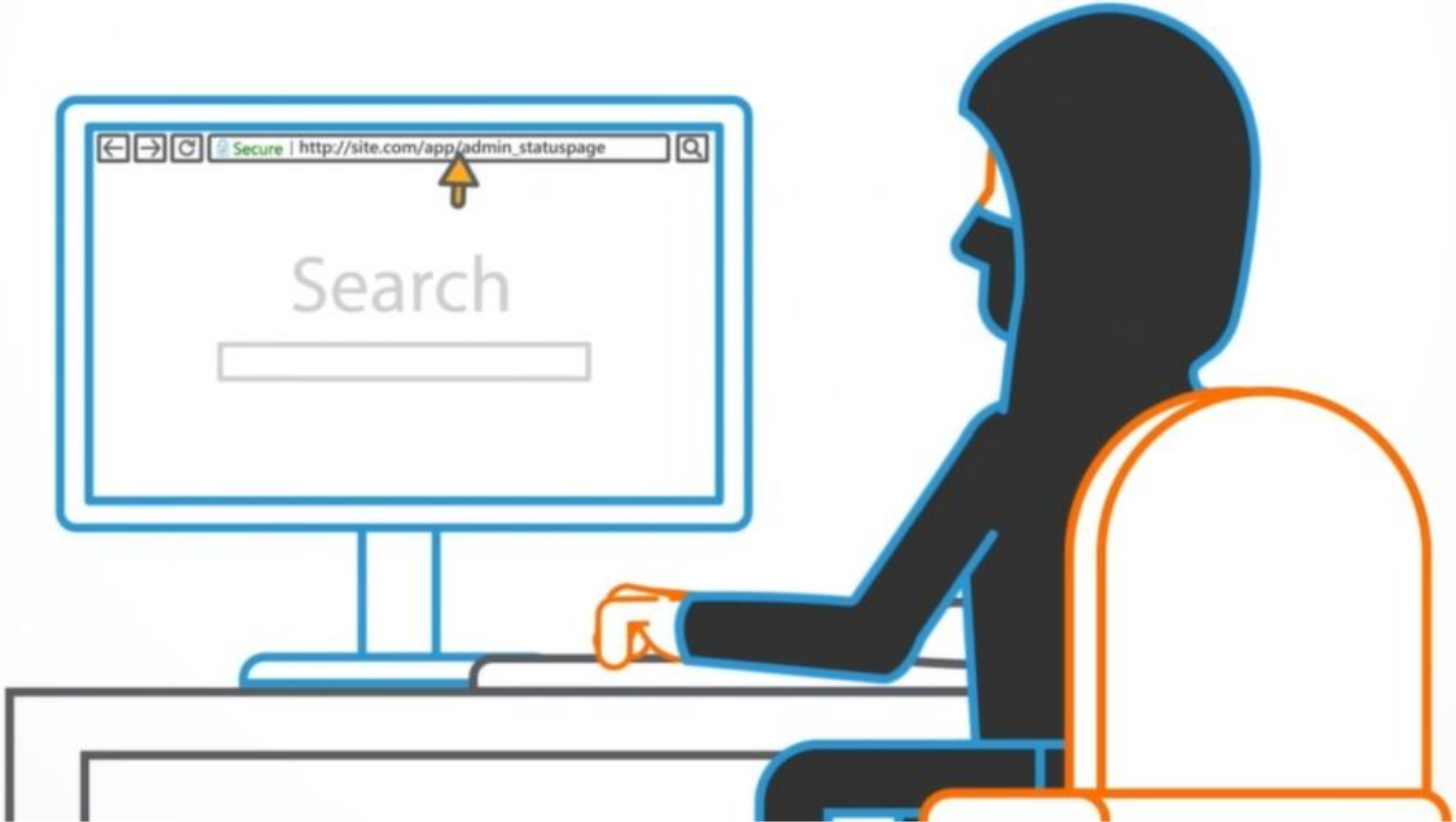

http://site.com/app/priv_statuspage
Secure | http://site.com/app/priv_statuspage
Search

They’re trying to gain elevated access to the application.



http://site.com/app/priv_statuspage

By either guessing or brute forcing URLs



they are able to find an unprotected administrative page.

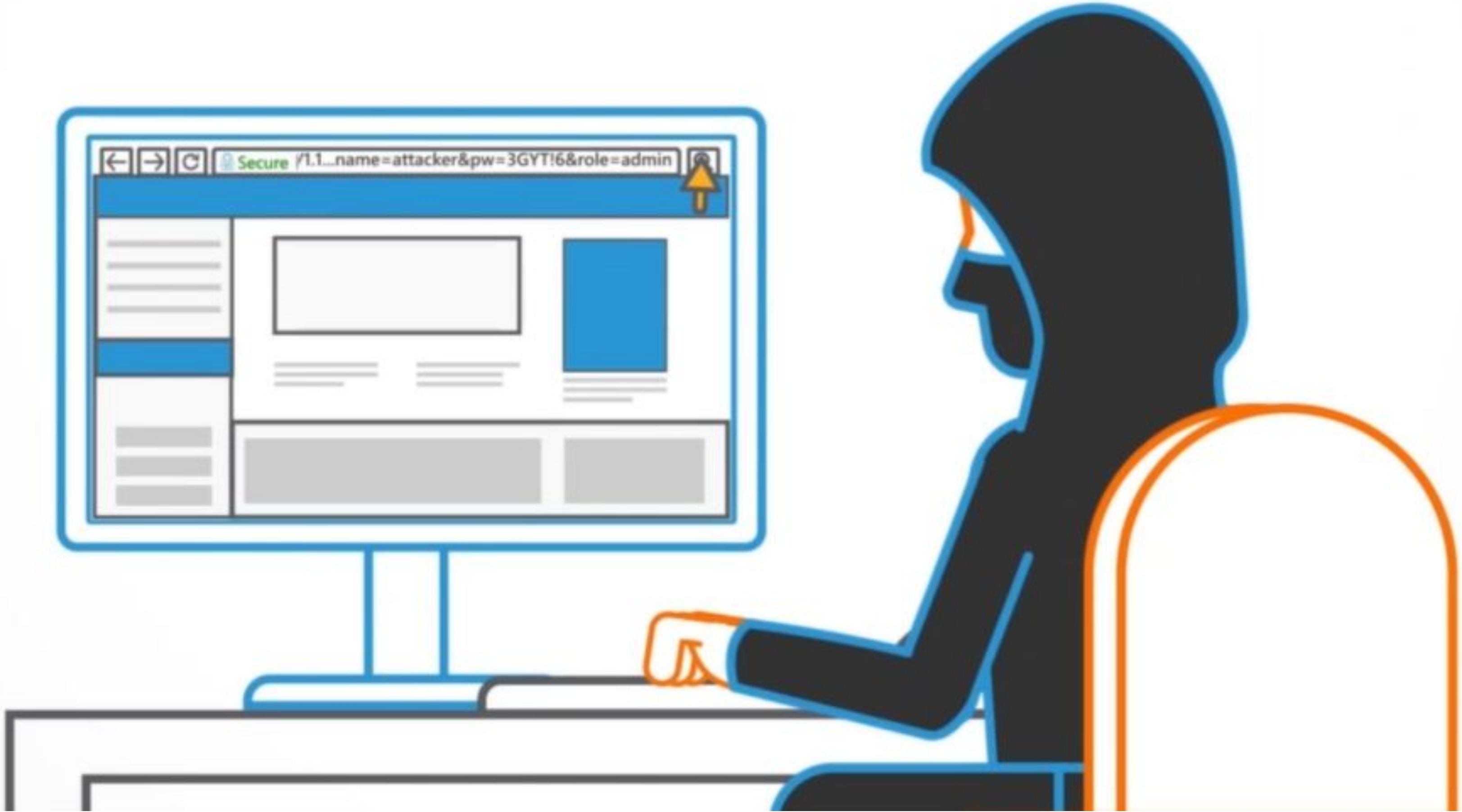
http://site.com/app/admin_statuspage

They browse to this page and can now access a status page that is only intended for administrators to view…

In our next example

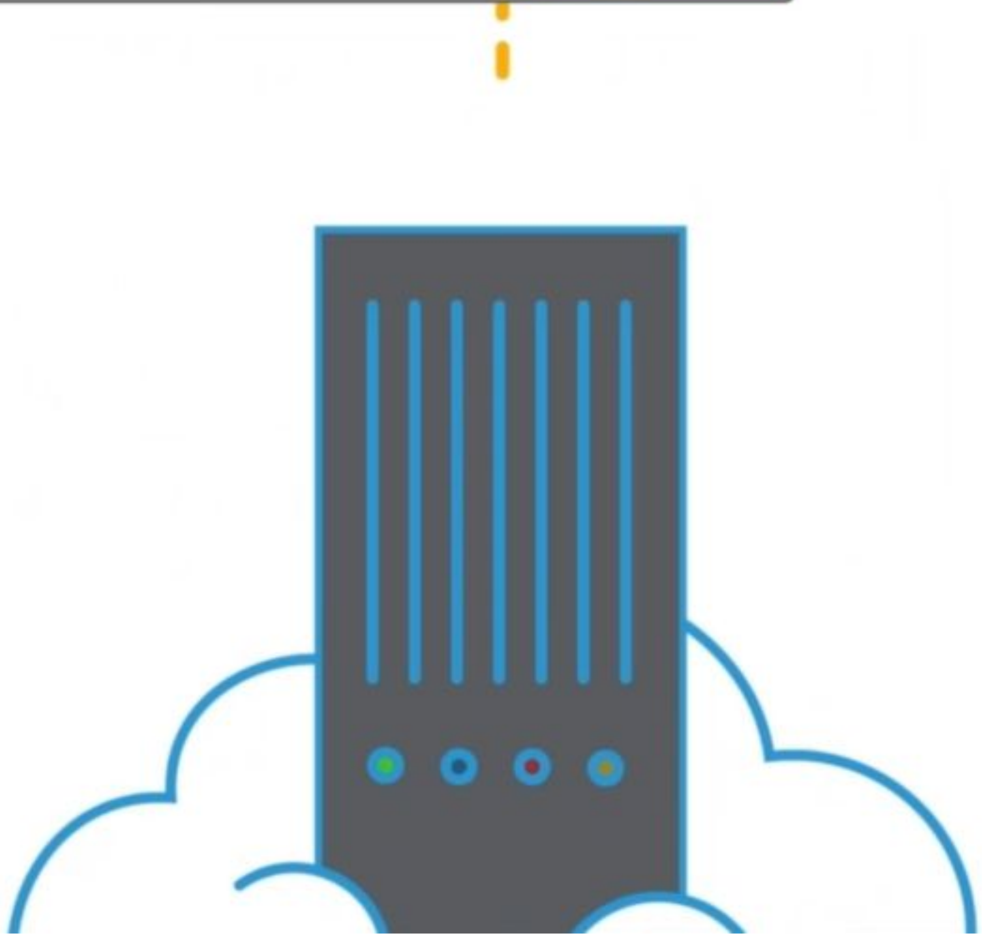
an attacker accessing unauthorized functions

An attacker is an authenticated user of a site which uses a popular framework.

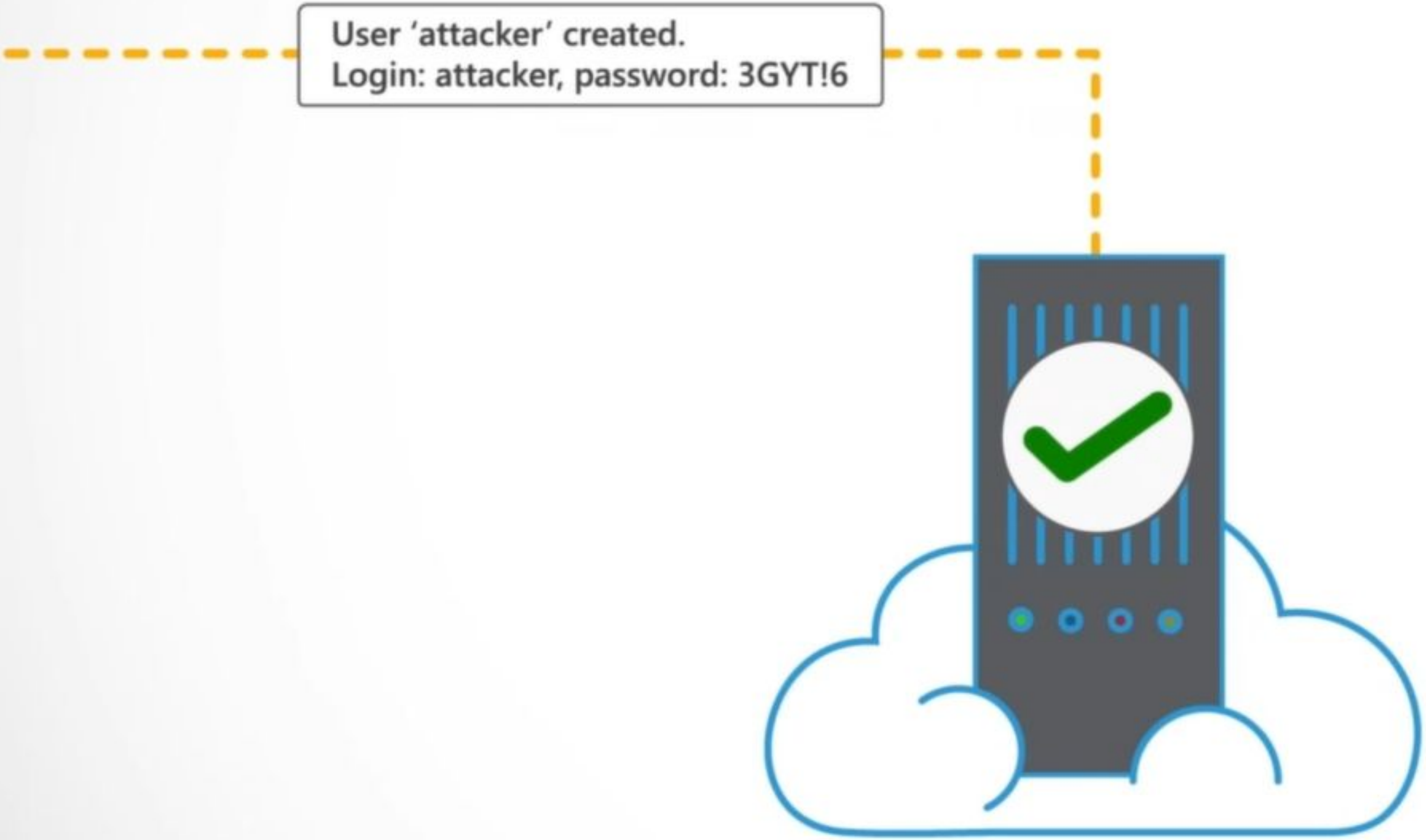


Knowing the framework, the attacker crafts a request to create a new user with elevated permissions.

```
POST/action/createUser HTTP/1.1...
name=attacker&pw=3GYT!6&role=admin
```

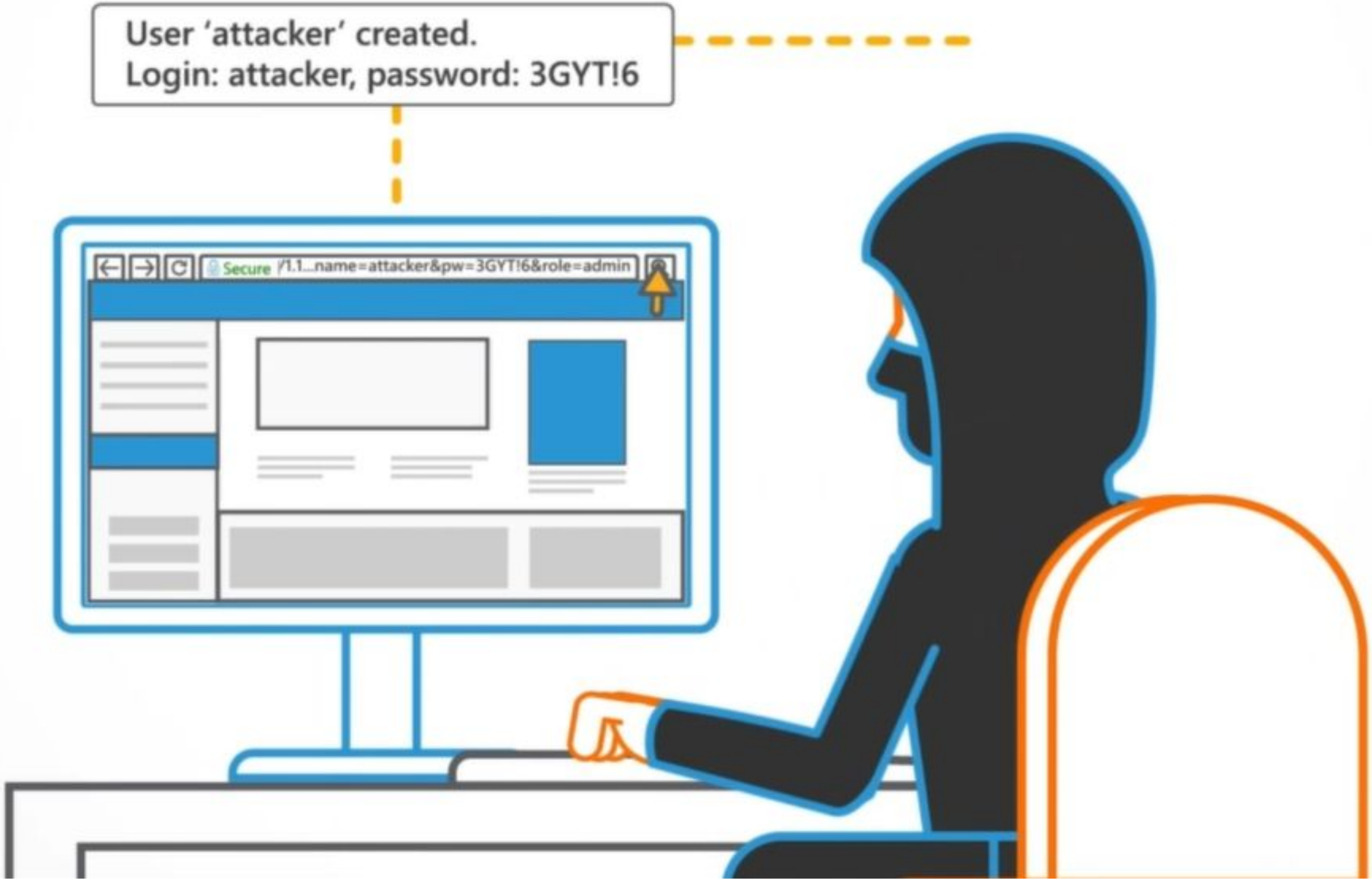


Since the ‘createUser’ function is not properly protected by control checks, the request succeeds and a new user is created.



User ‘attacker’ created.
Login: attacker, password: 3GYT!6

The attacker logs in using the credentials of the newly created admin user



User 'attacker' created.
Login: attacker, password: 3GYT!6

```
Secure /1.1...name=attacker&pw=3GYT!6&role=admin
```

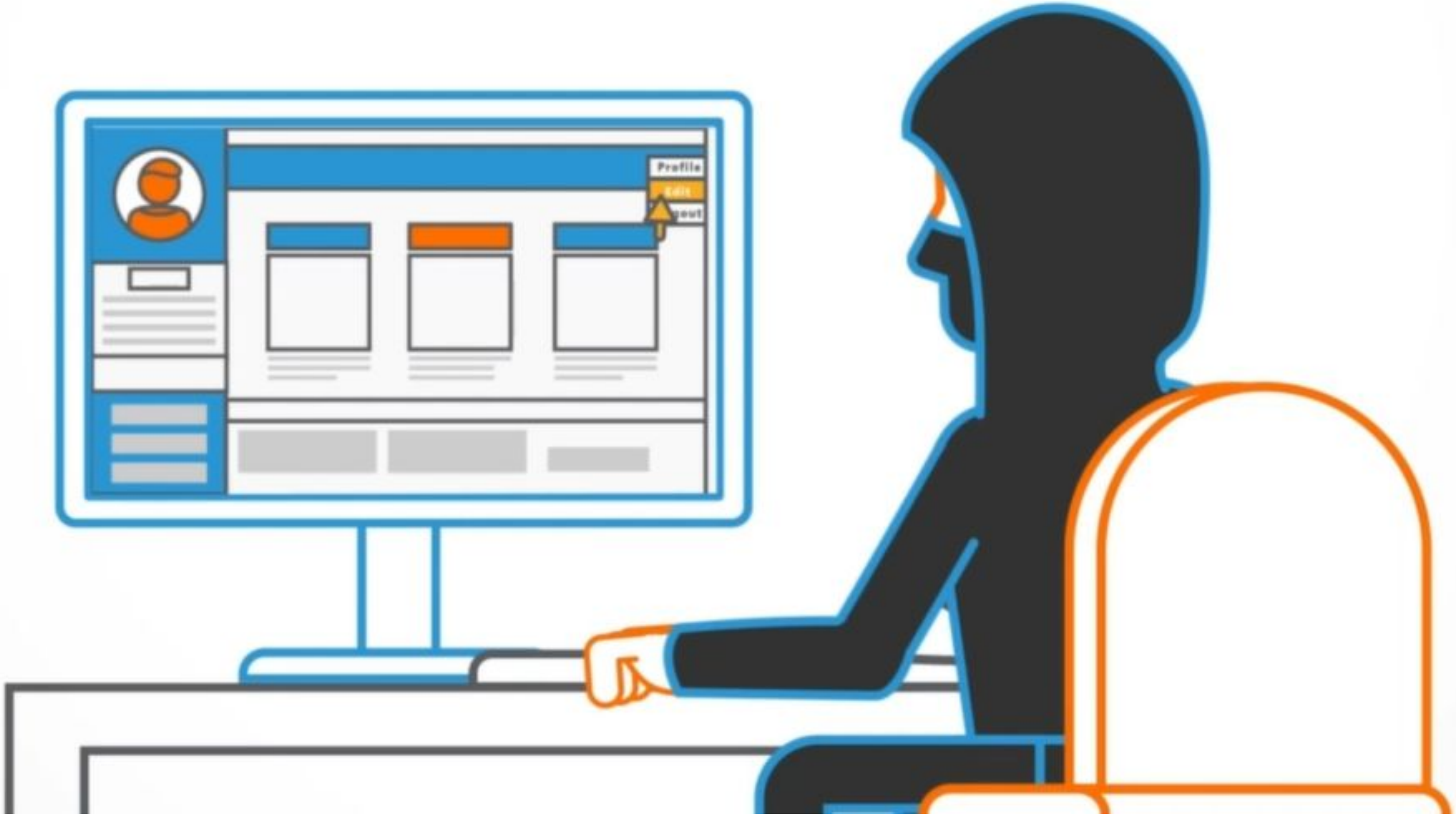
and can now steal all of your customer's data!

/1.1...name=a
ow=3GYT!6&role=admin
HACKED!

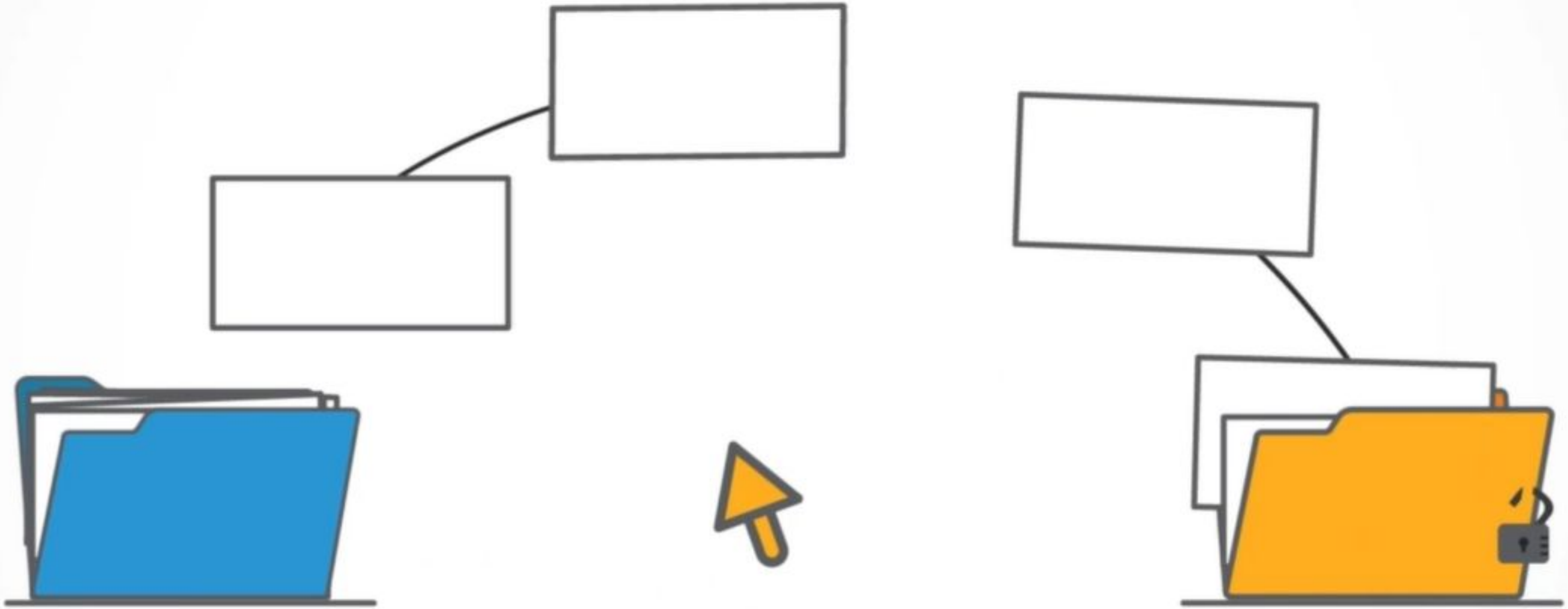
First Name	Last Name	Email Id	[illegible]
John	Doe	john.doe@xyzmail.com	[illegible]
David	Stuart	david.stuart@xyzmail.com	D4v!d$tuart
Jenny	Helen	jenny.helen@xyzmail.com	J3nny#eLen
Robert	Brian	david.stuart@xyzmail.com	R0Bert8r4!n
Linda	Smith	linda.smith@xyzmail.com	L!nd4$m!t#
Mark	Jason	mark.jason@xyzmail.com	m4rkJ4$0n
Betty	Jones	betty.jones@xyzmail.com	8ettyJ0ne$

“MISSING FUNCTION LEVEL ACCESS CONTROL” VULNERABILITIES CAN HAVE SIGNIFICANT IMPACTS

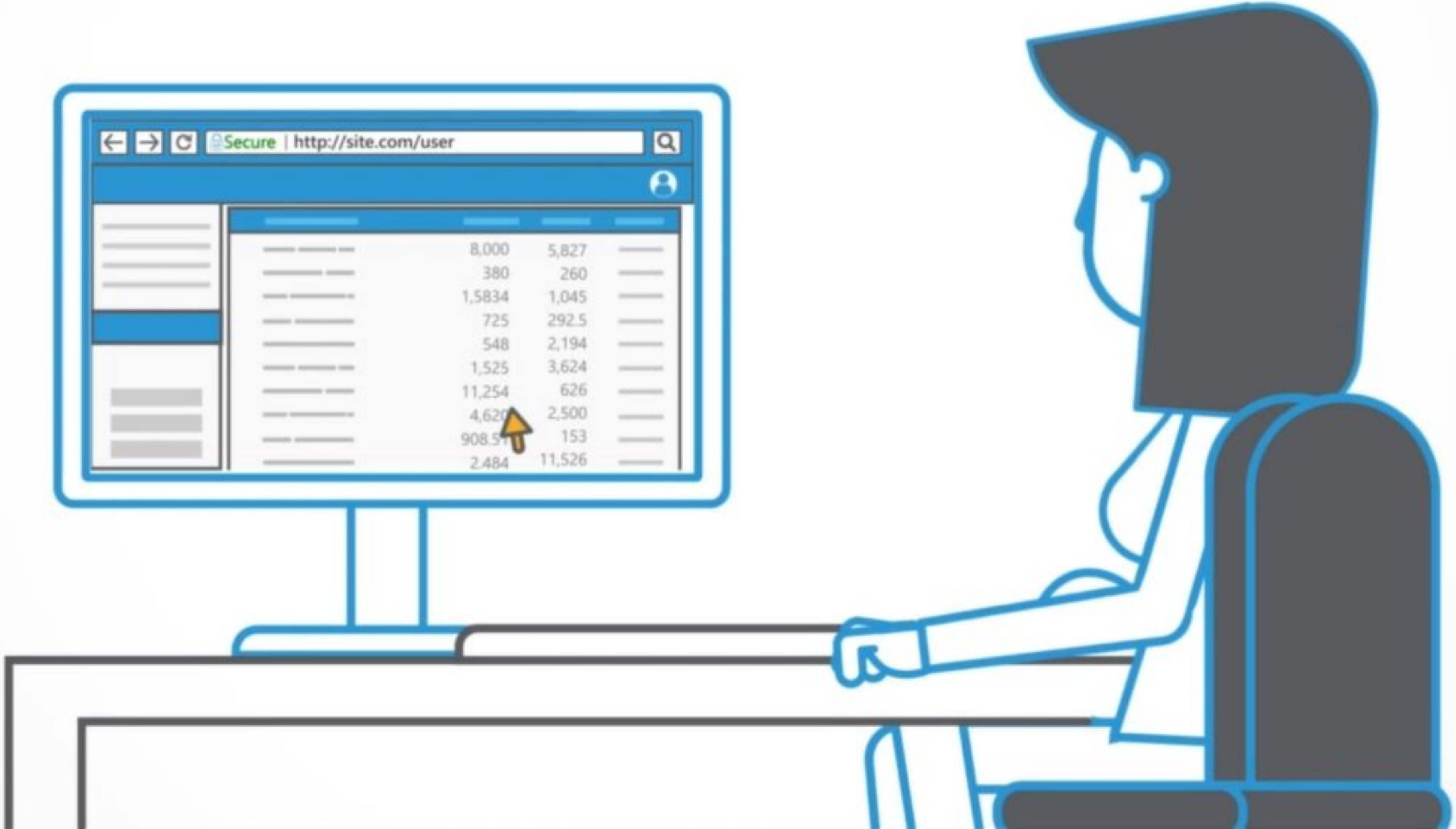
Accounts could be taken over, including privileged ones. With a stolen account, the attacker could do anything the victim could do.



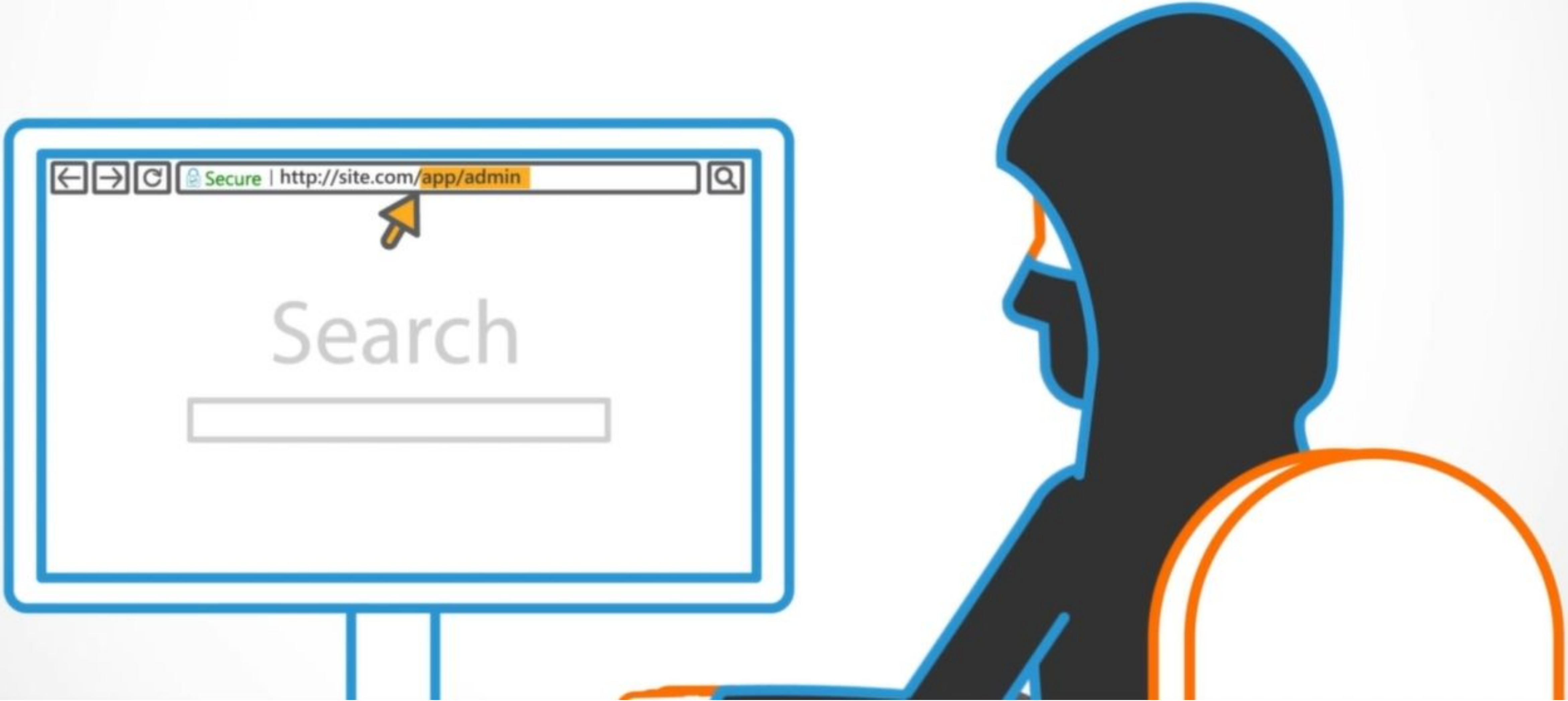
Sensitive end-user or customer data could be stolen, leading to reputational damage and revenue loss.



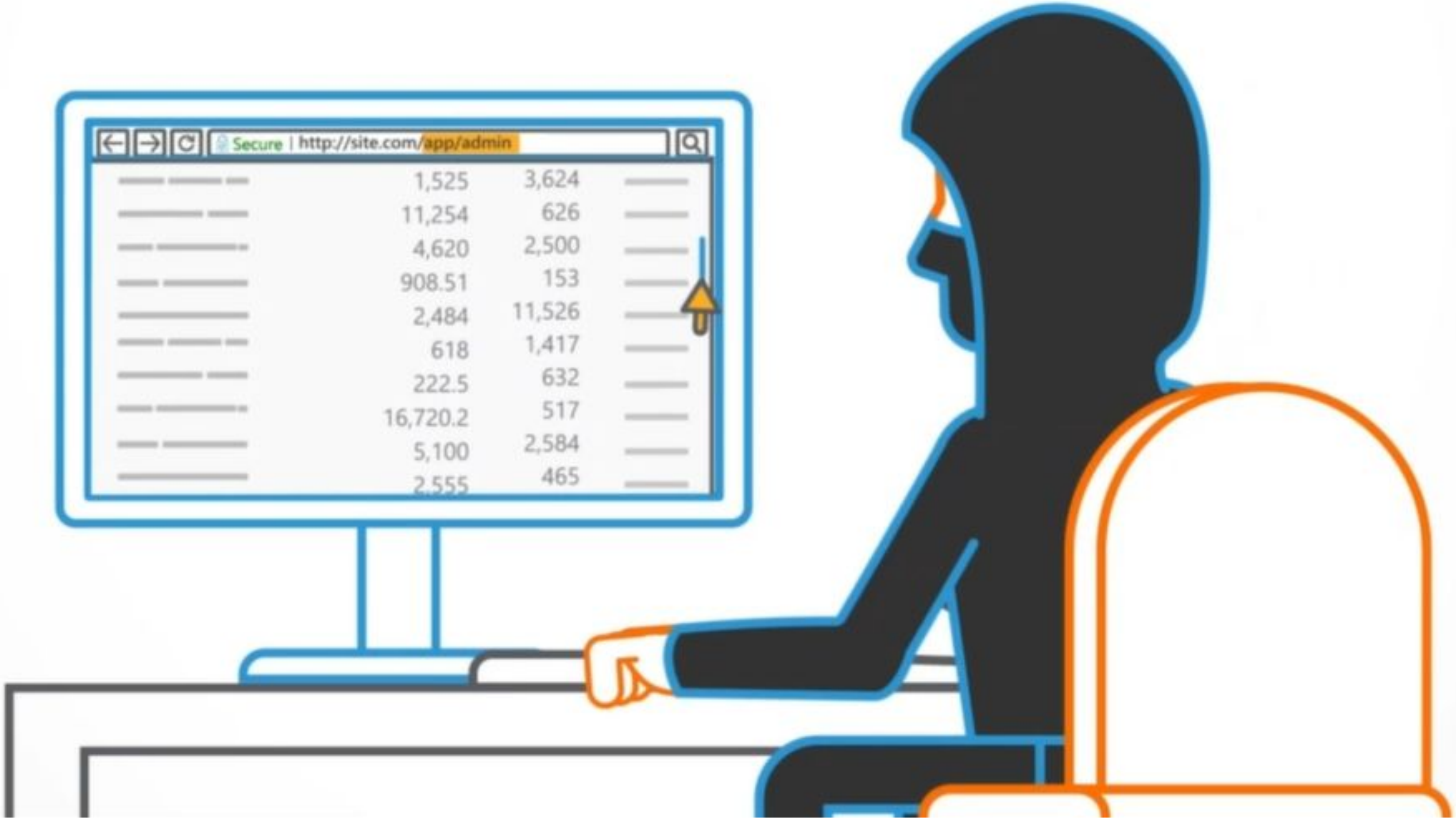
An employee from the sales department could view information from the financial department.



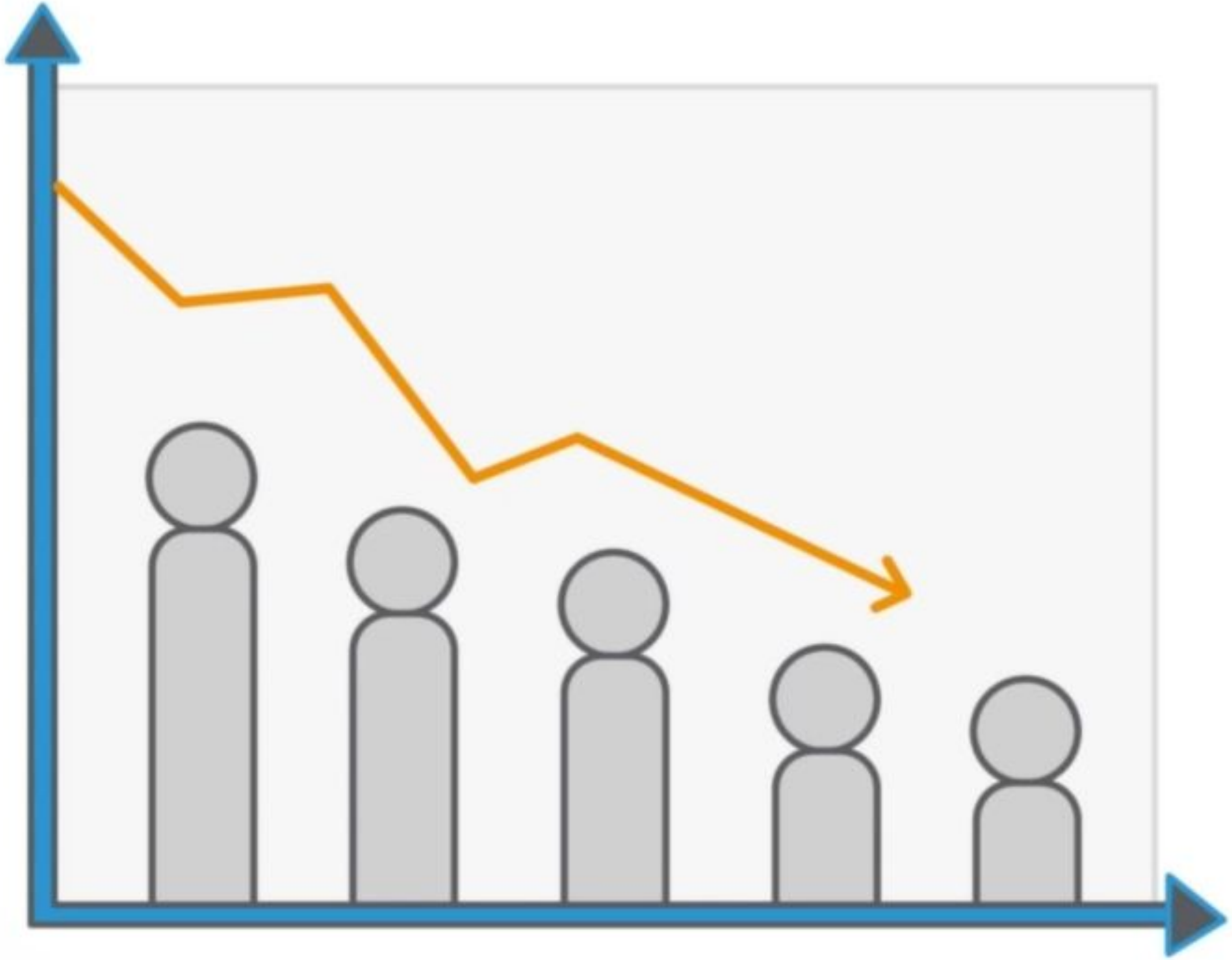
An attacker could forge requests in order to access functionality without proper authorization.



An attacker could also gain access to the administrative panel of your application,



leading to disruption of the website, causing loss of customers and revenue.



To prevent "Missing Function Level Access Control"

- Protect all business functions using a role based authorization mechanism, implemented on the server side

- Authorization should be applied using centralized routines either provided by the framework or easy to use external modules

To prevent “Missing Function Level Access Control”

- Always deny access by default. See the “Least Privilege” module for more information on this
- Implement function access control on the server, never the client.

Congratulations,

you have now completed this module, Missing Function Level Access Control!

SECURE
CODE
WARRIOR
®