XML external entity (XXE) Injection

<xml />

XML external entity (XXE) Injection

XML? → eXtensible Markup Language.

<xml />

XML external entity (XXE) Injection

XML? → eXtensible Markup Language.
Use? → Store and transport data.

<xml />

XML external entity (XXE) Injection

XML? → eXtensible Markup Language.
Use? → Store and transport date.

Example:

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

<xml />

XML external entity (XXE) Injection

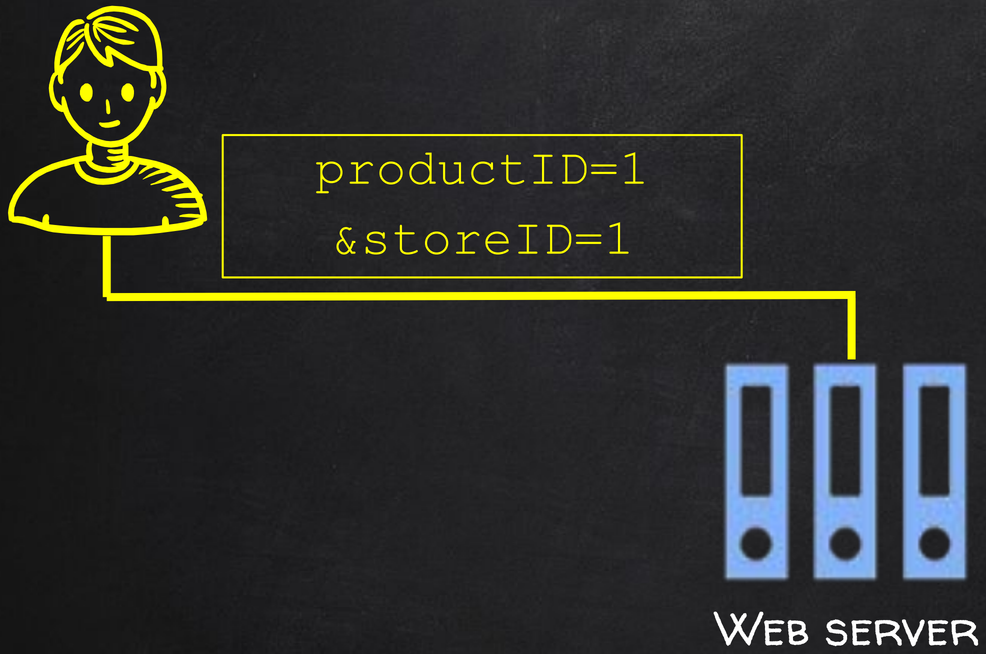
XML? → eXtensible Markup Language.
Use? → Store and transport date.

Example:

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

→ no work is done above!

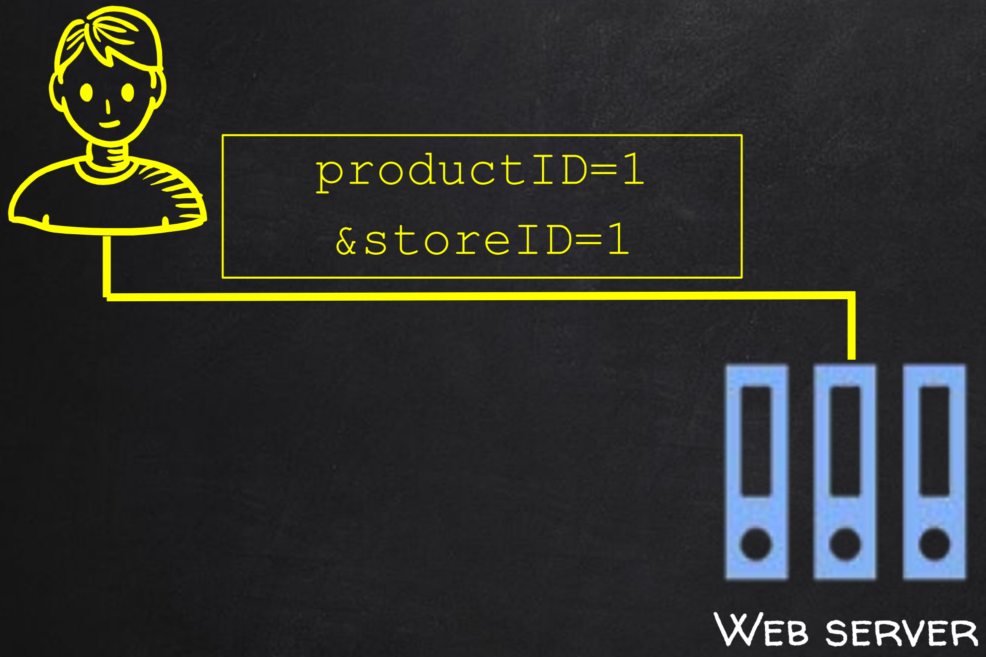
<xml />



productID=1
&storeID=1

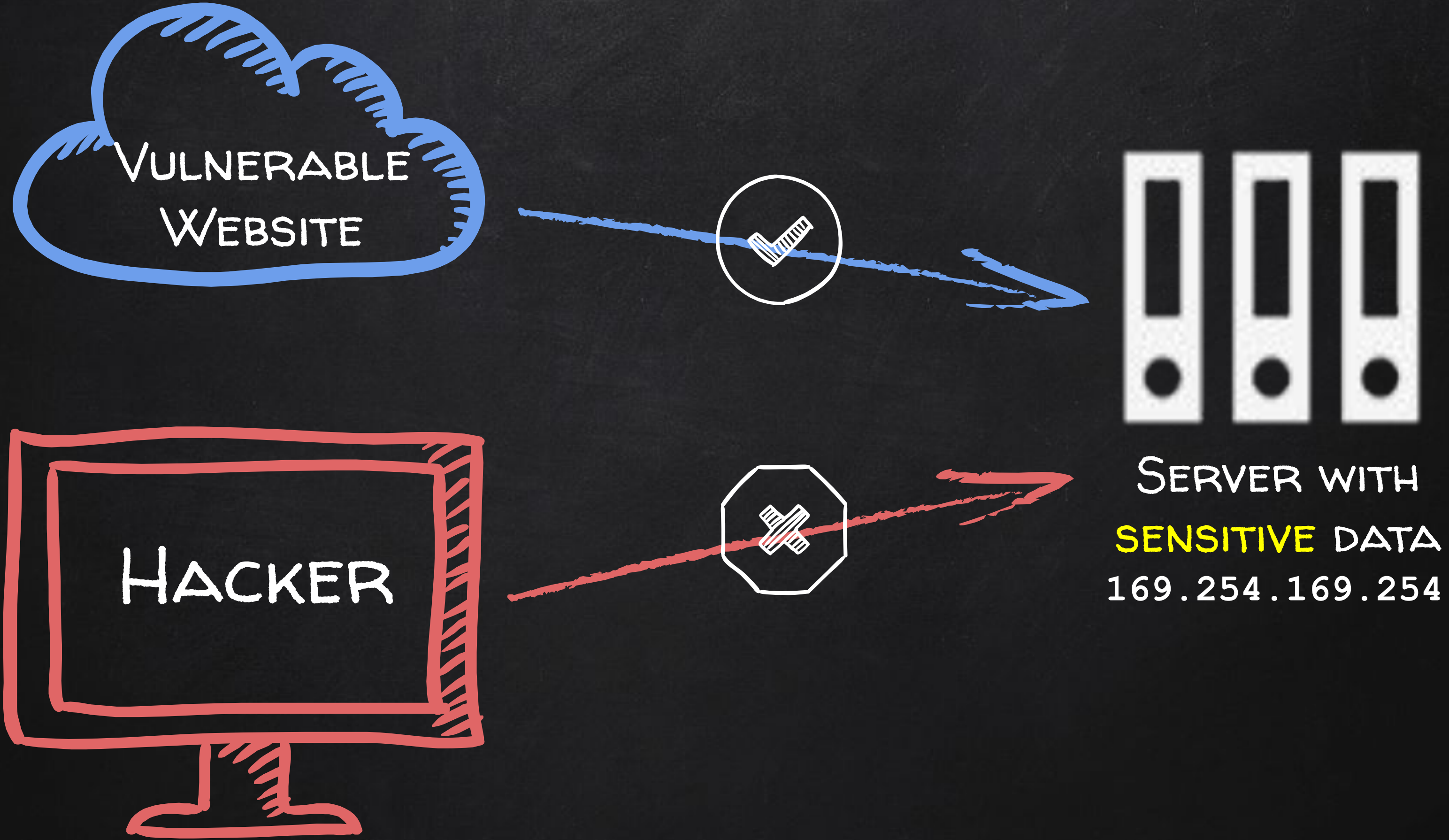
Web server

```
<stockCheck>
   <productId>1</productId>
   <storeId>1</storeId>
</stockCheck>
```



```
productID=1
&storeID=1
```

Web server



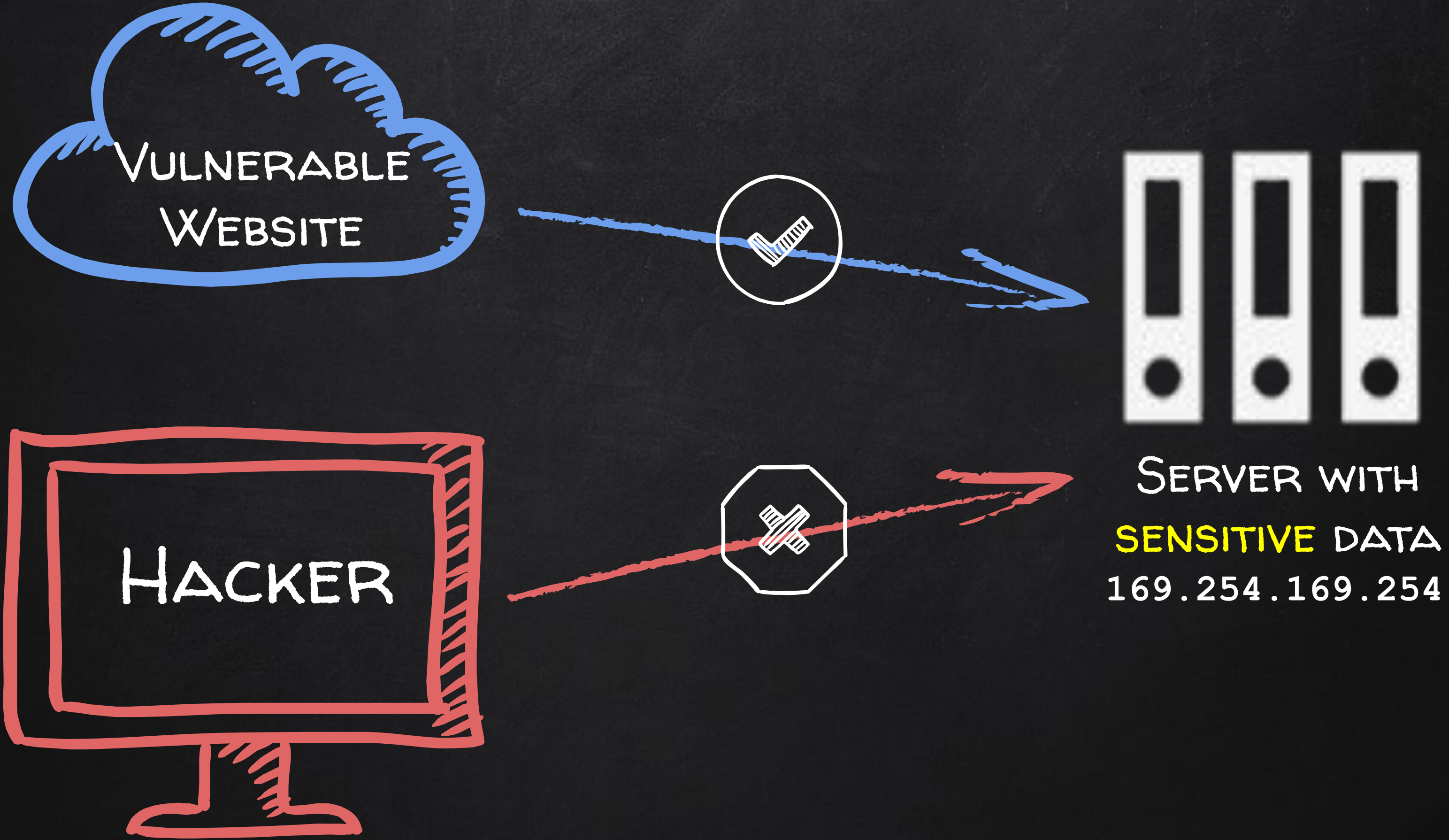
Vulnerable Website

Hacker

Server with

sensitive data

169.254.169.254



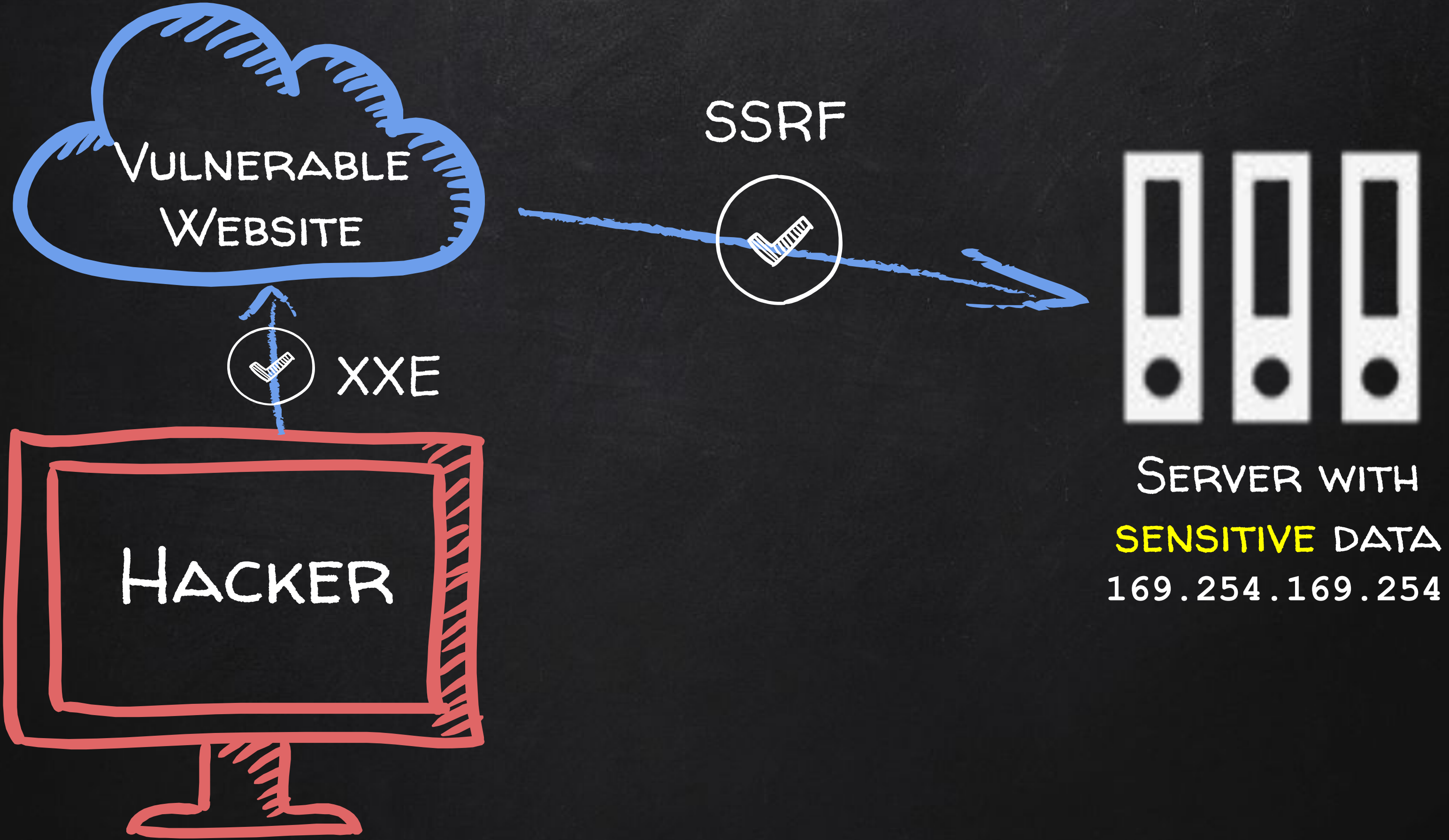
Vulnerable Website

Hacker

Server with

sensitive data

169.254.169.254



Vulnerable Website

SSRF

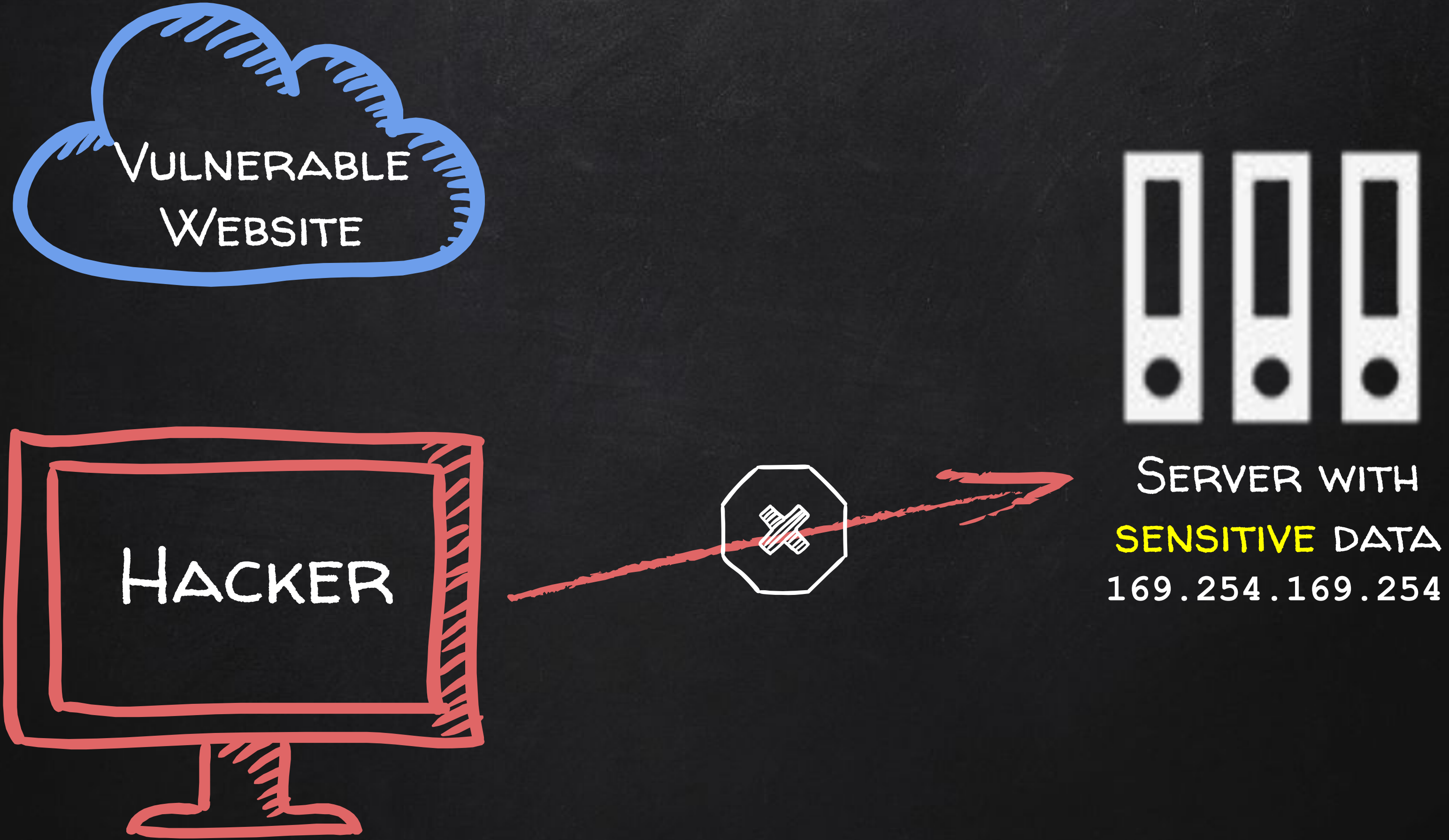
XXE

Hacker

Server with

Sensitive data

169.254.169.254

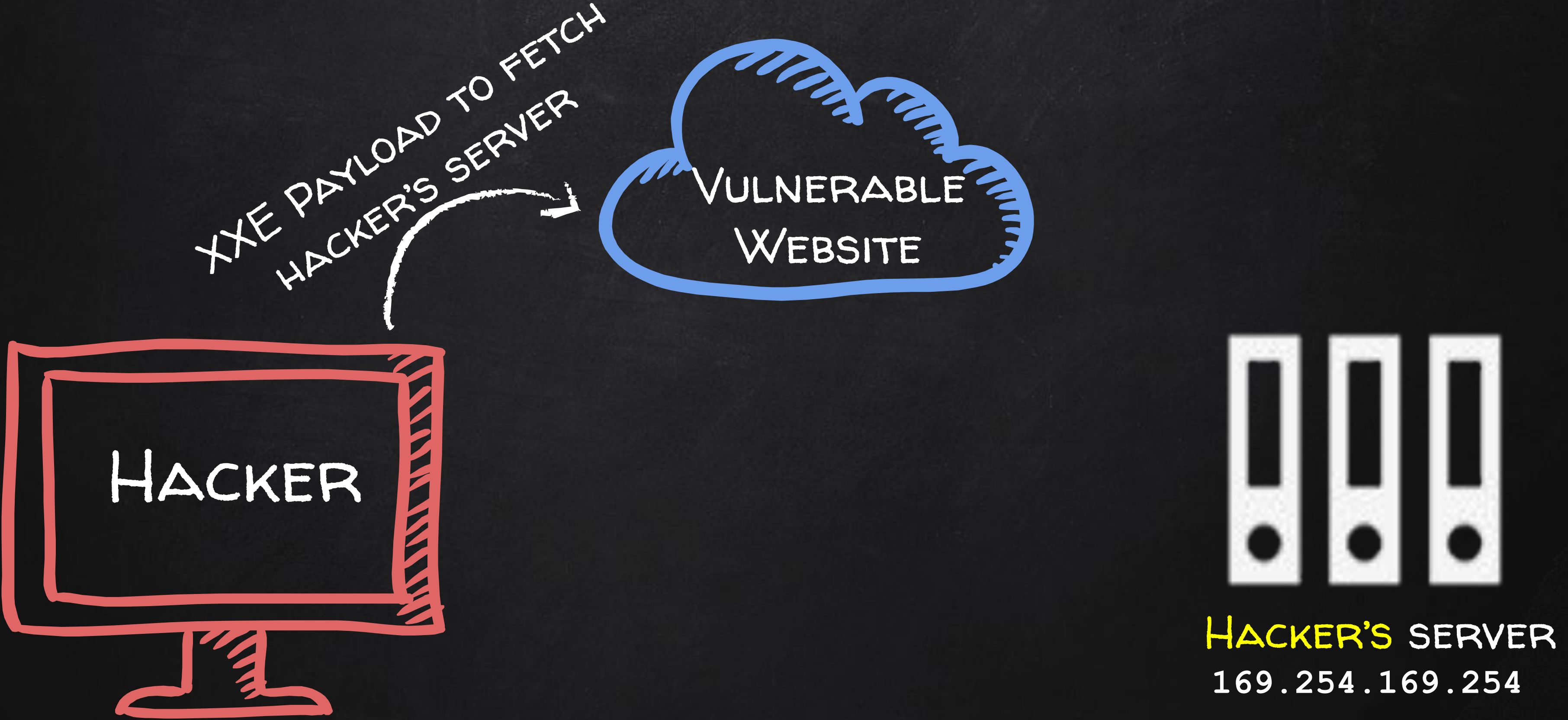


Vulnerable Website

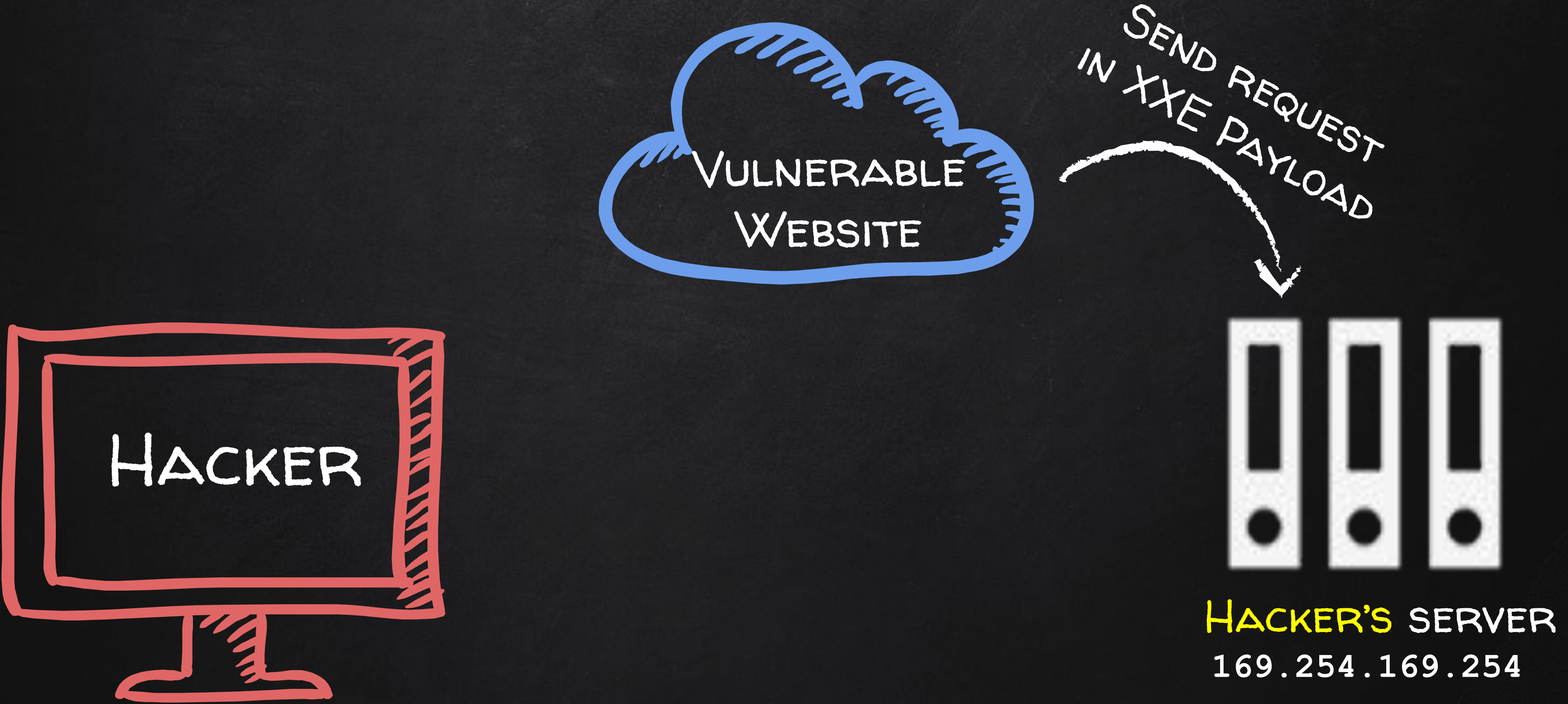
Hacker

Server with sensitive data 169.254.169.254

Out of bound XXE



Out of bound XXE



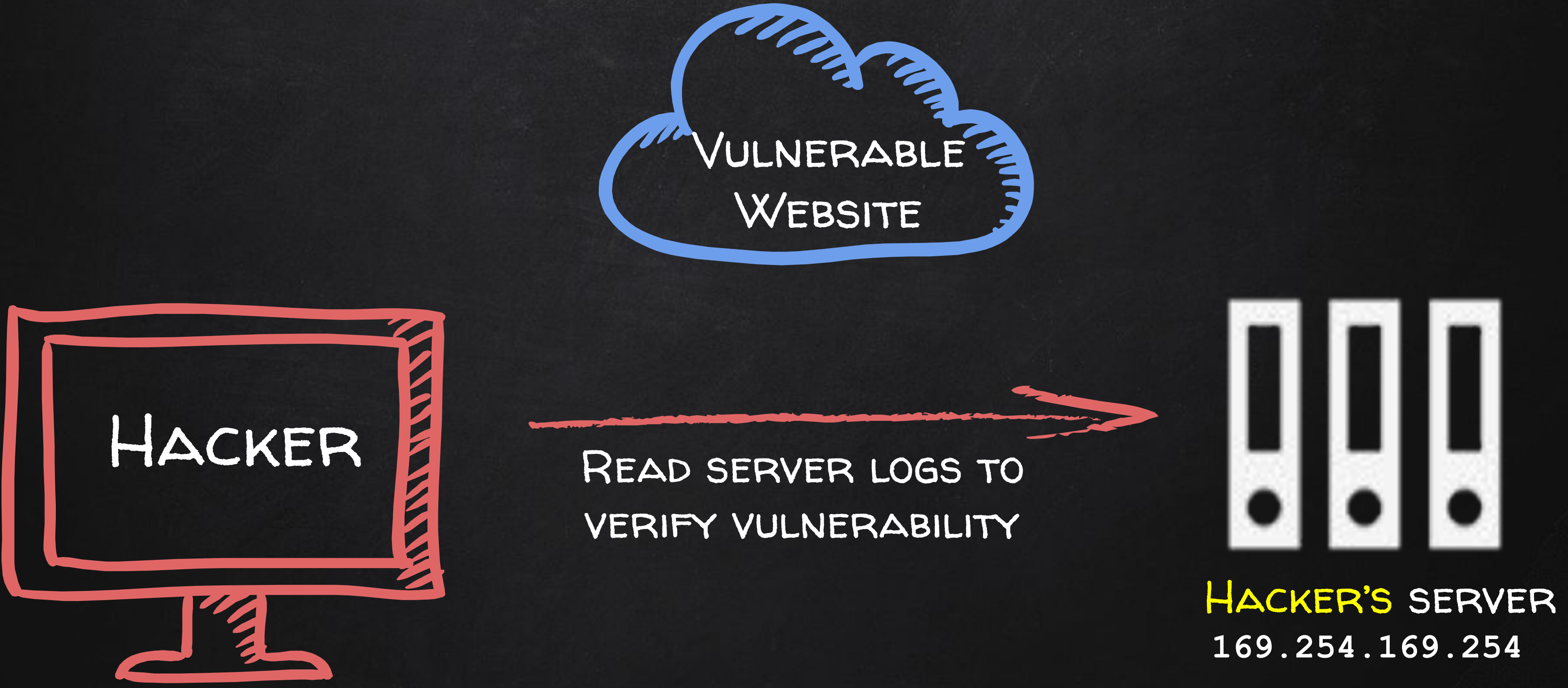
Vulnerable Website

Send request in XXE payload

Hacker

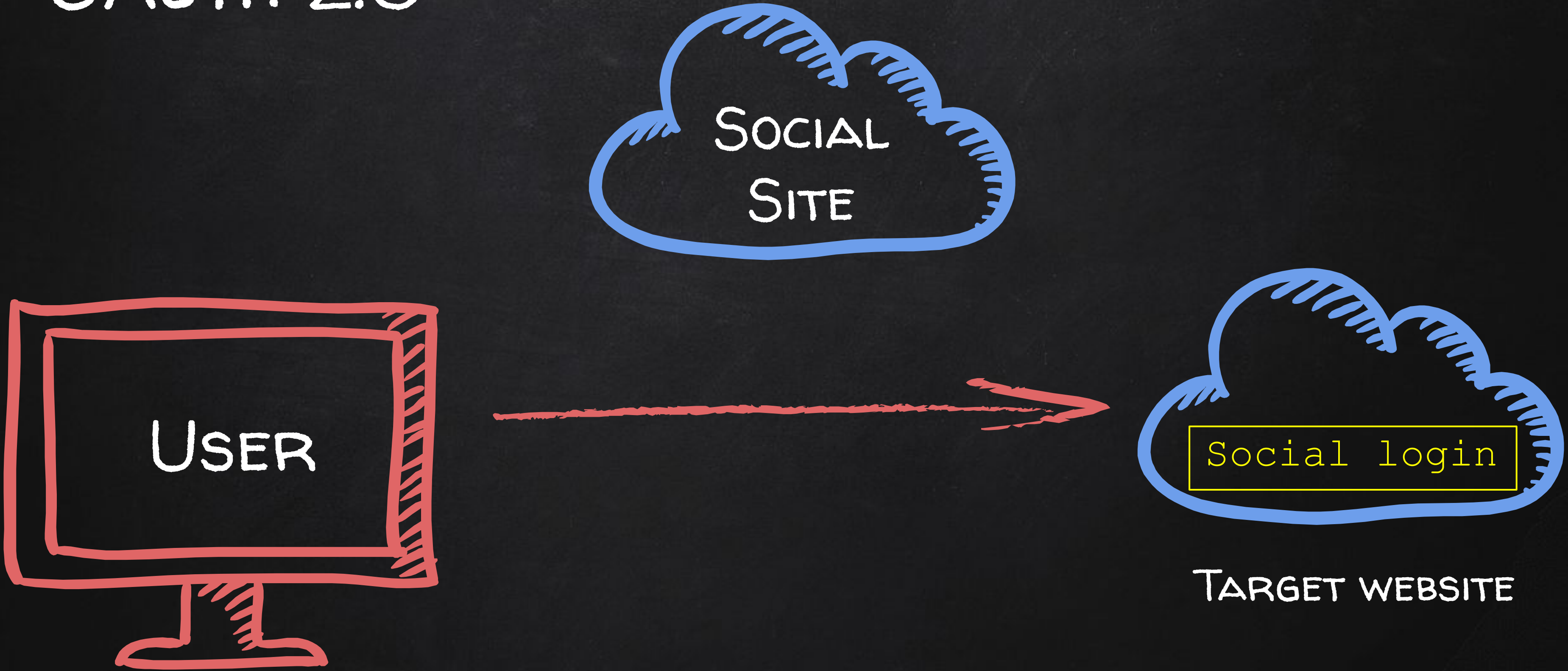
Hacker's server
169.254.169.254

Out of bound XXE



Read server logs to verify vulnerability

OAuth 2.0



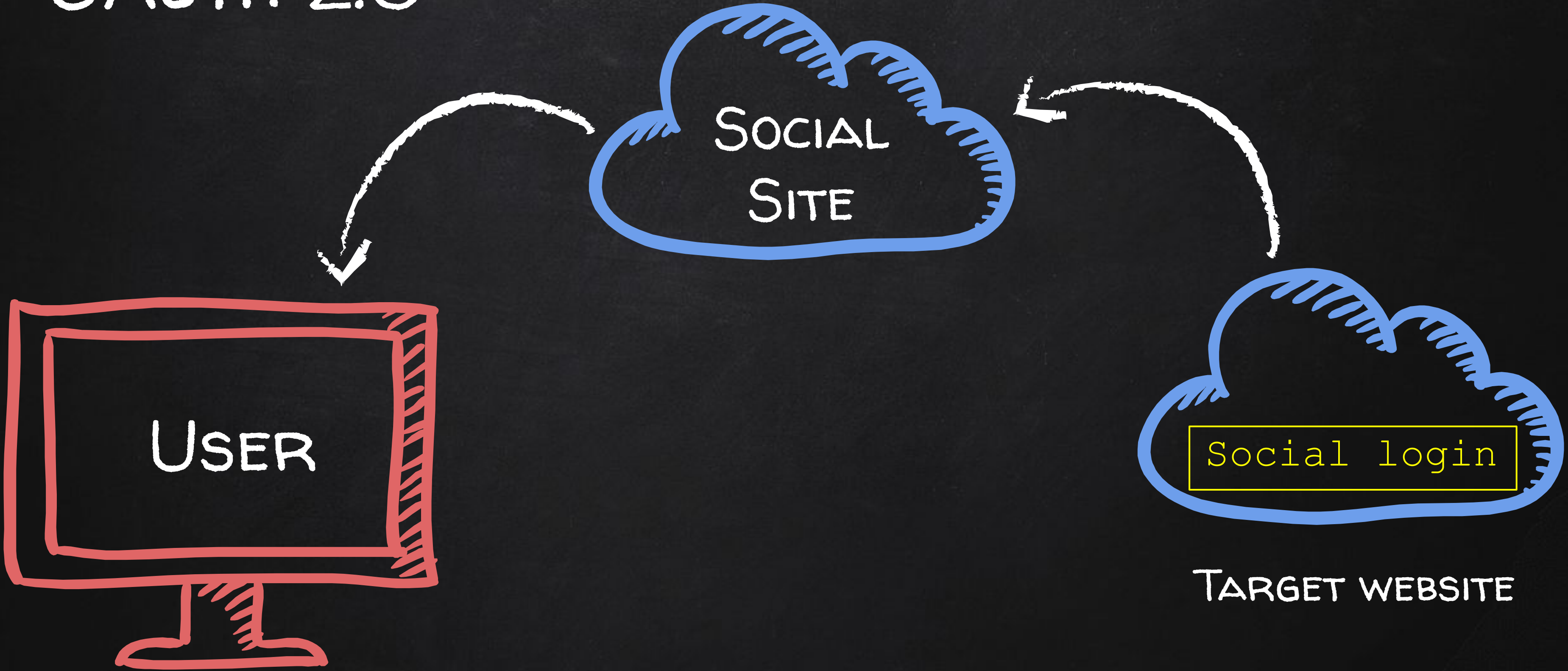
Social Site

User

Social login

Target website

OAuth 2.0



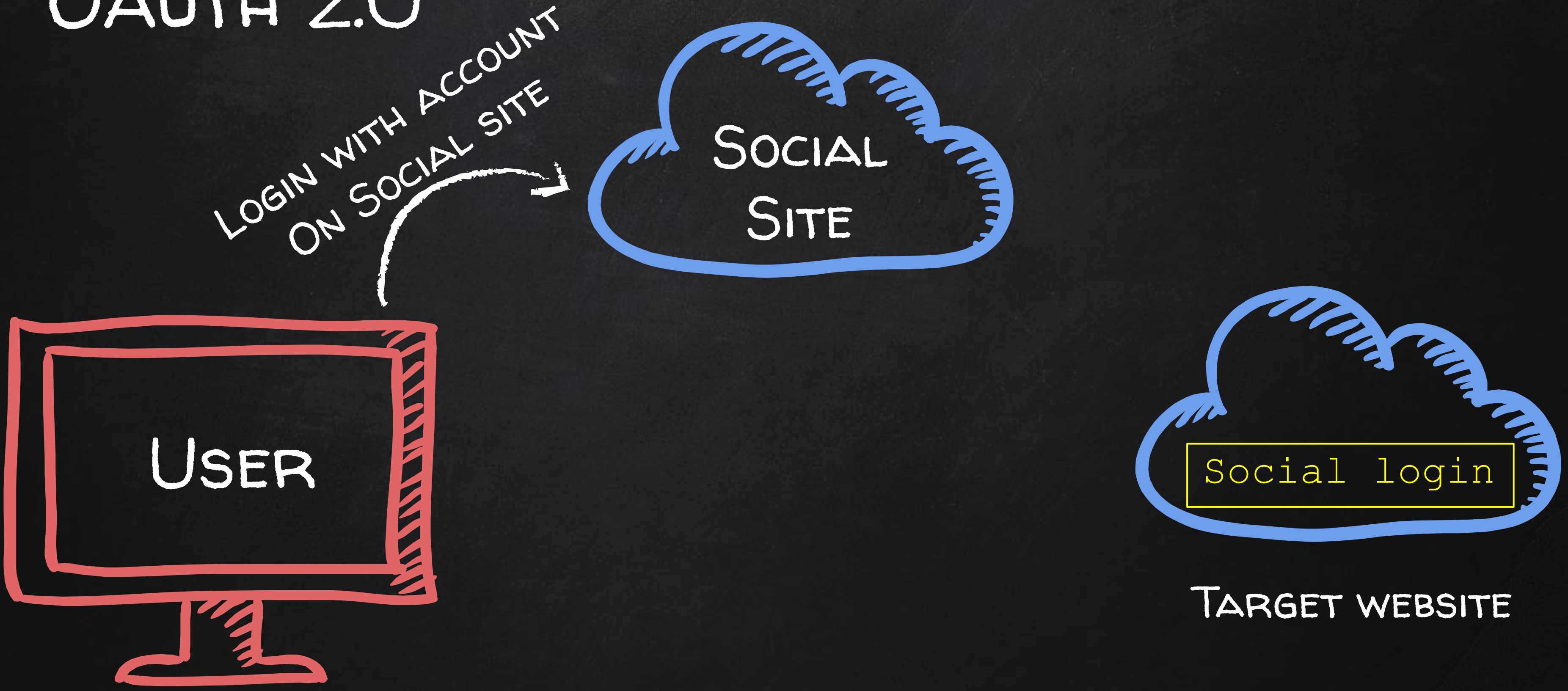
Social Site

User

Social login

Target website

OAuth 2.0



Login with account on Social site

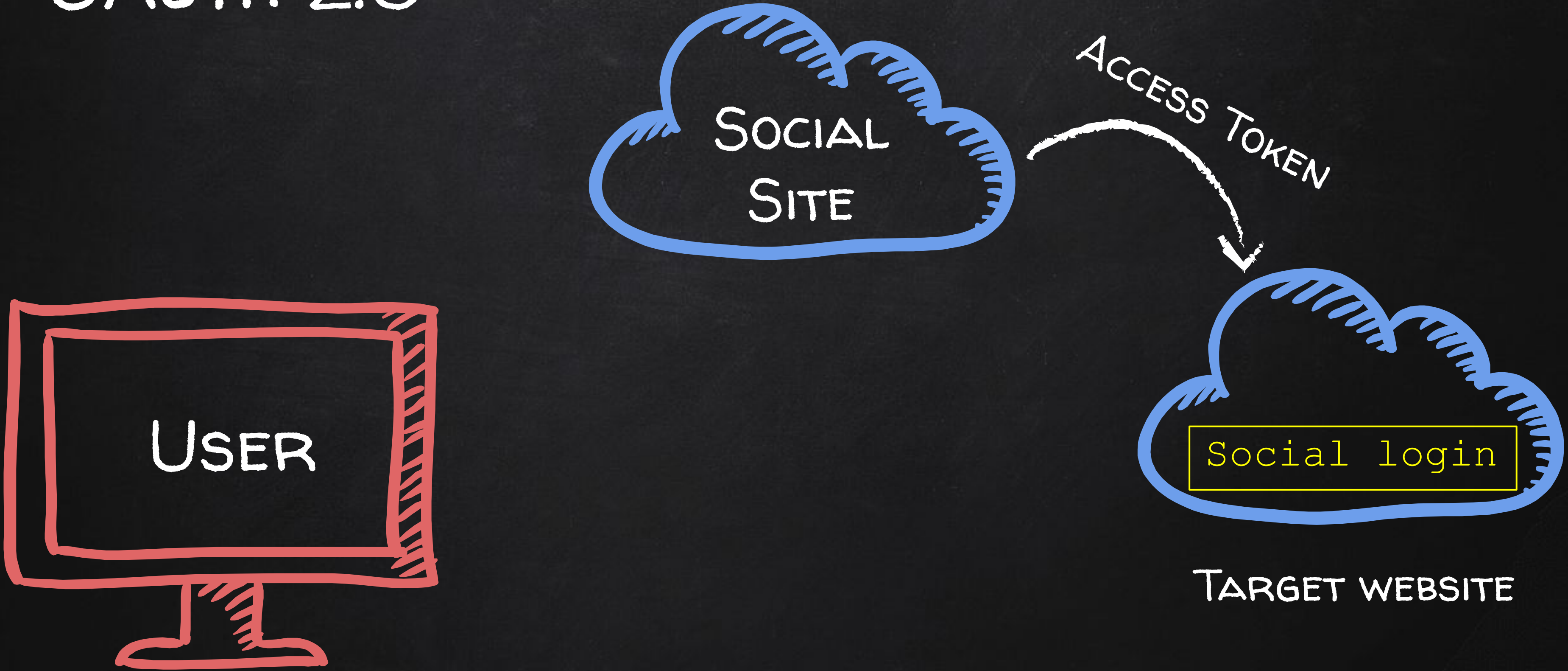
Social Site

User

Social login

Target website

OAuth 2.0



User

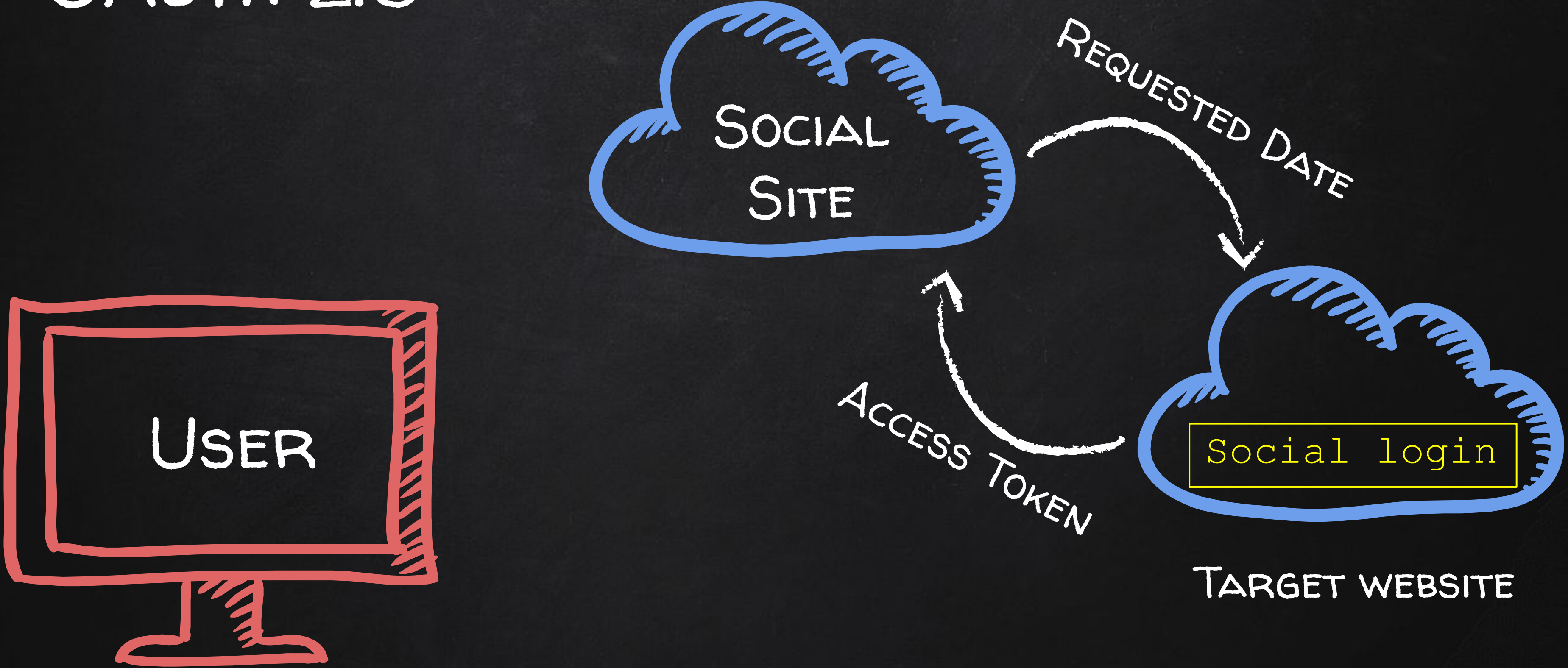
Social Site

Access Token

Social login

Target website

OAuth 2.0



User

Social Site

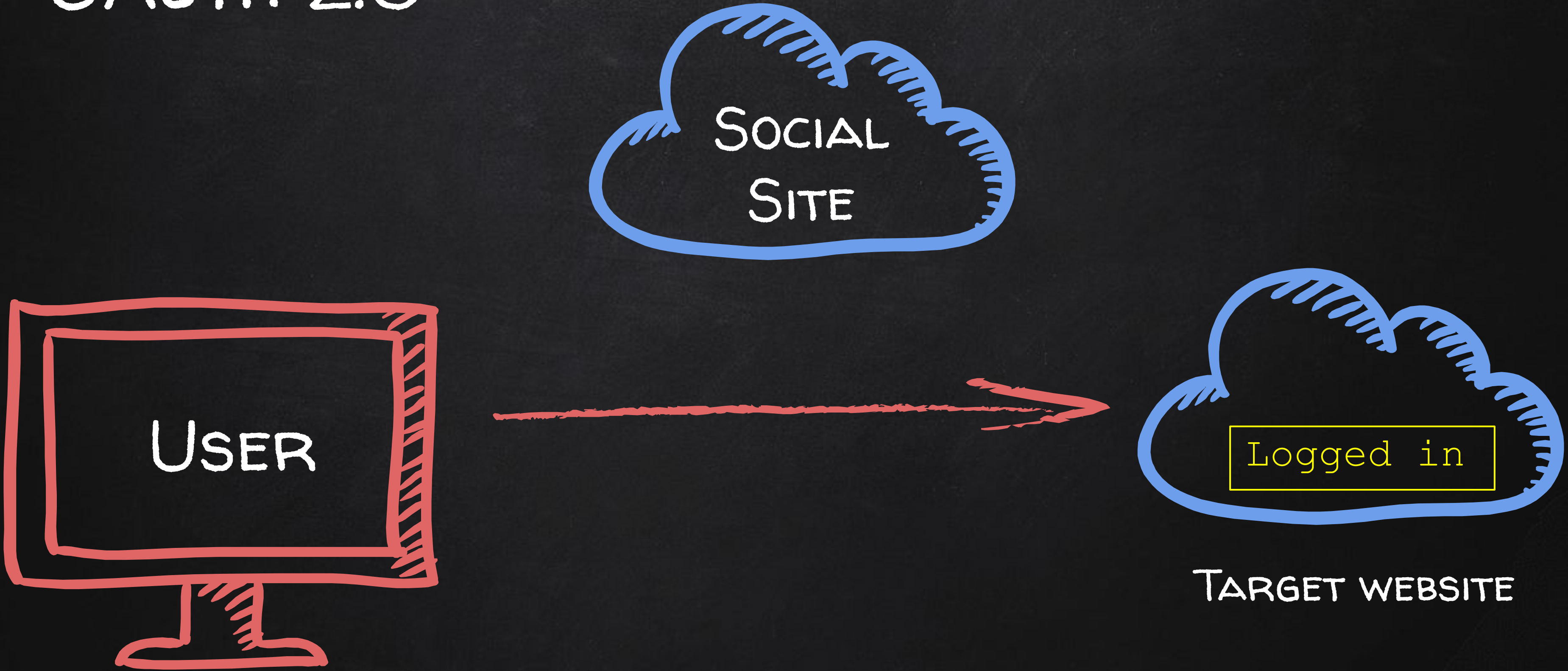
Requested Date

Access Token

Social login

Target website

OAuth 2.0



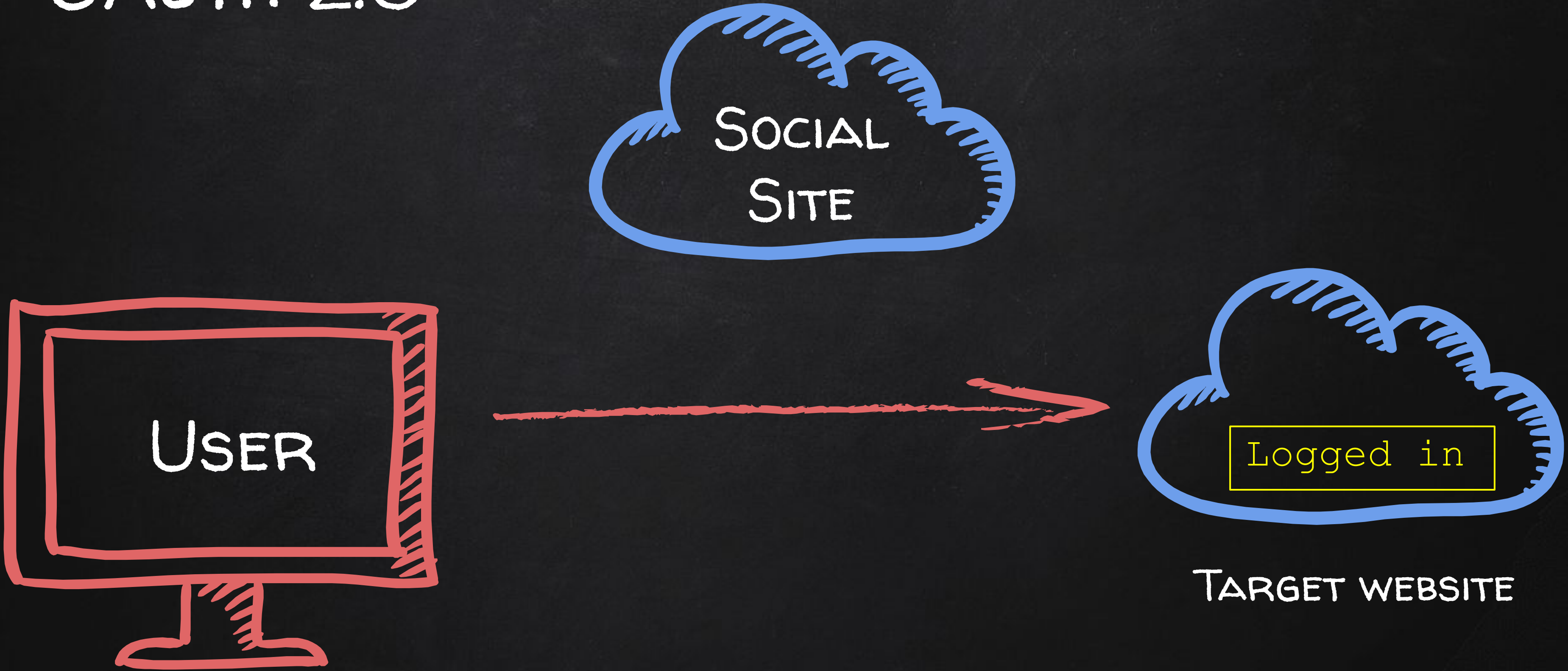
Social Site

User

Logged in

Target website

OAuth 2.0



Cross Site Request Forgery

CSRF

- Requests are not validated at the server side.
- Server does not check if the user generated the request.
- Requests can be forged and sent to users to make them do things they don't intend to do such as changing their password.

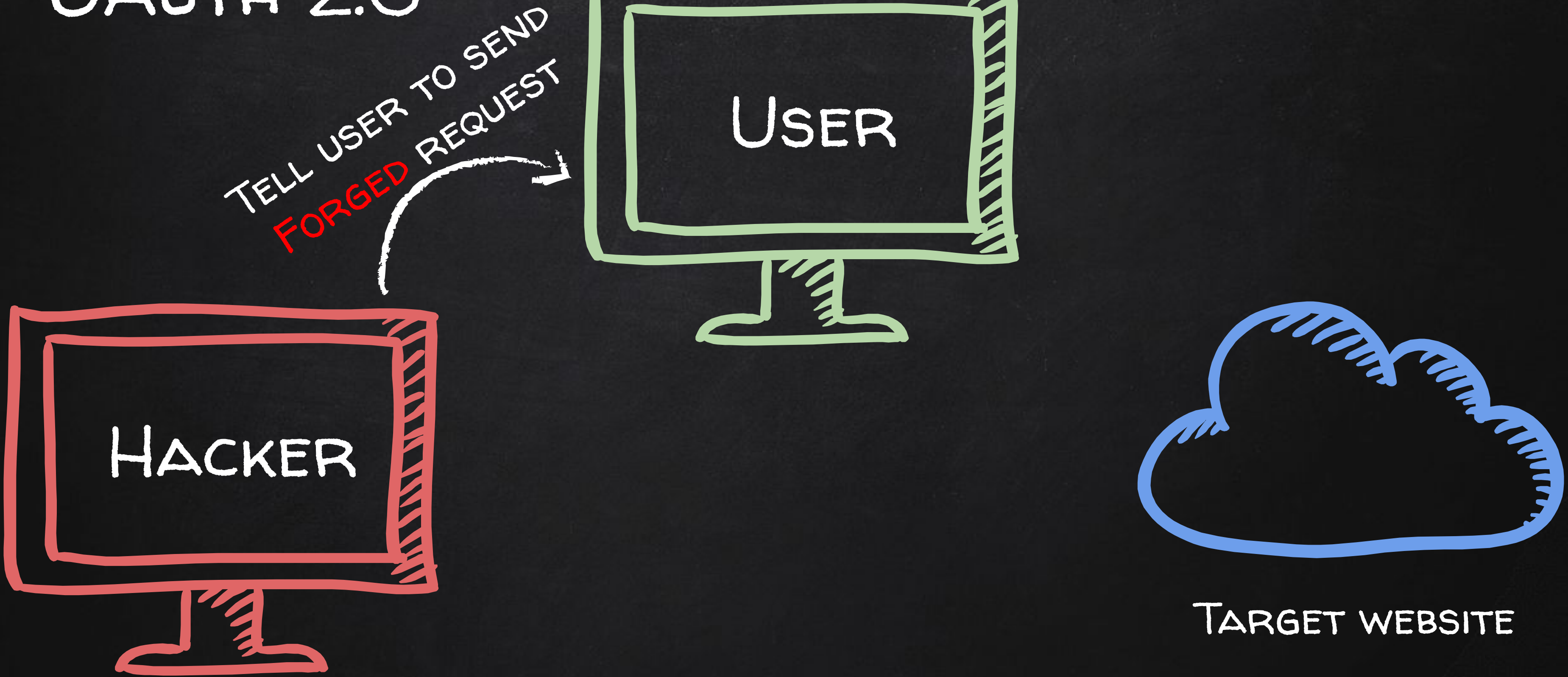
OAuth 2.0

User

Hacker

Target website

OAuth 2.0



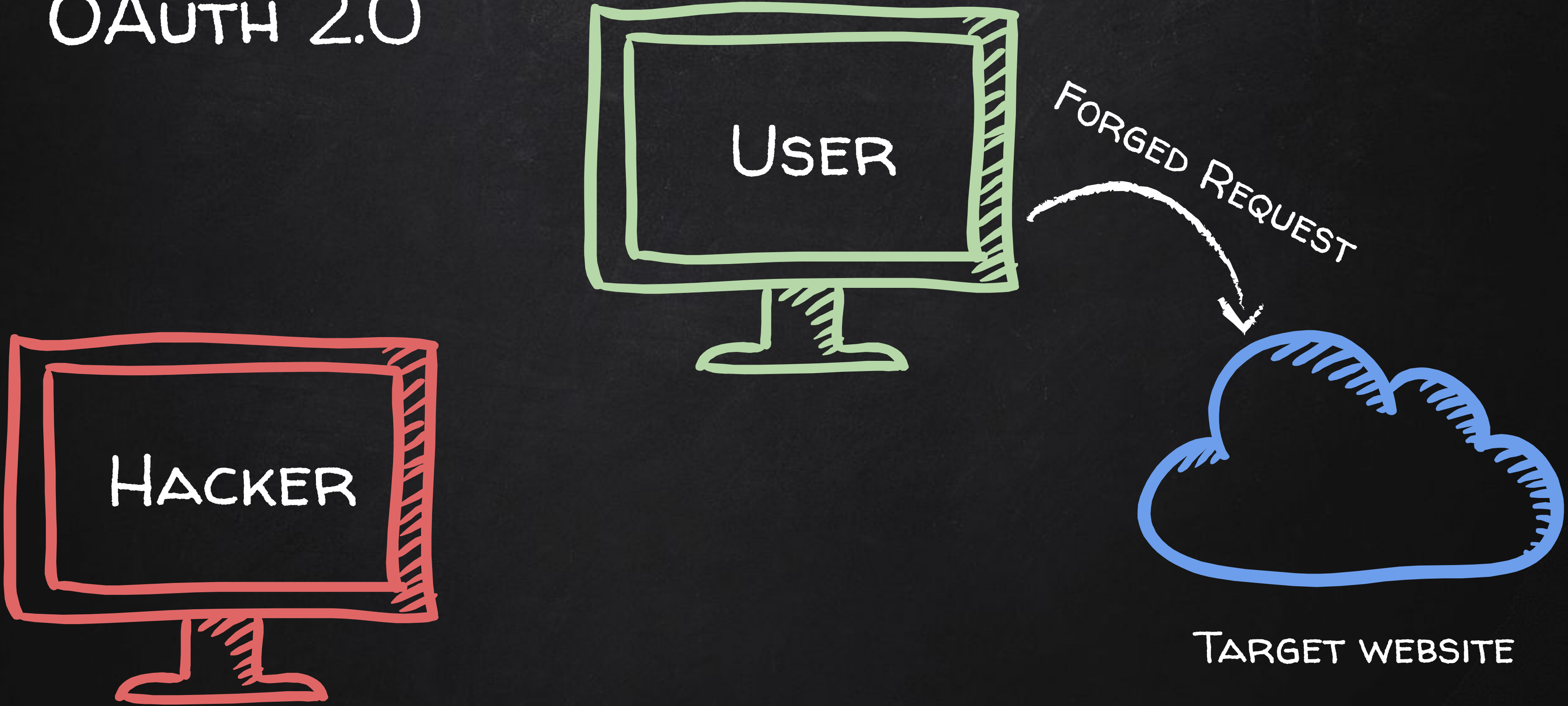
Tell user to send forged request

User

Hacker

Target website

OAuth 2.0



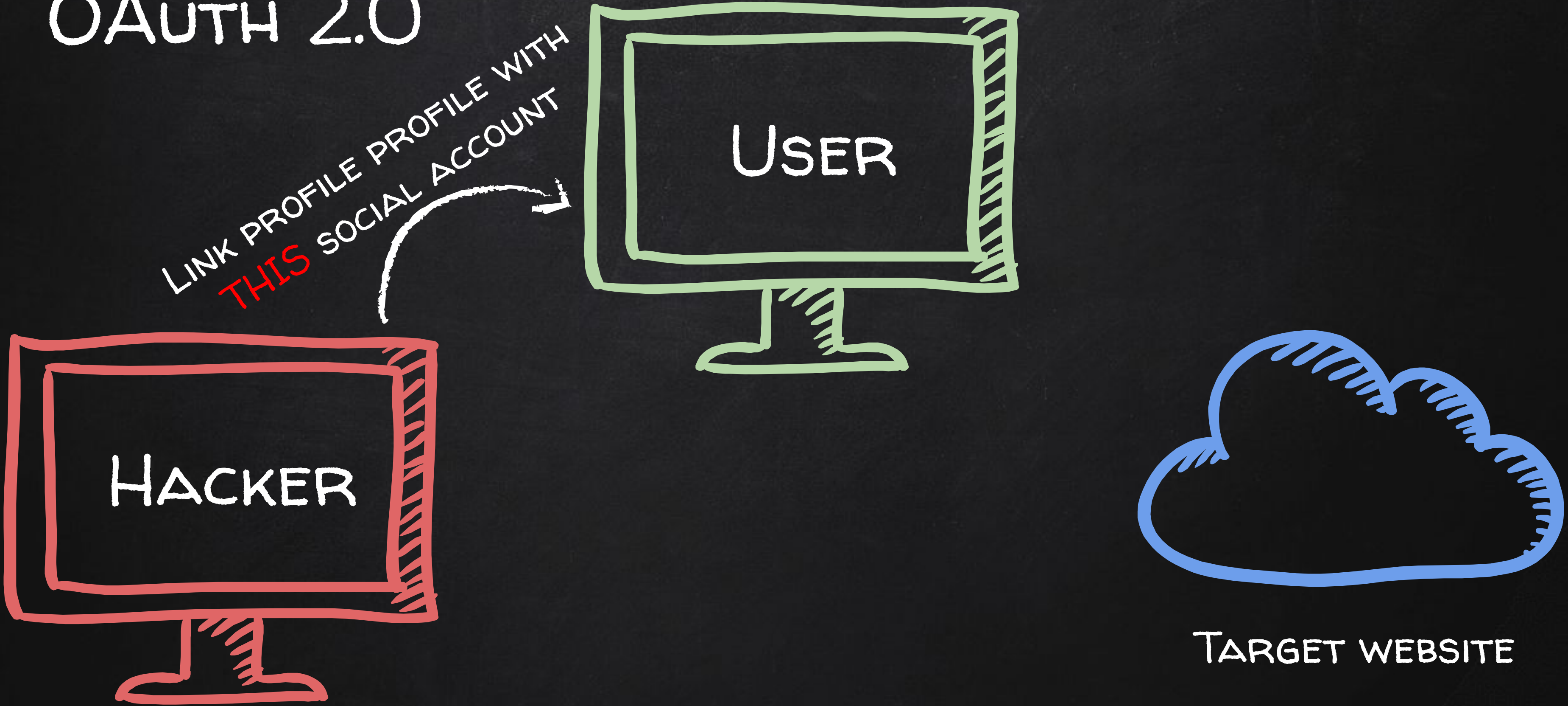
Hacker

User

Forged Request

Target website

OAuth 2.0



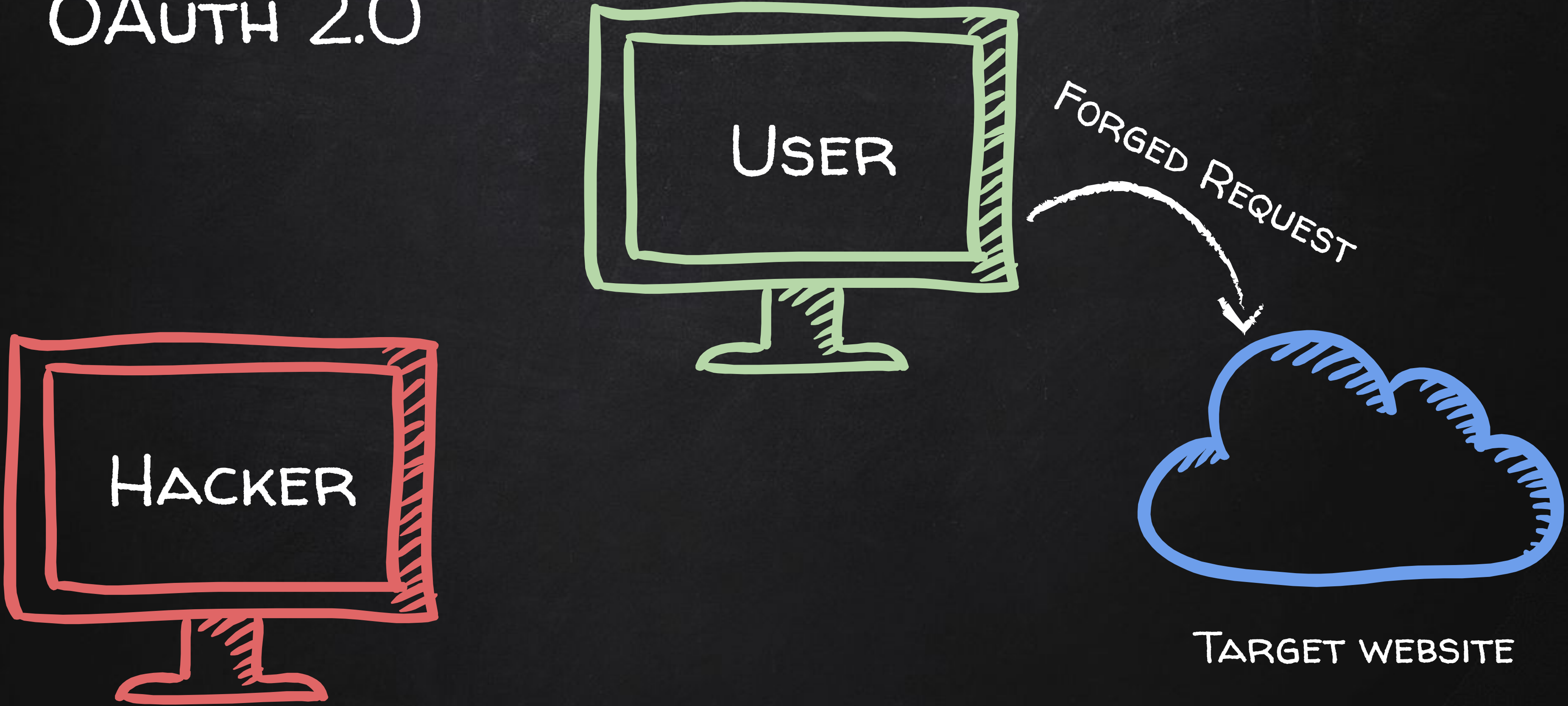
Link profile profile with THIS social account

User

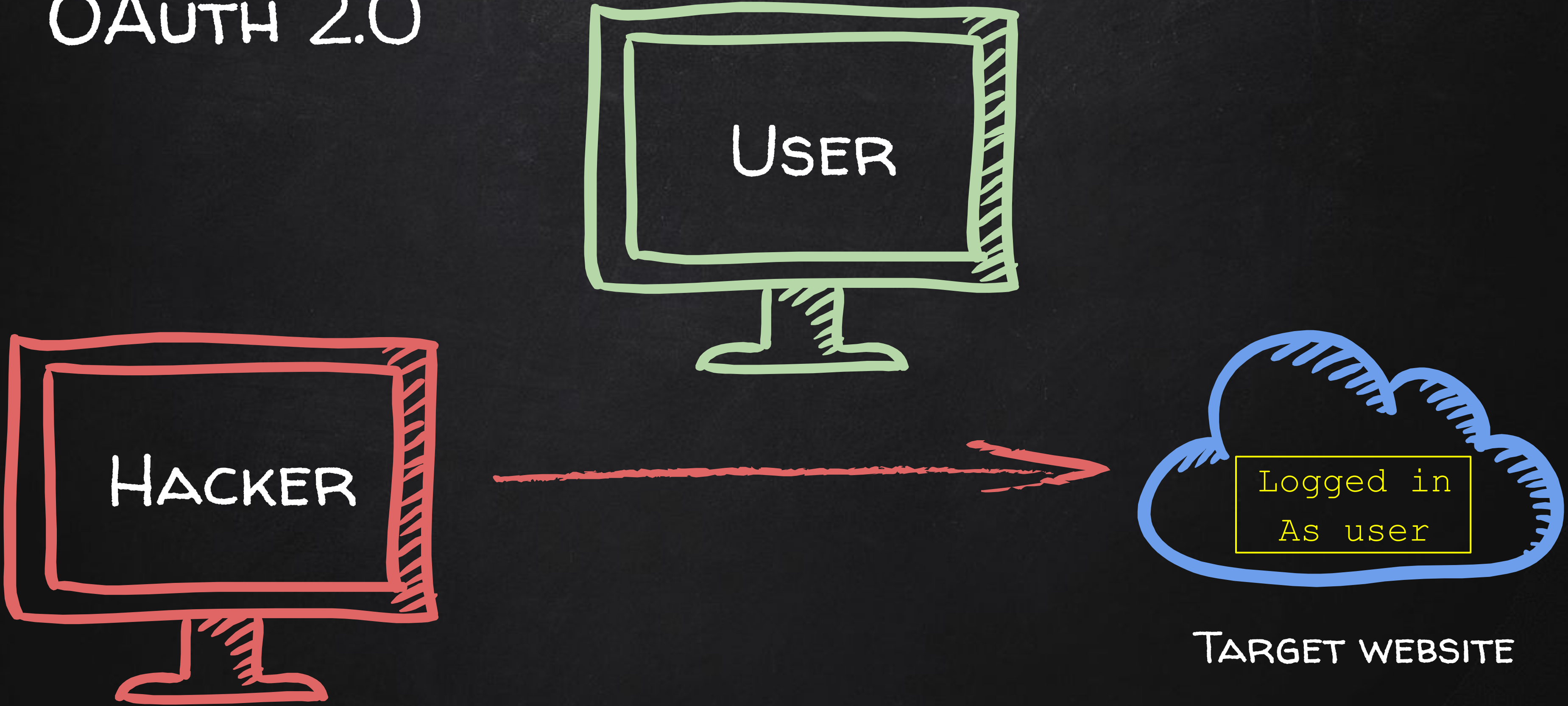
Hacker

Target website

OAuth 2.0



OAuth 2.0



User

Hacker

Logged in
As user

Target website