PHP For Hackers 2

GET and POST Request

GET & POST Request

GET & POST

GET Request

- بيستخدم لطلب البيانات من السيرفر. يعني لو عايز تشوف صفحة ويب أو تجيب بيانات معينة زي البحث عن حاجة على جوجل.
- البيانات اللي بتطلبها بتتبعت في الـ URL. يعني لو بتدور على حاجة، البيانات بتظهر في شريط العنوان.
- بيكون ليه حد أقصى لحجم البيانات اللي ممكن تبعتهم في الطلب (بسبب القيود على طول الـ URL).

مثال:

```
GET /search?q=programming
```

POST Request

- بيستخدم لإرسال البيانات للسيرفر. زي لما تسجل دخول أو تبعت بيانات في فورم.
- البيانات هنا بتتبعت في جسم (body) الطلب، مش في الـ URL. يعني بتكون مخفية في شريط العنوان.
- الـ POST بيستخدم في العمليات اللي ممكن تغير حاجة في السيرفر زي إضافة بيانات جديدة.
- مفيش حد أقصى لحجم البيانات اللي ممكن تبعتهم في POST زي GET.

مثال:

```
POST /login
```

الطلب بيكون مثلاً:

```json
{
  "username": "user",
  "password": "12345"
}
```

---

1. GET: البحث في جوجل

لما تدخل على جوجل وتكتب كلمة بحث، زي "programming"، وتضغط Enter، بيتم إرسال طلب GET للسيرفر علشان يجيب لك النتائج.

الـ URL اللي بيظهر فوق هيكون حاجة زي:

```
<http://www.google.com/search?q=programming>
```

هنا الـ GET طلب البيانات اللي هي نتائج البحث عن الكلمة "programming"، والكلمة نفسها ظهرت في الـ URL.

---

2. POST: تسجيل الدخول على فيسبوك

لما تفتح صفحة تسجيل الدخول على فيسبوك وتدخل الإيميل والباسورد وتضغط على "Log In"، السيرفر هنا هيستلم الطلب كـ POST.

الـ URL هيكون مثلاً:

```
<http://www.facebook.com/login>
```

لكن البيانات الخاصة بتسجيل الدخول (الإيميل والباسورد) مش هتظهر في الـ URL؛ هي هتتبعت في جسم الطلب بشكل مخفي.

جسم الطلب هيكون حاجة زي:

```json
{
  "email": "user@example.com",
  "password": "mypassword123"
}
```

هنا الـ POST بيستخدم لأنك بتبعت بيانات حساسة (الإيميل والباسورد) والعملية دي بتحتاج إنها تتعامل بسرية، وكمان ممكن تغير حاجة في السيرفر (زي تسجيل دخول المستخدم).

Cyber Security and Networks Usage

---

3. في الشبكات:

- الGET و POST بيستخدموا HTTP/HTTPS: البروتوكولات دي هي اللي بتحدد إزاي الأجهزة تتواصل مع بعضها على الشبكة. لما تعمل طلب GET أو POST، ده معناه إن في اتصال بيحصل بين جهازك (الـ client) والسيرفر في مكان تاني على الشبكة (ممكن يكون الإنترنت أو شبكة محلية).
- نقل البيانات: في الشبكات، كل طلب GET أو POST بيبعت بيانات على شكل حزم (packets) بتتنقل من جهازك إلى السيرفر والعكس. ففهم GET و POST بيساعدك تفهم إزاي البيانات بتتنقل في الشبكة وبتتواصل بين الأجهزة.

---

4. في الأمن السيبراني:

- عرضة للهجمات POST و GET:
  - في طلبات GET، البيانات بتظهر في الـ URL، وده ممكن يخليها هدف سهل لو حد بيستخدم الشبكة بتاعتك (مثلاً في Wi-Fi مش آمن أو على شبكة عامة). لو البيانات حساسة (زي كلمات المرور)، ممكن حد يشوفها أو يسرقها لو الاتصال

مش مشفر.

- الPOST أكتر أمان في نقل البيانات لأنها مش بتظهر في الـ URL، لكن مش معناه إنها مش معرضة للخطر. لو الاتصال نفسه غير مؤمن (مش بيستخدم HTTPS)، البيانات اللي بتتبعت ممكن تتعرض للاعتراض أو التعديل.
- أهمية الـ HTTPS:
  - هنا بتيجي أهمية استخدام HTTPS بدل HTTP. لو الطلبات (سواء GET أو POST) بتتم على HTTP العادي، البيانات بتكون غير مشفرة، وده يعرضها لهجوم زي Man-in-the-Middle (MITM) اللي بيقدر يعترض البيانات أثناء نقلها.
  - باستخدام HTTPS، كل الطلبات (سواء GET أو POST) بتبقى مشفرة، وده بيحمي البيانات من الاعتراض أو التعديل، سواء كانت ظاهرة في الـ URL أو في جسم الطلب.
- هجمات CSRF و XSS:
  - في طلبات POST، ممكن تتعرض لهجمات زي Cross-Site Request Forgery (CSRF) اللي فيها المخترق بيجبر المستخدم على إرسال طلبات غير شرعية للسيرفر من غير ما المستخدم يكون واعي.
  - في حالة GET، ممكن يحصل هجوم زي Cross-Site Scripting (XSS) لو المتصفح نفذ كود JavaScript ضار من خلال الـ URL، لأن بيانات GET بتكون جزء من الـ URL.

من الاخر:

- الGET و POST بيستخدموا في نقل البيانات بين العميل والسيرفر عبر الشبكة.
- الPOST أكتر أمان في نقل البيانات الحساسة زي كلمات المرور.
- استخدام HTTPS مهم لتأمين الطلبات من الهجمات والاعتراض.
- في الأمن السيبراني، لازم دايمًا تتأكد من حماية الشبكة والطلبات (GET و POST) لتجنب هجمات زي MITM و CSRF و XSS.

Input Fields in HTML

- اتمني منك تكون قرأت الجزء اللي فوق ده وفهمته هو طويل شويه بس رايق
- المهم احنا هنتعلم نعمل ال fields بتاعت الادخال بال html
- مش بتشوف ف اي موقع مثلا يديك مربع كدا يقولك دخل الايميل والباس؟ اهو احنا هنعمل كدا تعالي اوريك ازاي

```
<html>
    <head>
        <title>Mohammed Tantawy</title>
    </head>
    <body>
```

```
        <form action="welcome.php" method="POST">
            <input name="username">
            <input name="password">
            <input name="Ok" type=submit>
        </form>
        <?php
        ?>
    </body>
</html>
```

- اولا لازم tag ال input احطو في تاج form
- اول حاجه `action = "welcome.php"f` انا هنا بقولو يا باشا البيانات اللي هتتكتب فاليوزر والباس ابعتها ف فايل ال welcome وال `method="POST"f` بقولو ابعتلي البيانات بطريقه ال post وطبعا احنا عرفنا الفرق بينهم فوق
- بعد كدا `input name="username"f>f` الinput من اسمها معروفه اعملي حقل ادخال وname ده مش اسمه لأ دا الاسم اللي هحطو عشان اعرف اتعامل معاه فال php فالbackend واعمل عليه العمليات بتاعتي وهفهمك دلوقتي
- ال `<input name="Ok" type=submit>` نفس الكلام برضو هنا بس قولتلو type=submit معناه اني ده هيكون زرار اللي هو submit زي login وكدا يعني بيبعت البيانات

- بص هتبقي بالشكل ده ودخلت انا test اكني ده يوزر وباس يعني
- طبعا اللي انت فهمتو دلوقتي اني لما ادخل اليوزر والباس البيانات هتروح فين؟ صح كدا انت مصححص فالwelcome.php تعالي بقي اوريك ايه اللي هنكتبو ف فايل ال welcome ده

```
<?php
$x = $_POST['username'];
$y = $_POST['password'];

echo "<h1>$x</h1>";
echo "<h1>$y</h1>";
?>
```

- دا فايل ال welcome انا هنا قولتلو ايه
- اول حاجه `;x = $_POST['username']$` بقولو يا باشا هاتلي القيمه اللي جوا الusername واللي انا باعتها بطريقة الpost فالمتغير x
- خلي بالك الusername ده اللي هو قيمة ال name اللي حاطيناها فال input ومش لازم يكون اسمه username يعني حط اي اسم يعجبك بس تكون حاجه معبره يعني عشان متتلغبطش
- نفس الكلام عملتو مع ال password
- بعد كدا `;"< echo "<h1>$x</h1>"; echo "<h1>$y</h1` بقولو اعرضهملي

- طبعا مش من الطبيعي انه يتعرضو بس انا بوضحلك بس

test

test

- بص عرضلي الtest وال test اللي انا دخلتهم
- كل ده عملناه بطريقة الpost ولو تلاحظ لما دوسنا enter اللينك جابلنا ايه

`http://localhost/welcome.php`

- يعني بيقولي انت دلوقتي فالفايل welcome وده صح فعلا احنا قولناله البيانات اللي هتجيلك خزنها ف ال welcome واعرضها
- تعالي تحت اوريك لو بال GET هيبقي اللينك عامل ازاي

`http://localhost/?username=test&password=test&Ok=Submit+Query`

- تخيل معايا انت لو بتسجل دخول وبعد ما سجلت دخول جابلك اللينك قبل ماتعمل حاجه روح بلغها للموقع عشان دي ثغره
- تاني حاجه هتيجي تسألني ايه الخطوره يعني ف كدا انا قاعد فالبيت ومحدش شايف اللينك غيري
- اقولك لأ دا انت اللينك بتاعك شافه ملايين عشان في موقع اسمه Net Archive بيسجل اي لينك بيتطلب عنده فا تخيل حد فاتح الموقع لقي اللينك ده محطوط هيروح يا باشا واخد اليوزر والباس ويدخل علي اكونتك وشكر الله سعيك
- والكود عشان تعمله بالGET وعاوز تجرب يعني نفس الكود اللي فوق شيل بس كلمة POST وحط GET

`GET['the_name_here']_$`

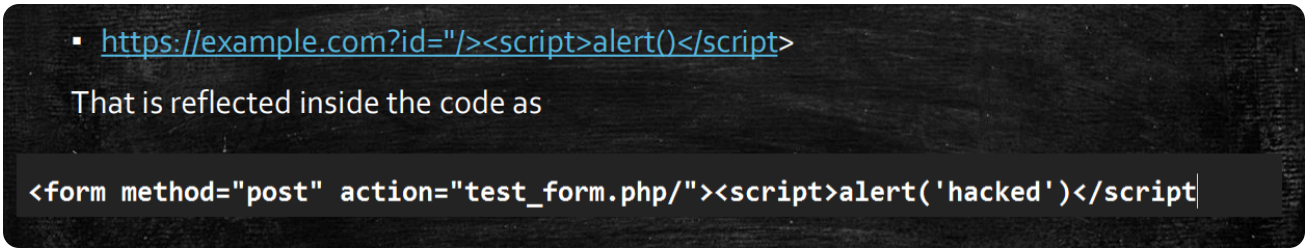
---

htmlspecialchars Method

- لو انت مش عارف الrequest اللي جاي ده GET ولا POST حط SERVER وريح دماغك وهو هيتعامل

`x = $_SERVER['the_name_here']$`

- طيب دلوقتي في ثغرة كدا اسمها XSS دي ببساطه كدا لما باجي ادخل اي كود script يعني في اي input field بيتنفذ معايا والمفروض ميتنفذش فا دي تعتبر ثغره اسمها XSS



- تأثيرها اني ممكن الهاكر يحط كود ويسمع فاللينك بتاع الموقع فا تبعت اللينك ده للvictim وتسرق منه ال Cookies بيكون فيها يعني الايميل والباس بتوعك وحاجات تانيه وكدا
- فا هنحل المشكله دي او الثغره باستخدام ميثود ال htmlspecialchars وهيبقي الكود بتاعي كدا

`<? ;php echo htmlspecialchars($_SERVER["your_name"])?>`