This is an HTML form that allows a user to upload a file to a web server using the HTTP POST method. The "action" attribute specifies the URL of the server-side script that will handle the file upload, in this case, it is set to "<http://192.168.100.171/app/wp-content/plugins/reflex-gallery/admin/scripts/FileUploader/php.php>". The "enctype" attribute specifies the content type of the data that will be submitted, in this case, it is set to "multipart/form-data" to allow the file to be included in the request. The input type "file" creates a file selector dialog box, allowing the user to select a file from their computer to upload. The "Submit" button triggers the submission of the form to the specified URL.

A PHP reverse shell is a script that allows an attacker to establish a shell on a compromised system and control it remotely. It works by uploading a PHP script to a vulnerable web server and executing it. Once the script is executed, it connects back to the attacker's machine and opens a shell session. This allows the attacker to execute commands on the compromised system and exfiltrate data.

A PHP reverse shell is often used in post-exploitation scenarios where the attacker has already gained access to a web server and wants to maintain persistent control. It can also be used to pivot to other systems on the same network, as the attacker can use the compromised web server as a jump box to launch further attacks.

This HTML code can be sent to a vulnerable server through various means. One possible way is through a cross-site scripting (XSS) vulnerability in a web application hosted on the server. In an XSS attack, the attacker injects malicious code (in this case, the HTML code with the file upload form) into a vulnerable web page or application. When a victim user visits the web page or application, the injected code is executed in the victim's browser, and the form is sent to the server specified in the "action" parameter of the "form" tag.

Another possible way is through a file upload vulnerability in a web application hosted on the server. In this case, the attacker may upload a file containing the HTML code to the vulnerable application, and then visit the uploaded file to trigger the execution of the code. The file upload vulnerability can be exploited if the web application does not properly validate the file type, size, or content, and allows the uploaded files to be accessed or executed by other users.