Diario di lavoro

Luogo	Canobbio
Data	27.09.2019

Lavori svolti

13h15 – 13h40

Ho creato la guida per la vulnerabilità Cross-site Scripting (XSS).

Cross Site Scripting (XSS)

di Administrator

Pubblicato il 27.09.2019 alle 13:33

Gli attacchi Cross-Site Scripting (XSS) sono un tipo di iniezione, in cui gli script dannosi vengono iniettati in siti web altrimenti benigni e affidabili. Gli attacchi XSS si verificano quando un attaccante utilizza un'applicazione web per inviare codice dannoso, generalmente sotto forma di script lato browser, a un altro utente finale. I difetti che permettono il successo di questi attacchi sono abbastanza diffusi e si verificano ovunque un'applicazione web utilizza l'input di un utente all'interno dell'output che genera senza convalidarlo o codificarlo.

Un attaccante può usare XSS per inviare uno script dannoso ad un utente ignaro. Il browser dell'utente finale non ha modo di sapere che lo script non deve essere considerato attendibile e lo eseguirà. Poiché pensa che lo script provenga da una fonte attendibile, lo script dannoso può accedere a qualsiasi cookie, token di sessione o altre informazioni sensibili conservate dal browser e utilizzate con quel sito. Questi script possono anche riscrivere il contenuto della pagina HTML.

All'interno di HackerLab è presente una vulnerabilità di tipo XSS, ed è proprio in questa pagina che può accadere.
Nonostante vi siano dei controlli nella validità del testo contenuto in un articolo è possibile comunque eseguire un attacco di tipo xss. Di seguito segue un esempio:

```
<a onclick="alert('XSS :D');">Cliccami</a>
```

Prova a cliccare il seguente testo:Cliccami
Potrai notare l'esecuzione dello script javascript.
Questa vulnerabilità è molto pericolosa e potente.
Fonti

Figura 1 Articolo XSS

13h40 – 14h05

Ho implementato una vulnerabilità di tipo Failure to Restrict URL Access e creato un articolo all'interno del blog che spiega come sfruttare la vulnerabilità.

Failure to Restrict URL Access

di Administrator

Pubblicato il 27.09.2019 alle 13:56

Se l'applicazione non riesce a limitare adeguatamente l'accesso agli URL, la sicurezza può essere compromessa da una tecnica chiamata navigazione forzata. La navigazione forzata può essere un problema molto serio se un aggressore cerca di raccogliere dati sensibili attraverso un browser Web richiedendo pagine specifiche o file di dati. Questo significa che l'applicativo è affetto da una vulnerabilità di tipo Failure to restrict URL Access.

HackerLab a sua volta è vulnerabilie a questo tipo di attacco. Solitamente nel file robots.txt vengono salvate delle regole riguardanti i percorsi del sito web specificando ai bot che indicizzano siti web cosa fare.
Richiedendo il file robots.txt al percorso `/robots.txt` è possibile vedere il seguente contenuto:

```
# Regola da applicare a tutti i robot
User-agent: *
# Non fare accedere alle pagine di amministrazione
Disallow: info.php
Disallow: /admin/
```

Possiamo notare due regole che ci possono interessare, ovvero il fatto di bloccare l'indicizzazione della cartella `admin` e del file `info.php`.
Non ci resta che provare ad accedere a queste cartelle direttamente dal browser.
Accedendo alla pagina `/info.php` possiamo notare come vengano caricate e mostrate tutte le informazioni riguardanti PHP, questo è molto pericoloso in quanto un attaccante può ricercare vulnerabilità in base alle versioni installate, inoltre è possibile vedere altre informazioni come per esempio il percorso del progetto, ...
Questa vulnerabilità sfruttata con altre vulnerabilità può essere molto pericolosa.
Fonti

Figura 2 Articolo Failure to Restrict URL Access

14h05 – 14h20
Implementata una vulnerabilità di tipo Security Misconfiguration, inoltre ho creato anche il relativo articolo all'interno del sito web che la documenta.

Security Misconfiguration

di Administrator

Pubblicato il 27.09.2019 alle 14:12

Una vulnerabilità di tipo Security Misconfiguration è quando un applicativo è messo in produzione con impostazioni di configurazion errate. Esempi possono essere: password di default, messaggi di debug, ...
È una vulnerabilità molto comune all'interno di siti web.
Una vulnerabilità di questo tipo è presente all'interno di HackerLab.

Ti basterà andare nella sezione dei commenti e aprire il profilo di un utente "eliminato", noterai un messaggio di errore proveniente dal Framework utilizzato per lo sviluppo di questo sito web.
In questo modo l'attaccante avrà informazioni in più sul sito web e possibili vulnerabilità da sfruttare.

Fonti

Figura 3 Articolo Security Misconfiguration

14h20 – 14h45

Ho creato un utente dedicato al sito web per prevenire che attraverso vulnerabilità si possano toccare anche altri database esterni.

15h00 – 16h20

Ho iniziato a pensare a come implementare la vulnerabilità SQL Injection in un modo che non sia distruttiva. Al momento è stata implementata all'interno della funzionalità di ricerca di HackerLab.
I problemi che può causare al momento sono:

- Update, Insert, Delete, Select e Drop del database hackerlab
- Possibilità di eliminare file

Il problema che devo risolvere è la possibilità di eliminare file. Al momento se un utente modifica tramite SQL Injection l'immagine di un articolo può eliminare qualsiasi file a sua scelta, esempio:

```
a%'; UPDATE articles SET image = "../composer.json"; --
```

Questo perché la funzione di delete è implementata nel seguente modo:

```
unlink(__DIR__.'/../../storage/'.$article["image"]);
```

Quindi, ho intenzione di rendere sicura la rimozione del file in modo di permettere solamente azioni al database.

16h20 – 16h30

Stesura diario.

Problemi riscontrati e soluzioni adottate
Non ho riscontrato nessun problema.

Punto della situazione rispetto alla pianificazione
Mi trovo molto avanti rispetto alla pianficia.

Programma di massima per la prossima giornata di lavoro
Rendere sicura eliminazione di file, implementare la funzionalità di reset dei dati e del database.