Web Security 2
Iniezioni di codice

Riccardo BONAFEDE

Università degli Studi di Padova

https://cybersecnatlab.it

License & Disclaimer

License Information

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Obiettivi

- Comprendere il concetto di iniezione
- Illustrare le tipologie di iniezioni più diffuse

Prerequisiti

- Modulo WS_2.1 - Backend e Database

Argomenti

- Iniezioni
- SQL Injections
- Command Injections

Argomenti

- Iniezioni
- SQL Injections
- Command Injections

Iniezioni

- Le iniezioni sono una macro-categoria di vulnerabilità
- Esistono perché i back-end hanno la necessità di interagire con altri software
- In queste interazioni spesso vengono utilizzati dati inseriti da utenti

Iniezioni

- Per capire il concetto, si pensi ad una applicazione che utilizza un file CSV per salvare le utenze

```
username,   privilegio, password
admin,      1,          12345
mario,      3,          12345
giovanni,   3,          12345
```

Iniezioni

- I vari utenti sono separati da un carattere \n e i vari campi degli utenti da una virgola
- L'applicazione è incaricata di gestire la corretta formattazione di ogni dato
- Ad esempio, per un utente che si registra, l'applicazione crea una nuova riga con i suoi dati

Iniezioni

- I vari utenti sono separati da un carattere \n e i vari campi degli utenti da una virgola
- L'applicazi corretta formattazi
- Ad esempi tra, l'applicazione crea una nuova riga con i suoi dati

```
username,   privilegio, password
admin,      1,          12345
mario,      3,          12345
giovanni,   3,          12345
pippo,      3,          potamo
```

Iniezioni

- Cosa succede se un utente si registra inserendo una virgola (,) all'interno del nome?
- Se l'applicazione non filtra la virgola, allora il file CSV risulterà corrotto

Iniezioni


```
username,   privilegio, password
admin,      1,          12345
mario,      3,          12345
giovanni,   3,          12345
pippo,,     3,          potamo
```

- La formattazione è errata
- Qual è la password?
- Qual è il privilegio?

Iniezioni

- Un attaccante potrebbe anche sfruttare la cosa a suo favore, ad esempio inserendo come username:

```
pippo,1,potamo \n luigi
```

- Il CSV diventerebbe

```
username,   privilegio, password
admin,      1,          12345
mario,      3,          12345
giovanni,   3,          12345
pippo,1,potamo
luigi,      3,          password
```

Iniezioni

- L'attaccante in questo caso riesce a registrare due utenti diversi, di cui uno con privilegi elevati
- L'applicazione dovrebbe controllare che i dati inseriti dagli utenti non contengano caratteri pericolosi

Argomenti

- Iniezioni
- SQL Injections
- Command Injections

SQL Injections

- Queste vulnerabilità si presentano spesso con i database
- Prendiamo ad esempio il seguente codice

```
$userQuery = mysqli_query("SELECT * FROM users
    WHERE email = '"    . $_POST['email']       . "'
    AND password = '"   . $_POST['password']    . "'"
);
```

SQL Injections


```php
$userQuery = mysqli_query("SELECT * FROM users
    WHERE email = '"    . $_POST['email']       . "'
    AND password = '"   . $_POST['password']    . "'"
);
```

```sql
SELECT * FROM users
WHERE email = 'admin@gmail.com'
AND password = '123456'
```

- Query per un'operazione di log in all'interno di un'applicazione
- Recupera le informazioni dal database in base ai dati inseriti da un utente
- Le stringhe sono delimitate da singoli apici

SQL Injections

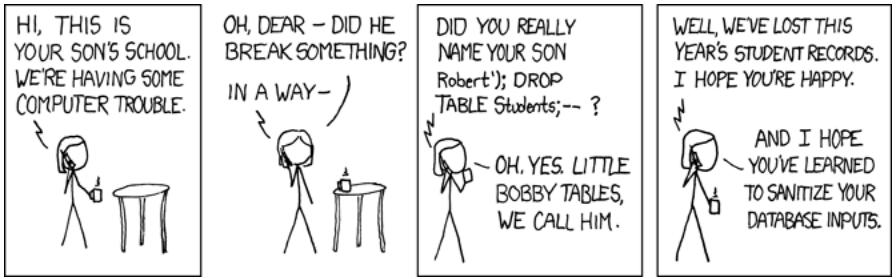
- Cosa succede se la mail contiene un apice (') ?

```
SELECT * FROM users
WHERE email = ''admin@gmail.com'
AND password = '123456'
```

- Il database non è in grado di riconoscere quali degli apici presenti sono input di un utente e quali parte del codice

SQL Injections

- Potendo *scappare* dagli apici, è possibile iniettare del codice SQL
- Questo tipo di vulnerabilità è chiamato SQL Injection



SQL Injections

- In questo caso è possibile iniettare condizioni logiche:

```
SELECT * FROM users
WHERE email = 'admin@gmail.com' OR 1='1'
AND password = '123456'
```

- Se questa fosse la query responsabile per il login di un utente, questa SQL injection permetterebbe di autenticarsi senza usare nessuna password

SQL Injections

- Il modo migliore per trovare SQL Injection è inserire apici in tutti gli input processati dal back-end
- Se vi è una iniezione possibile in una query, questa molto probabilmente non sarà sintatticamente corretta
- Osservando la risposta, si può intuire se il database ha restituito un errore per la query non corretta oppure no:
  - *Una pagina bianca*
  - *Status code 500*
  - *Nei casi fortuiti il messaggio di errore del database*

Argomenti

- Iniezioni
- SQL Injections
- Command Injections

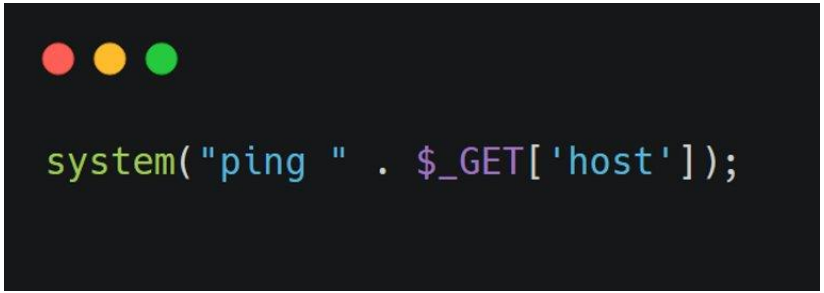
Command Injections

- Oltre ai database, i back-end hanno la necessità di interagire con altri programmi
- Generalmente per utilizzare funzioni che sarebbe altrimenti costoso in termini di tempo da re-implementare
  - Ad esempio la conversione di file in altri tipi vedi *ImageMagick*

Command Injections

- Un esempio è l'utility *ping*, la quale richiede dei permessi speciali per poter essere eseguita
- Apparecchi come router e modem generalmente hanno bisogno di poter usare le sue funzionalità
- Il modo migliore è quella di eseguirla come comando fornendogli come argomenti i dati inseriti dall'utente

Command Injections



- Esegue il programma *ping*, fornendo l'host passato dall'utente
- La funzione *system* esegue programmi utilizzando *bash*

Command Injections

- Come nel caso delle SQL injections, è possibile in questo caso iniettare codice bash
- Usando il carattere del punto e virgola (;) è possibile far eseguire al sistema remoto ulteriori comandi oltre a *ping*
- Generalmente questo genere di vulnerabilità viene chiamato *Command Injection*

Command Injections

- Inserendo www.google.com; cat /etc/passwd

```
www.google.com; cat /etc/passwd
```

- Il comando eseguito sarà

```
ping www.google.com; cat /etc/passwd
```

- Che permette di leggere il file */etc/passwd*

Command Injections

- Oltre a *system* vi sono molte altre funzioni che permettono attacchi di questo tipo:
  - exec
  - shell_exec
  - Popen
- Ogni linguaggio definisce le sue funzioni specifiche

Command Injections

- È anche importante notare che è possibile iniettare anche altre tipologie di codice oltre ai comandi di sistema
- Molti linguaggi di scripting supportano l'esecuzione dinamica di comandi attraverso funzioni quali l'eval

Command Injections

- Come nel caso delle SQL Injections, il modo migliore per trovare questo tipo di vulnerabilità è di inserire in input caratteri speciali e osservarne la risposta
- Ogni linguaggio ha i suoi caratteri speciali, ma in via generale si può provare ad esempio con i caratteri:
  - ' " & ; () $

Web Security 2
Iniezioni di codice

Riccardo BONAFEDE

Università degli Studi di Padova

https://cybersecnatlab.it