NETWORK SECURITY

한양대학교 소프트웨어융합대학 소프트웨어학부
이연준 교수

주요 사항

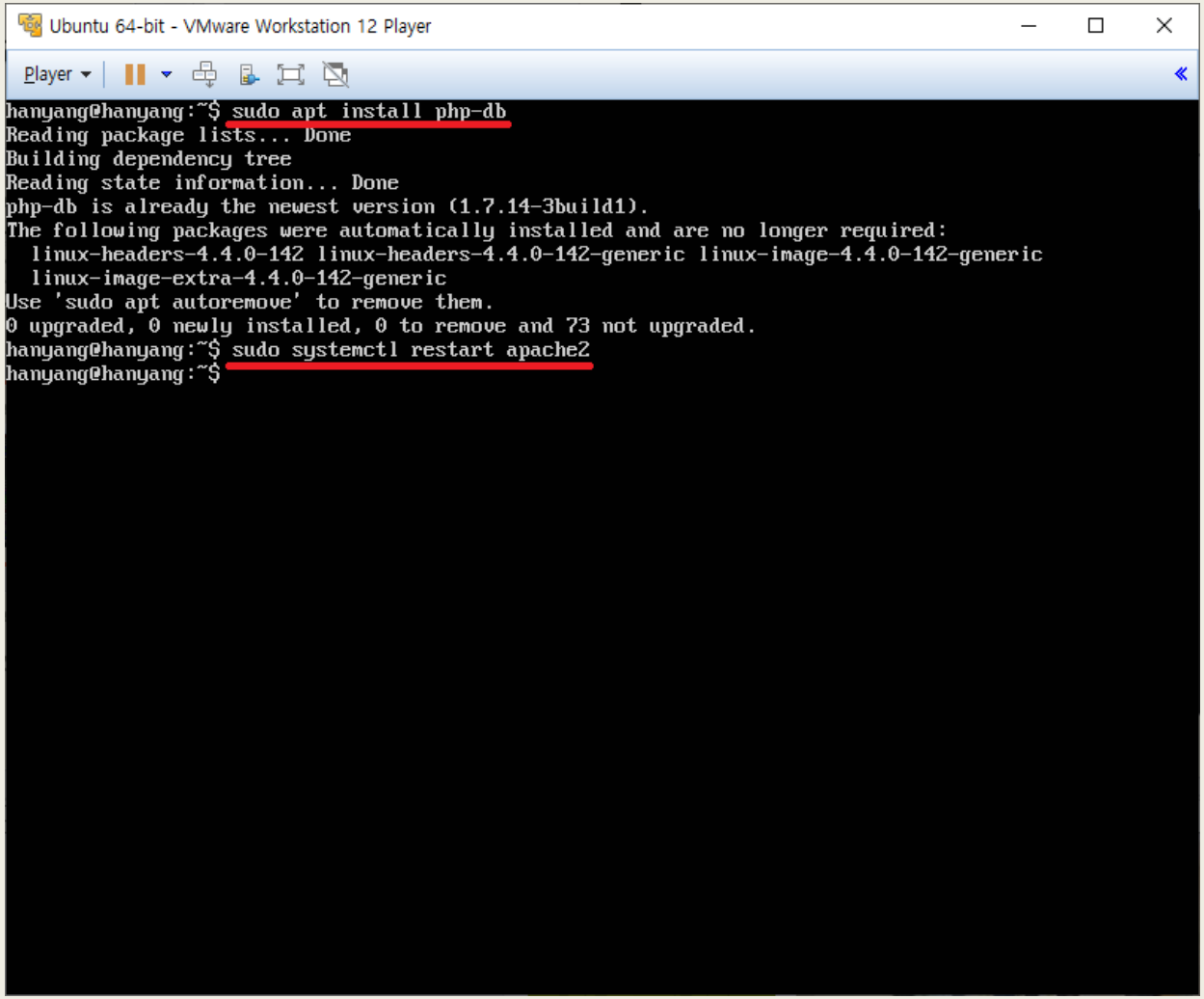
- **SQL Injection** 방어 기법에 대한 이해
- **Lab Preparation**
  - **PDO (PHP Data Objects) Database Driver** 설치
- **Lab Task**
  - **MySQL User** 권한 부여
  - **SELECT Query**에 대한 **Prepared Statement** 사용
  - **Database** 수정 **Query**에 대해 **Prepared Statement** 및 **Stored Procedure**를 사용
  - 입력 형식 지정을 통한 **Script Injection** 방지
- **Lab Question**
- **Evaluation**

LAB PREPARATION

실습 환경 구성 준비

- **PDO(PHP Data Objects)** 설치
- **MySQL**에서는 매개변수화된 **Query**나 **Prepared Statement**를 제공하지 않음
- 따라서 이를 사용하기 위해 **PDO MySQL** 드라이버를 설치함
- **sudo apt install php-db**
- 설치 후 해당 내용의 적용을 위해 **apache** 서비스를 재시작
- **sudo systemctl restart apache2**

실습 환경 구성 준비


Ubuntu 64-bit - VMware Workstation 12 Player
Player
hanyang@hanyang:~$ sudo apt install php-db
Reading package lists... Done
Building dependency tree
Reading state information... Done
php-db is already the newest version (1.7.14-3build1).
The following packages were automatically installed and are no longer required:
linux-headers-4.4.0-142 linux-headers-4.4.0-142-generic linux-image-4.4.0-142-generic
linux-image-extra-4.4.0-142-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 73 not upgraded.
hanyang@hanyang:~$ sudo systemctl restart apache2
hanyang@hanyang:~$

LAB TASK

MySQL 사용자 권한 제어

- **Localhost**에서만 연결되고 **LocationShare** 데이터베이스에서 **Select**문만 실행할 수 있는 **MySQL** 사용자를 추가.
- https://www.digitalocean.com/community/tutorials/how-to-create-a-new-user-and-grant-permissions-in-mysql
- https://dev.mysql.com/doc/refman/8.0/en/grant.html
- 사용자 계정명은 본인 학번으로 할 것
- **SHOW GRANTS FOR** '본인학번'@'localhost';

MySQL 사용자 권한 제어

- 생성 후 **db.php**에서 **//for lab 5 only** 주석 아래에 있는 부분을 방금 생성한 계정과 비밀번호를 입력
- **Stored Procedure** 사용을 위해 다음과 같이 권한을 부여할 것

```
mysql> show grants for '2019101059'@'localhost';
+-----------------------------------------------------------------------------------------+
| Grants for 2019101059@localhost                                                         |
+-----------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO '2019101059'@'localhost'                                          |
| GRANT SELECT, INSERT, UPDATE, EXECUTE ON `LocationShare`.* TO '2019101059'@'localhost' |
+-----------------------------------------------------------------------------------------+
2 rows in set (0.00 sec)

mysql>
```

- 권한 확인은 **SHOW GRANTS FOR '본인학번'@'localhost';**

Prepared Statement

- **PDO** 드라이버를 설치를 통해 **Prepared Statement** 사용 가능
- **functions.php**에 있는 기존의 내용들을 변경함
- **getReadPDO()**와 **getWritePDO()**는 데이터베이스 연결이 설정된 후 **PDO** 오브젝트를 반환함
- 기존의 코드 또는 **Prepared Statement**를 통해 **DB** 연동을 실패할 경우 **/var/log/apache2/error.log**에 오류 내용이 기록됨
- 아래의 4개 함수를 **Prepared Statement**를 이용하여 변경할 것
  - **usernameToId**
  - **idToUsername**
  - **getLocationUpdates**
  - **tryLogin**

Prepared Statement

- 참고 링크
- https://www.php.net/manual/en/book.pdo.php
- https://phpdelusions.net/pdo

Stored Procedure

- **SELECT** 문을 제외한 **UPDATE, INSERT, DELETE**문을 실행하는 모든 쿼리는 **Stored Procedure**를 생성할 것
- 마찬가지로 **function.php**를 수정해야하며 **Stored Procedure** 호출은 **Prepared Statement**를 통해서 호출
- 아래 **3**개의 함수를 수정할 것
  - **sendMessage**
  - **trySignup**
  - **setPendingFollower**

Stored Procedure

- **Hint**
  - 모두 쓰기 작업이므로 **getReadPDO()**가 아닌 **getWritePDO()**를 사용하여 **PDO** 오브젝트를 사용함
- 참고 링크
  - https://medium.com/@peter.lafferty/mysql-stored-procedures-101-6b4fe230967
  - https://dev.mysql.com/doc/connector-net/en/connector-net-tutorials-stored-procedures.html
  - https://dev.mysql.com/doc/refman/8.0/en/create-procedure.html

SQL Injection 재공격

- 이전 실습에서 진행했던 **SQL Injection**을 다시 시도해볼 것
- 이전과 동일하게 공격한 후 결과가 성공했는지 실패했는지 여부를 캡처할 것

LAB QUESTION

Lab Question

1. 사용자 입력단에서 **SQL**을 주석처리하는 '—'을 제거한다면 모든 **SQL Injection**을 방지할 수 있습니까? 이에 대한 답과 이유를 서술하세요.
2. 단순히 **Addslashes()** 또는 **mysql_real_escape_string()**을 사용하여 **SQL Injection** 방어를 하면 어떤 문제가 있습니까?
3. **DB**에 대한 보안적 접근을 유지하는데 있어서 **UNIX** 계열 **OS**의 접근 제어 메커니즘이 충분하지 않은 이유는 무엇입니까?
4. 기능적인 부분에 있어서 **addslashes()**와 **htmlentities()**의 차이는 무엇입니까?

Evaluation

- **Lab Task** 진행
  - **MySQL** 유저에게 권한 부여 확인
    - 권한을 확인하여 캡처
  - **Prepared Statement**를 통한 함수 수정
    - 수정한 함수 **4**개를 캡처 또는 복사할 것
  - **Stored Procedure**를 이용한 함수 수정
    - 수정한 함수 **3**개를 캡처 또는 복사할 것
  - **SQL Injection**을 다시 시도한 후의 결과 확인
- **Lab Question**
  - 주어진 문항에 대한 답과 해결 방안에 대해 간략하게 서술
- **Lab Task** 수행 결과를 위와 같이 명시한 대로 캡처하여 **MS Word** 또는 **PDF** 파일로 결과를 제출할 것.

Evaluation

- 과제 제출 기한 : 2019/10/14 23:59
- 과제 제출 시 메일 제목은 ‘본인 이름_학번’으로 제출
  - 예) 이석원_2019101059
- sevenshards00@gmail.com으로 보낼 것.

Q&A