



웹 취약점 점검 실습

시큐리티 아카데미 직무 1반 모의 해킹 팀

김명지 김하늘 노승찬 박선아 박수아 유건우

목차

최강모의해커즈

01

프로젝트 소개

1.1 주제 및 범위

1.2 팀원 소개

1.3 수행 일정

02

프로젝트 수행

2.1 프로젝트 요약

2.2 문제 출제 항목

2.3 개념 학습 내용

2.4 실습 환경 구축

2.5 실습 문제 구축

03

프로젝트 결과

3.1 문제 페이지 시연

3.2 최종 결과

3.3 향후 계획

1. 프로젝트 소개

1.1 주제 및 범위

1.2 팀원 소개

1.3 수행 일정

1.1 프로젝트 주제



웹 기본 개념 및 공격 기법 학습

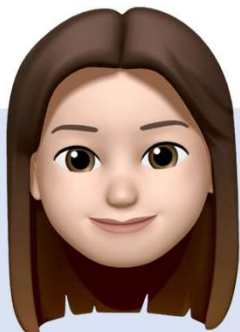
PHP 언어 사용해 웹 해킹 문제 사이트 구축

최종적으로, **웹 해킹에 대한 역량 향상** 목표

1.1 프로젝트 범위

- ✓ 웹 기본 개념 및 구조 학습
- ✓ 취약점 점검 기준(주요 통신기반시설 취약점 분석 평가 가이드)에 따라 보안 취약점, 대응방안 등 학습을 통한 이해
- ✓ 학습 페이지를 구축하고, 공격 기법을 활용할 수 있는 문제 풀이 사이트 제작

1.2 팀원 소개



김명지

팀장



김하늘

발표



노승찬

페이지 디자인



지한별 멘토님



박선아

발표자료 제작



박수아

개념 정리



유건우

실습 페이지 구축



1.3 수행 일정

	8월 3번째 주	8월 4번째 주	8월 5번째 주	9월 1번째 주	9월 2번째 주	9월 3번째 주	산출물
	08.12~08.15	08.16~08.24	08.25~08.30	08.31~09.04	09.05~09.08	09.09~09.20	
프로젝트 계획 및 웹 구조 학습과 이해							웹 개념 학습 보고서
웹 취약점 조사 및 웹 사이트 환경 구성							웹 사이트 코드
학습 페이지 개발							학습 페이지 코드
문제 페이지 개발							실습 문제 코드
웹 사이트 점검 및 마무리							
최종 결과 보고							발표 자료, 최종결과보고서

2. 프로젝트 수행

2.1 프로젝트 요약

2.2 문제 출제 항목

2.3 개념 학습 내용

2.4 실습 환경 구축

2.5 실습 문제 구축



2.1 프로젝트 요약

다음 취약점에 대해, 총 11개 문제 제작

문제 난이도	기준
1단계	개념 설명만으로 풀이 가능
2단계	간단한 필터링 우회 필요
3단계	복잡한 필터링 우회 필요

최강모의해커즈

취약점 종류	문제명	개념 학습	실습 페이지	문제 난이도
SQL Injection	Find User 1&2	○	○	★☆☆
	Login			★★★☆☆
XSS	XSS	○	○	★☆☆
	XSS_Stored 1&2			★★★☆☆
Command Inejction	Ping 1&2	○	○	★★★☆☆
	file			★★☆☆☆
Directory Injection	페이지 자체 취약점	○	○	★☆☆
Find Upload	Upload	○	○	★☆☆
	Upload-base64			★★★☆☆
File Download		○	X	



2.1 프로젝트 요약

팀별로 문제를 만든 후, 문제를 서로 풀어보며 학습 완료

Web Vulnerability Lab

Welcome to the Web Vulnerability Lab

Security_Academy 4기 모의해킹 미니 프로젝트입니다.

취약점에 대한 설명과 실습을 진행해보세요

Command Line Injection

SQL Injection (SQLi)

Cross Site Script (XSS)

File Upload

Directory Indexing

file Download

© 2024 4th Security Academy Mini_project

2.2 문제 항목 선정 - 선정기준

<OWASP Top 10: 2021>

A01	2021	Broken Access Control
A02	2021	Cryptographic Failures
A03	2021	Injection
A04	2021	Insecure Design
A05	2021	Security Misconfiguration
A06	2021	Vulnerable and Outdated Components
A07	2021	Identification and Authentication Failures
A08	2021	Software and Data Integrity Failures
A09	2021	Security Logging and Monitoring Failures
A10	2021	Server-Side Request Forgery (SSRF)

<2021년 주요정보통신 기반시설 취약점 분석 평가 기준 준용>

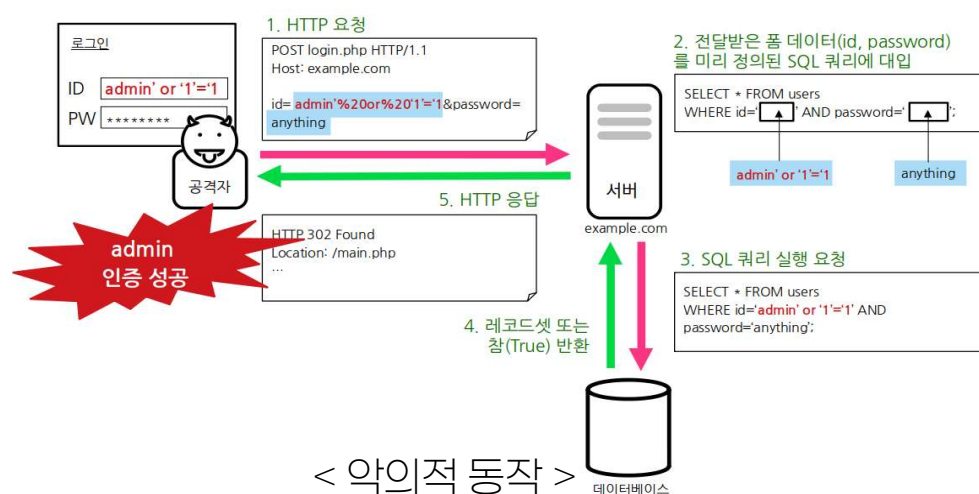
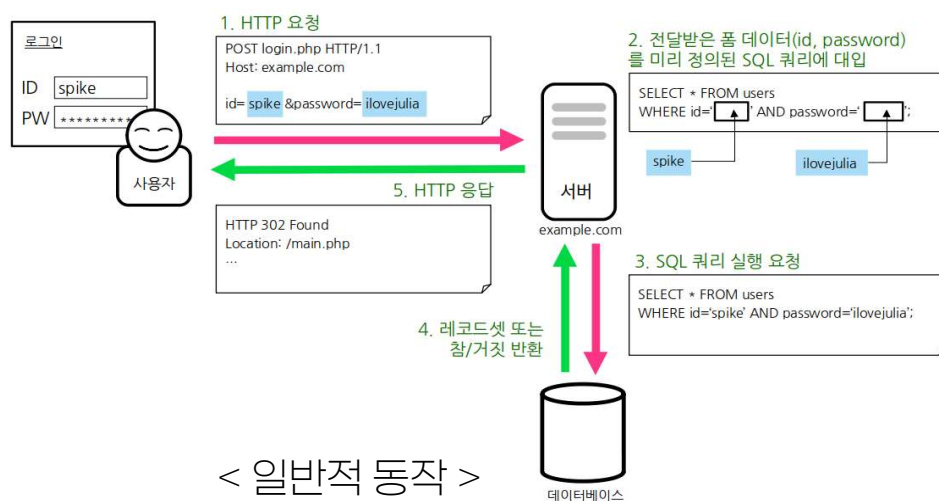
점검항목	항목 중요도	항목코드
버퍼 오버플로우	상	BO
포맷스트링	상	FS
LDAP 인젝션	상	LI
운영체제 명령 실행	상	OC
SQL 인젝션	상	SI
SSI 인젝션	상	SS
XPath 인젝션	상	XI
디렉터리 인덱싱	상	DI
정보 누출	상	IL
악성 콘텐츠	상	CS
크로스사이트 스크립팅	상	XS
악화 무자열 강도	상	BF
⋮		
프로세스 검증 누락	상	PV
파일 업로드	상	FU
파일 다운로드	상	FD
관리자 페이지 누출	상	AF

2.3 개념 학습 내용 – SQL Injection

정의

공격자가 웹 애플리케이션의 입력 필드를 통해 악의적인 SQL 구문을 삽입하여, 데이터베이스를 조작하거나 민감한 정보를 획득할 수 있는 취약점

공격 방식



대응 방안

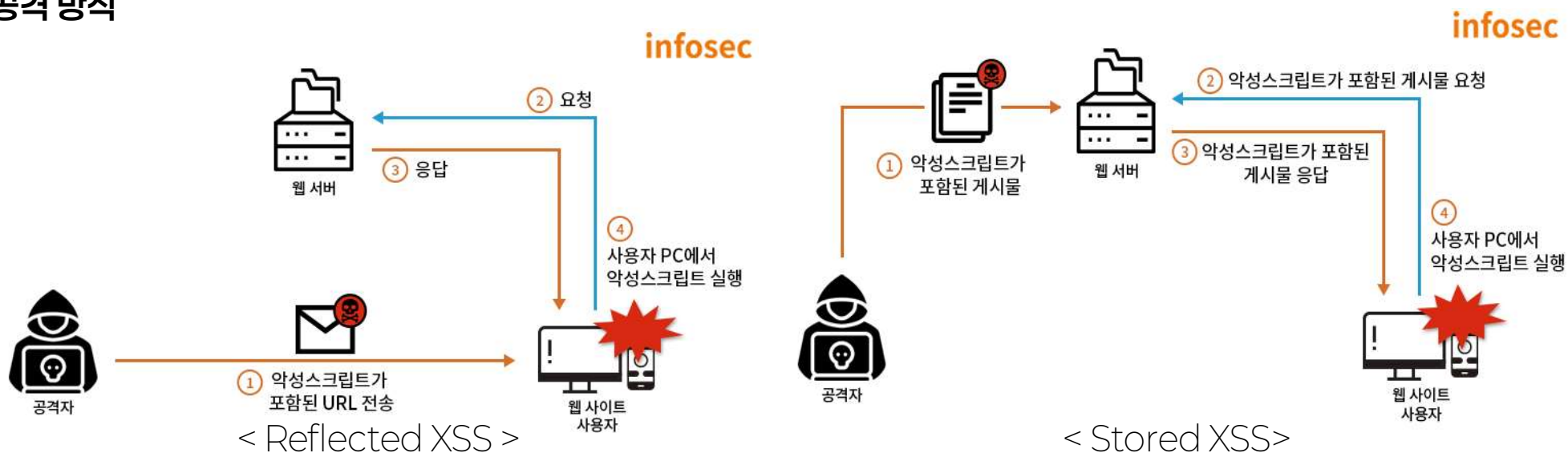
- 매개 변수화된 쿼리
- ORM 사용
- 입력 값 검증
- 최소 권한 원칙 적용

2.3 개념 학습 내용 – Cross-Site Scripting(XSS)

정의

공격자가 웹 페이지에 악성 스크립트를 삽입하여 사용자의 브라우저에서 실행되도록 해 세션 정보 탈취나 임의의 동작을 수행할 수 있는 취약점

공격 방식



대응 방안

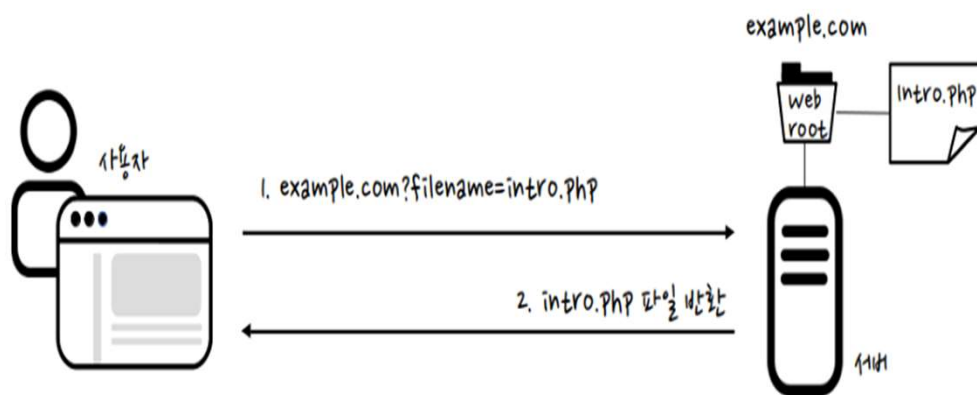
- 사용자 입력 값 검증
- 올바른 데이터 출력 값 처리
- 쿠키에 Http Only 플래그 설정
- CSP(content Security Policy) 적용

2.3 개념 학습 내용 – Directory Indexing

정의

웹 서버에서 특정 디렉터리의 콘텐츠가 노출되어, 해당 디렉터리의 파일 리스트가 사용자의 웹 브라우저를 통해 표시되는 보안 취약점

공격 방식



< 일반적 동작 >



< 악의적 동작 >

대응 방안

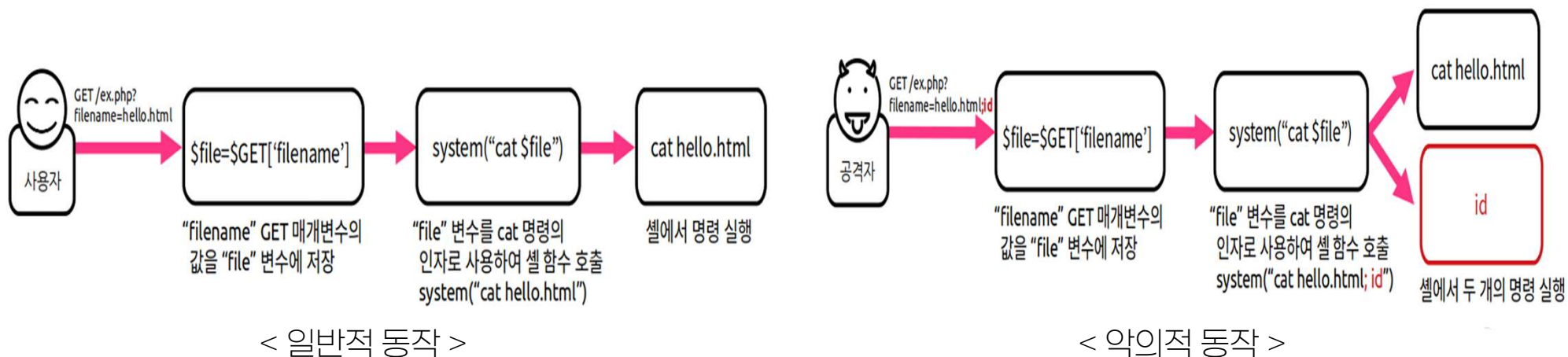
- 상위 디렉터리 접근 방지
- 보안 헤더 설정 강화
- 디렉토리 인덱싱 비활성화
- .htaccess 파일 사용(Apache 서버)

2.3 개념 학습 내용 – Command Injection

정의

공격자가 웹 애플리케이션에서 시스템 명령을 조작해, 악성 명령어를 삽입하고 실행시킬 수 있는 취약점

공격 방식



대응 방안

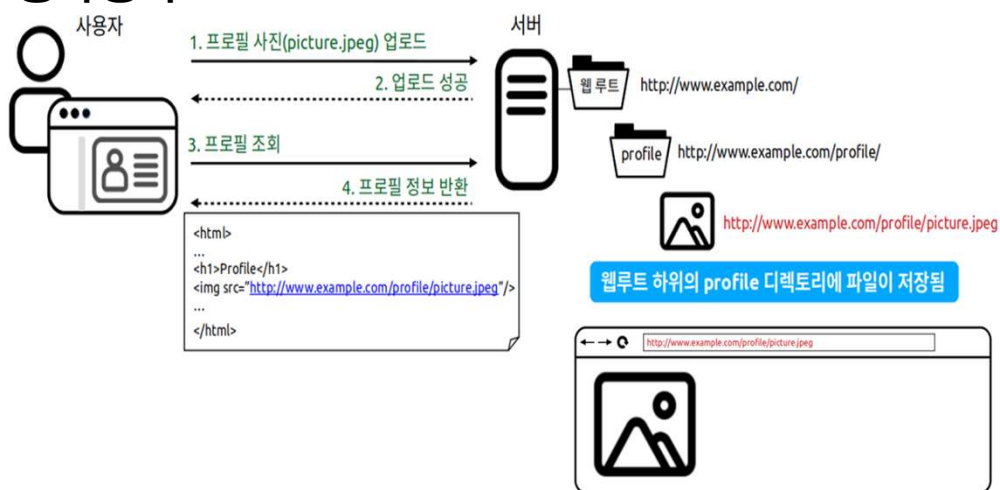
- 명령으로 전달되는 입력에 강한 입력 유효성 검사
- 최소 권한 사용
- 특수문자 이스케이프 처리 또는 요청 차단
- 자주 업데이트 및 패치

2.3 개념 학습 내용 – File Upload

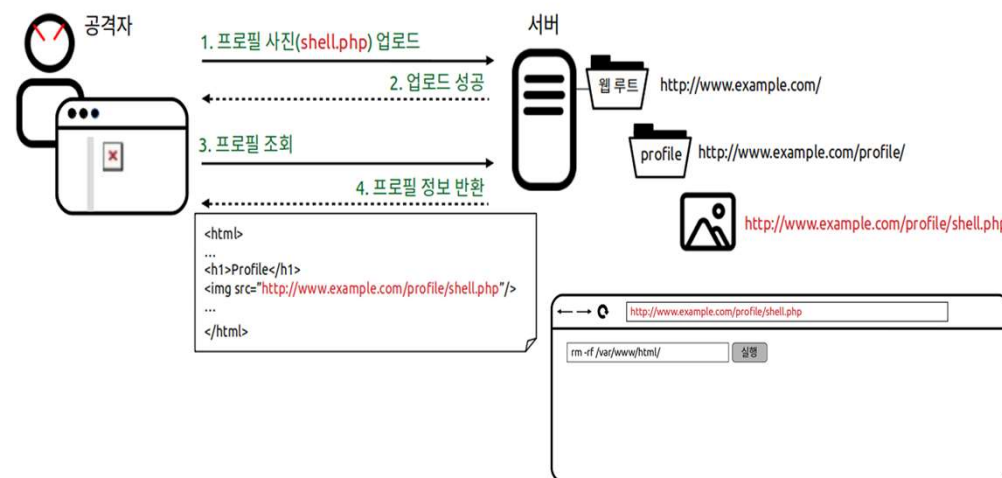
정의

파일 업로드 기능이 존재하는 웹 상에서, 서버에서 실행될 수 있는 스크립트 파일(asp, jsp, php 등)이 업로드 되어 실행될 수 있는 취약점

공격 방식



< 일반적 동작 >



< 악의적 동작 >

대응 방안

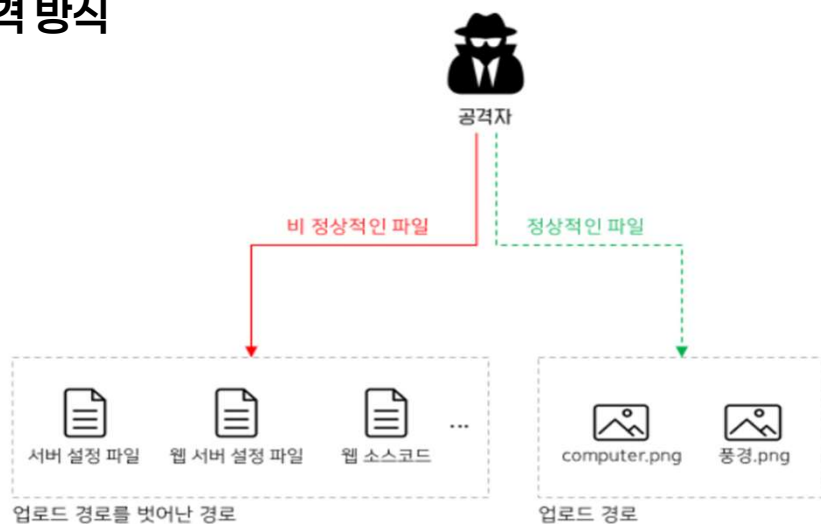
- 파일 확장자 검증
- 파일 Magic Number 검증
- 파일명 변경 또는 난독화

2.3 개념 학습 내용 – File Download

정의

파일 다운로드 기능 사용 시 임의의 문자나 주요 파일의 입력을 통해 웹 서버의 홈 디렉터리를 벗어나 임의의 위치에 있는 파일을 다운 가능한 취약점

공격 방식



< 악의적 동작 >

`http://www.test.co.kr/download.jsp?filename=../../../../etc/passwd`



공격자



웹 서버

대응 방안

- 파일 이름과 경로명을 데이터베이스에서 관리 및 비교
- 특정 디렉터리에서만 다운로드가 가능하도록 설정
- 다운로드 경로를 사용자가 확인할 수 없게 제한
- `../`와 같은 특수문자를 필터링 처리

2.4 실습 환경 구축

-- Server File version --



Ubuntu 22.04



PHP 8.1.2-1 Ubuntu2.18



Apache/2.4.52 (Ubuntu)



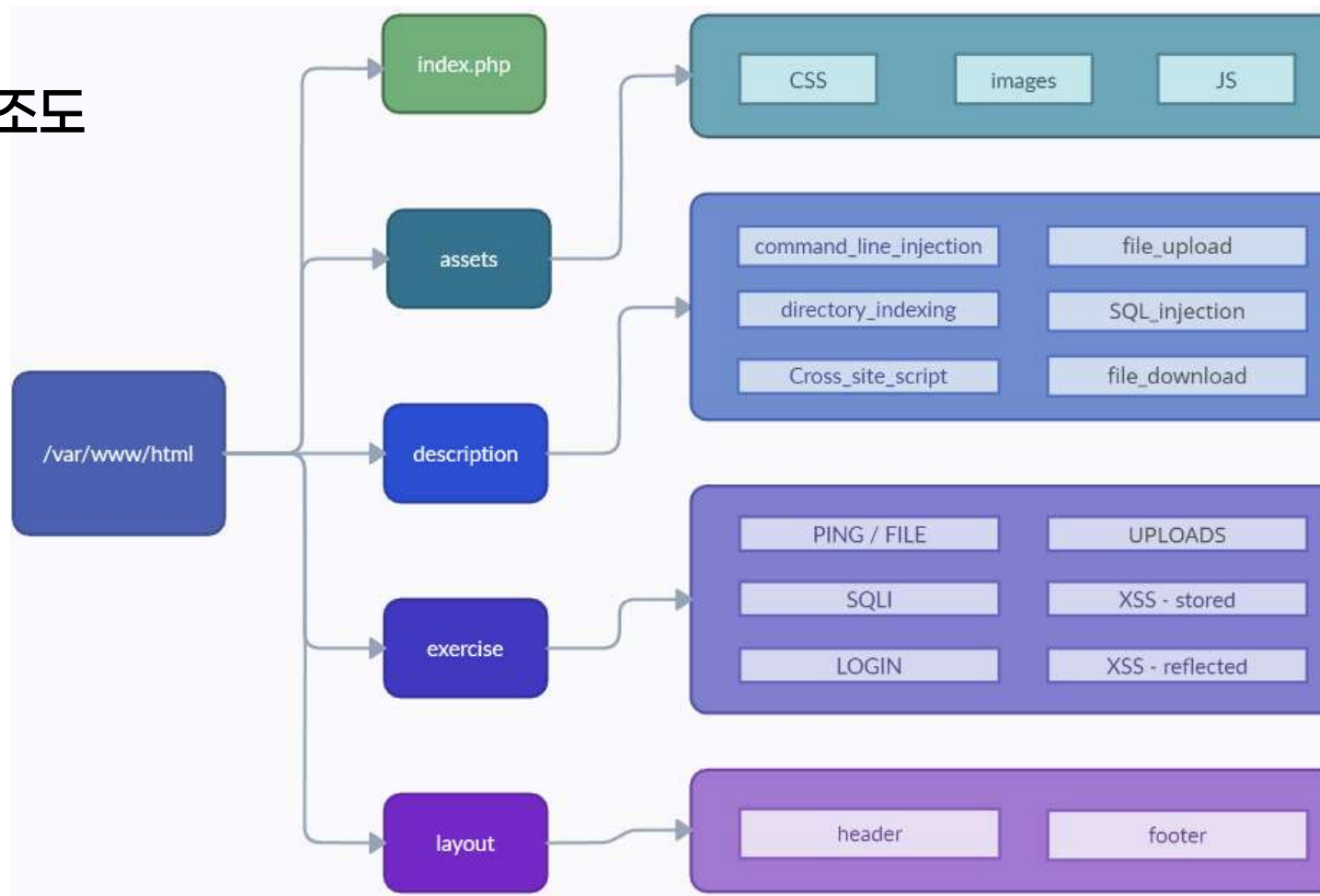
MySQL 8.0

개발 환경

플랫폼	WEB
언어	PHP 8.1.2-1
서버	Apache/2.4.52
DB	MySQL 8.0
호스팅환경	Local Hosting

2.4 실습 환경 구축

실습 파일 구조도



2.4 실습 환경 구축

DB 항목

```
mysql> desc users;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
name	varchar(30)	NO		NULL	
password	varchar(30)	NO		NULL	
email	varchar(40)	NO		NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> desc memo;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
title	varchar(255)	NO		NULL	
content	text	NO		NULL	
author	varchar(50)	NO		NULL	
created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

```
5 rows in set (0.00 sec)
```



2.5 실습 문제 구축 1_SQL Injection - Find User (★☆☆)

기본 화면

Web Vulnerability Lab

Find User

제출

Go to Find user 2 Page

정상적인 검색 출력

Web Vulnerability Lab

Find User

제출

ID: 1 이름: admin 이메일: admin@example.com

Web Vulnerability Lab

Find User

제출

ID: 2 이름: user1 이메일: user1@example.com

2.5 실습 문제 구축 1_SQL Injection - Find User (★☆☆)

악의적인 검색 출력

Find User

'or 1=1 #

User 검색창에 간단한 SQL Injection
공격 구문을 넣으면 취약점 발생

제출

ID: 1	이름: admin	이메일: admin@example.com
ID: 2	이름: user1	이메일: user1@example.com
ID: 3	이름: guest	이메일: guest@test.com
ID: 4	이름: guest1	이메일: guest1@test.com
ID: 5	이름: guest2	이메일: guest2@test.com
ID: 6	이름: guest3	이메일: guest3@test.com
ID: 7	이름: guest4	이메일: guest4@test.com
ID: 8	이름: guest5	이메일: guest5@test.com
ID: 9	이름: guest6	이메일: guest6@test.com
ID: 10	이름: guest7	이메일: guest7@test.com

취약한 코드 부분

```
$name = $_REQUEST['name'];
```

```
$query = "SELECT id, name, email FROM users WHERE name = '$name'";  
$result = mysqli_query($conn, $query);
```



2.5 실습 문제 구축 2_XSS- XSS (★☆☆)

기본 화면

Web Vulnerability Lab

Memo Board - Search Memos

Title:

Title	Content	Author	Date
SQL Injection Example	This memo details SQL injection techniques.	Bob	2024-09-07 23:28:16
Server Security	A memo about securing web servers from attacks.	Charlie	2024-09-07 23:28:16
Cross-Site Scripting	An explanation on how XSS can be exploited.	David	2024-09-07 23:30:00
Password Best Practices	Best practices for securing user passwords.	Emma	2024-09-07 23:31:00
Database Encryption	The importance of encrypting sensitive data in databases.	Frank	2024-09-07 23:32:00
Phishing Attack Prevention	Tips on how to prevent phishing attacks.	Grace	2024-09-07 23:33:00
Web Application Firewalls	How to protect your website with WAFs.	Henry	2024-09-07 23:34:00
SSL/TLS Best Practices	Guidelines for implementing SSL/TLS on websites.	Isabella	2024-09-07 23:35:00
DNS Security	Ensuring that your DNS configurations are secure.	Jack	2024-09-07 23:36:00

정상적인 검색 출력

Web Vulnerability Lab

Memo Board - Search Memos

Title:

Search results for: 'Security'

Title	Content	Author	Date
Server Security	A memo about securing web servers from attacks.	Charlie	2024-09-07 23:28:16
DNS Security	Ensuring that your DNS configurations are secure.	Jack	2024-09-07 23:36:00

2.5 실습 문제 구축 2_XSS- XSS (★☆☆)

공격 예 1)

Memo Board - Search Memos

Title:

Title 검색창에 공격 구문을 넣으면 취약점 발생

59.15.158.20:7676 내용:

1

악의적인
검색 출력

공격 예 2)

Memo Board - Search Memos

Title:

document.cookie 공격 구문을 통해 쿠키값 획득

59.15.158.20:7676 내용:

Cookie=Security_Academy

```
if (isset($_GET['title']) && $_GET['title'] != '') {
    $sql = "SELECT title, content, author, created_at FROM memo WHERE title LIKE '%" .
        $_GET['title'] . "%'";
    echo "Search results for: '" . $_GET['title'] . "'";
} else {
    $sql = "SELECT title, content, author, created_at FROM memo";
}
```

취약한 코드 부분



2.5 실습 문제 구축

최강모의해커즈

3_Command Injection - Ping(★☆☆)

기본 화면

Web Vulnerability Lab

Ping Utility

Enter an IP address to check its connectivity:

CheckGo to Ping-2 Page

정상적인 작동

Ping Utility

Enter an IP address to check its connectivity:

CheckGo to Ping-2 Page

Command: ping -c 4 8.8.8.8

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=34.3 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=34.8 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=33.9 ms  
64 bytes from 8.8.8.8: icmp_seq=4 ttl=113 time=33.9 ms  
  
--- 8.8.8.8 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3004ms  
rtt min/avg/max/mdev = 33.903/34.229/34.836/0.381 ms
```

2.5 실습 문제 구축 3_Command Injection - Ping(★☆☆)

pwd 명령어를 통한 현재 경로 확인

ls 명령어를 통한 파일 목록 확인

Ping Utility

Enter an IP address to check its connectivity:

8.8.8.8 ; pwd

Check

Command: ping -c 4 8.8.8.8 ; pwd

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=34.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=33.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=56 time=33.9 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=56 time=34.3 ms
```

```
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 33.750/33.999/34.290/0.208 ms
/var/www/html/exercise
```

Enter an IP address to check its connectivity:

8.8.8.8 ; ls

Check

Command: ping -c 4 8.8.8.8 ; ls

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=33.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=33.5 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=56 time=33.0 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=56 time=33.6 ms
```

```
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 32.975/33.275/33.605/0.208 ms
```

```
flag.txt
login
login.tar
ping-2.php
ping.php
sqli_find_user.php
sqli_find_user_2.php
sqli_login.php
trash
upload-base64.php
upload.php
uploads
uploads-base64
xss - 복사본.php:Zone.Identifier
xss.php
xss_stored.php
```

취약한 코드 부분

```
if (isset($_POST['ipaddress'])) {
    $ipaddress = ($_POST["ipaddress"]);
    $cmd = "ping -c 4 " . $ipaddress;
    echo "<h3>Command: $cmd</h3>";
    $output = shell_exec($cmd);
    echo "<pre class='result-output'>$output</pre>";
}
```

2.5 실습 문제 구축

4_Directory Indexing - 홈페이지 내 자체 취약점 (★☆☆)

최강모의해커즈

7. 실습

- 해당 페이지에 **Directory Indexing** 취약점이 있습니다.
- 어떤 식으로 접근을 하면 해당 취약점을 이용할 수 있는지 확인해보세요.

해당 URL
접근 시,
Index of 페이지
접근 가능

← → ↻ △ 주의 요함 59.15.158.20:7676/description/

Index of /description

	Name	Last modified	Size	Description
📁	Parent Directory	-	-	-
📄	command_line_injection.php	2024-08-29 17:39	7.4K	
📄	directory_indexing.php	2024-09-02 22:27	6.2K	
📄	file_upload.php	2024-08-29 17:34	5.9K	
📄	sample.php	2024-08-29 09:47	3.2K	
📄	sqli.php	2024-09-06 11:27	7.6K	
📁	trash/	2024-08-29 17:22	-	
📄	xss.php	2024-09-07 23:30	4.8K	

Apache/2.4.52 (Ubuntu) Server at 59.15.158.20 Port 7676

← → ↻ △ 주의 요함 59.15.158.20:7676/exercise/

Index of /exercise

	Name	Last modified	Size	Description
📁	Parent Directory	-	-	-
📄	flag.txt	2024-08-28 16:18	37	
📄	login.tar	2024-09-06 10:28	13K	
📁	login/	2024-09-06 02:54	-	
📄	ping-2.php	2024-08-29 17:37	941	
📄	ping.php	2024-08-28 16:10	907	
📄	sqli_find_user.php	2024-09-05 13:59	2.2K	
📄	sqli_find_user_2.php	2024-09-07 22:58	1.8K	
📄	sqli_login.php	2024-09-05 14:19	2.2K	
📁	trash/	2024-08-29 17:44	-	
📄	upload-base64.php	2024-08-28 23:45	2.5K	
📄	upload.php	2024-08-29 17:36	1.8K	
📁	uploads-base64/	2024-08-29 16:57	-	
📁	uploads/	2024-09-07 22:59	-	
📄	xss - 복사본.php:Zone.Identifier	2024-09-07 23:28	0	
📄	xss.php	2024-09-07 23:56	1.6K	
📄	xss_stored.php	2024-09-07 23:30	1.0K	

Apache/2.4.52 (Ubuntu) Server at 59.15.158.20 Port 7676



2.5 실습 문제 구축

최강모의해커즈

5_File Upload - Upload-base64 (★★★)

기본 화면

File upload Vulnerability

파일 선택 선택된 파일 없음

Upload

이 페이지에는 업로드된 파일의 리스트를 출력하지 않습니다.

Hint.

1. 업로드된 파일은 `uploads-base64/`에 저장됩니다.
2. 업로드 파일 이름을 그대로 저장하지 않습니다.

파일명이 base64로 인코딩 되어 저장됨

정상적인 작동

Web Vulnerability Lab

File upload Vulnerability

파일 선택 선택된 파일 없음

Upload

File is successfully uploaded.

이 페이지에는 업로드된 파일의 리스트를 출력하지 않습니다.




Hint.

1. 업로드된 파일은 `uploads-base64/`에 저장됩니다.
2. 업로드 파일 이름을 그대로 저장하지 않습니다.

2.5 실습 문제 구축

5_File Upload - Upload-base64 (★★★)

파일 형식의 검사 없이 업로드 되며
악성 스크립트나 파일을 올릴 수 있다는 취약점 발생

	ZmxhZw==.txt	2024-08-29 10:50	35
	d2Vic2h1bGw=.php	2024-08-29 16:57	40
	dGVzdA==.php	2024-08-28 23:18	376

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {  
    if (isset($_FILES['file']) && $_FILES['file']['error'] == UPLOAD_ERR_OK) {  
        $fileName = pathinfo($_FILES['file']['name'], PATHINFO_FILENAME);  
        $fileExtension = pathinfo($_FILES['file']['name'], PATHINFO_EXTENSION);  
  
        $encodedFileName = base64_encode($fileName);  
        $uploadFile = $uploadDir . $encodedFileName . '.' . $fileExtension;  
  
        if (move_uploaded_file($_FILES['file']['tmp_name'], $uploadFile)) {  
            echo "<p>File is successfully uploaded.</p>";  
        } else {  
            echo "<p>File upload failed!</p>";  
        }  
    } else {  
        echo "<p>No file was uploaded or there was an upload error.</p>";  
    }  
}
```

취약한 코드 부분

3. 프로젝트 결과

3.1 문제 페이지 시연

3.2 최종 결과

3.3 향후 계획



3.1 문제페이지 시연 – XSS: xss_stored 문제 풀이

Web Vulnerability Lab

Welcome to the Web Vulnerability Lab

Security_Academy 4기 모의해킹 미니 프로젝트 입니다.

취약점에 대한 설명과 실습을 진행해보세요

Command Line Injection

SQL Injection (SQLi)

Cross Site Script (XSS)

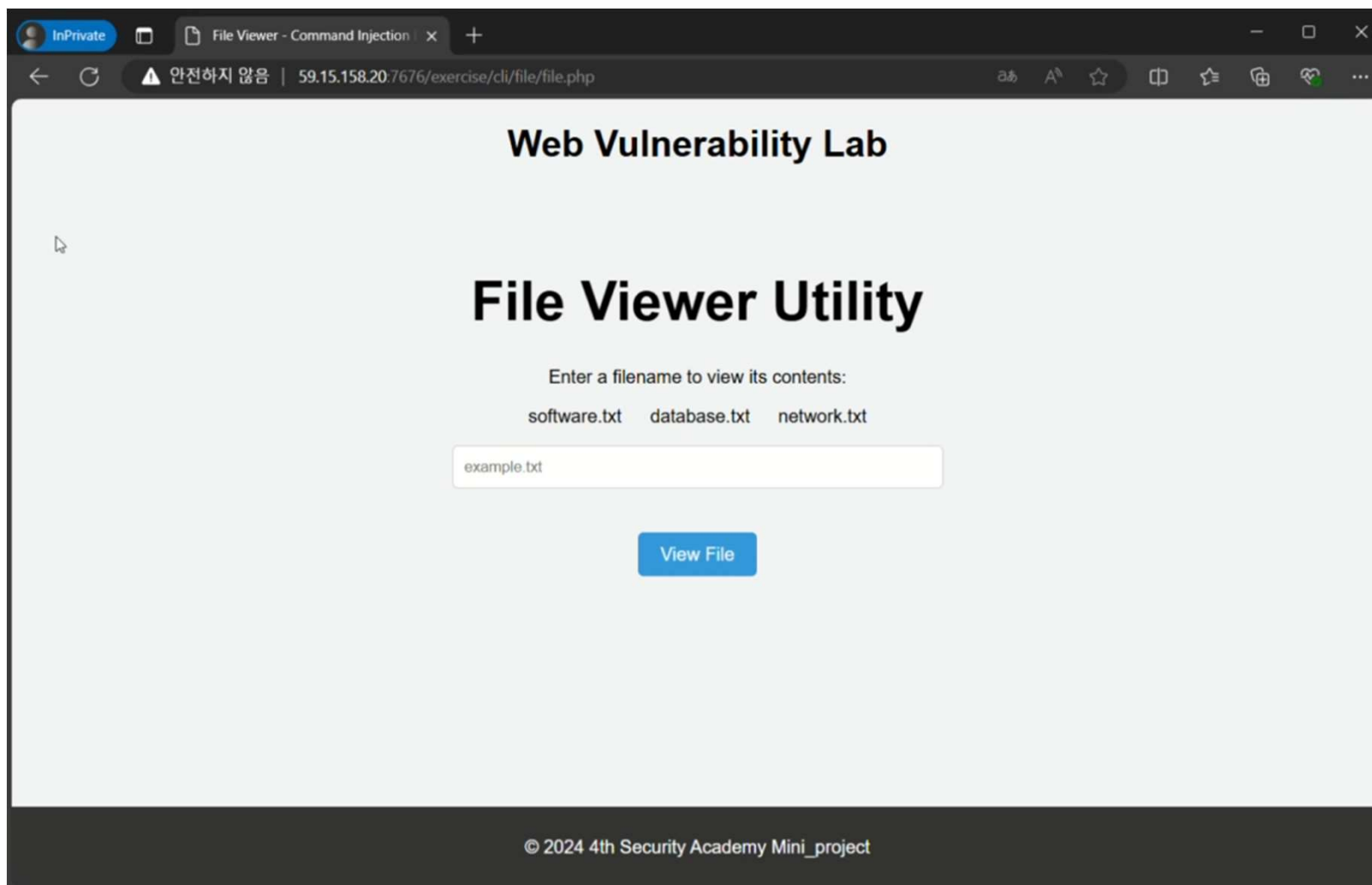
File Upload

Directory Indexing

file Download

© 2024 4th Security Academy Mini_project

3.1 문제페이지 시연 - Command Injection: file 문제 풀이



<http://59.15.158.20:7676/exercise/cli/file/file.php>

3.2 최종 결과



6개

학습한 취약점 종류



11문제

제작 문제



33일

총 진행일수

✓ 웹 취약점 이론 학습 및 실습을 통해 실제로 구현하며 웹 취약점에 대한 이해를 심화함

✓ 모의해킹 직무에 대한 이해도 높임

✓ 팀 단위 실습을 통해 협업 능력 강화함

목표 달성률

100%

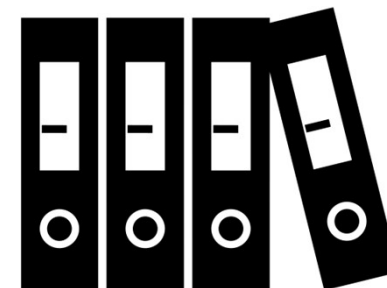
3.3 향후 계획



랭크, 점수
시스템 도입



문제에 대한 해설
및 힌트 제공



학습 사이트
제공



감사합니다

최강 모의해커즈

시큐리티 아카데미 직무 1반 모의 해킹 팀

