웹 취약점 진단 시스템 개발
(OWASP TOP 10 기반)

팀 명 : WeST

지도 교수 : 양환석 교수님

팀 장 : 박의명

팀 원 : 심명섭

오경준

송요섭

강보경

조예림

2019. 10.

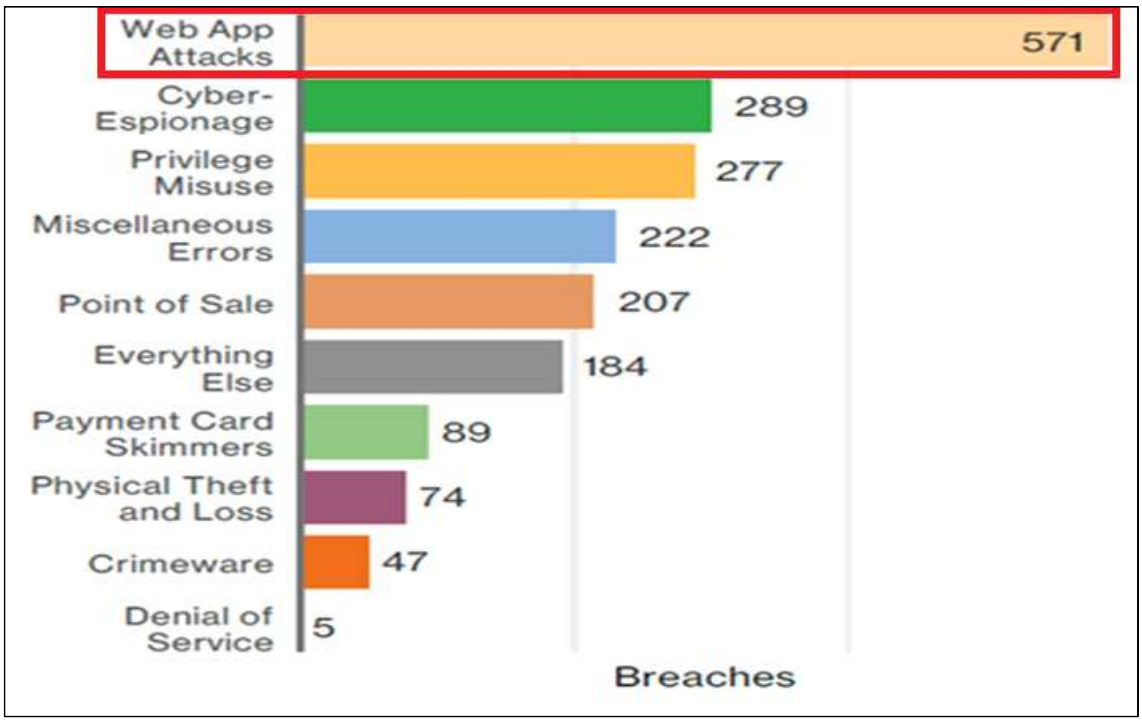
중부대학교 정보보호학과

목 차

1. 서 론

1.1 연구 목적 및 필요성

웹은 접근의 용이성 때문에 보안사고의 대부분을 차지할 정도로 집중적인 공격 대상이 되고 있어 그 범위와 피해가 날이 갈수록 증가하고 있다. 그래프(출처 : verizon data breach investigations report)를 보면 웹 어플리케이션을 공격하는 방식이 가장 많은 공격 비중을 차지한다는 것을 알 수 있다. 이에 OWASP TOP 10을 기준으로 하여 웹 취약점을 진단하고 대응 방안까지 알려주는 보안 시스템을 개발하려고 한다.



<그림 1 : Verizon 데이터 침해 조사 보고서 >

1.2 연구 주제 선정 이유

시중에 사용되고 있는 취약점 진단 서비스는 면담을 통해 수기로 이루어 지거나 직접 고객사와 만나며 이루어져야 한다는 제한성을 가지고 있다.

그에 비해 WeST팀은 직접 웹 취약점 진단 사이트를 구축함으로써 사용자의 인증 절차를 통해 접근을 편리성을 가진 웹 사이트와 발견된 취약점에 대한 대응 방안을 볼 수 있는 리포팅 시스템을 개발한다.

2. 관련연구

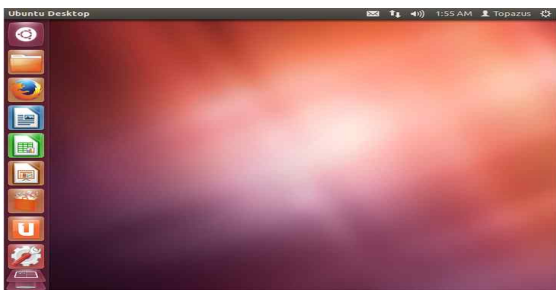
2.1 Python

python은 1991년 프로그래머인 Guido van Rossum이 발표한 고급 프로그래밍 언어로 플랫폼 독립적이며 인터프리터식, 객체지향적, 동적 타이핑 대화형 언어이다. Python은 비영리의 Python 소프트웨어 재단이 관리하는 개방형, 공동체 기반 개발 모델을 가지고 있다 C언어로 구현된 C파이썬 구현 사실상의 표준이다.

< 그림 2-1 : 파이썬 로고 >

2.2 Ubuntu

우분투는 컴퓨터에서 프로그램과 주변기기를 사용할 수 있도록 해주는 운영체제 중 하나이다. 안드로이드 운영체제처럼 리눅스 커널에 기반한 운영체제로 모바일과 데스크톱 PC, 서버에도 우분투 운영체제를 설치해 사용할 수 있다. 리눅스는 리누스 토발즈라는 개발자가 어셈블리어라는 프로그래밍 언어로 유닉스를 모델 삼아 개발한 오픈소스 운영체제이다. 우분투는 리눅스 OS의 배포판 중 하나로 특히 데스크톱 PC에서 사용할 수 있게 특화된 운영체제이다.



< 그림 2-2 : Ubuntu >

2.3 Apache

Apache HTTP 서버는 아파치 소프트웨어 재단에서 관리하는 HTTP 웹 서버이다. BSD, 리눅스 등 유닉스 계열 뿐 아니라 마이크로소프트 윈도우나 노벨 넷웨어 같은 기종에서

도 운용할 수 있다. 아파치 프로젝트는 공동제작과 합의에 기반한개발 프로세스와 오픈 되고 실용적인 소프트웨어 라이선스라는 특징으로 규정된다.

2.4 MySQL

MYSQL은 세계에서 가장 많이 쓰이는 오픈 소스의 관계형 데이터 베이스 관리 시스템이다. RDBMS라고도 불린다. 다중 스레드, 다중 사용자의 형식의 구조질의어 형식의 데이터베이스 관리 시스템으로써 MYSQL AB가 관리 및 지원하고 있으며 Qt처럼 이중 라이선스가 적용된다. 하나의 옵션은 GPL이며 GPL이외의 라이선스로 적용시키려는 경우 전통적인 지적재산권 라이선스의 적용을 받는다.

2.5 Php

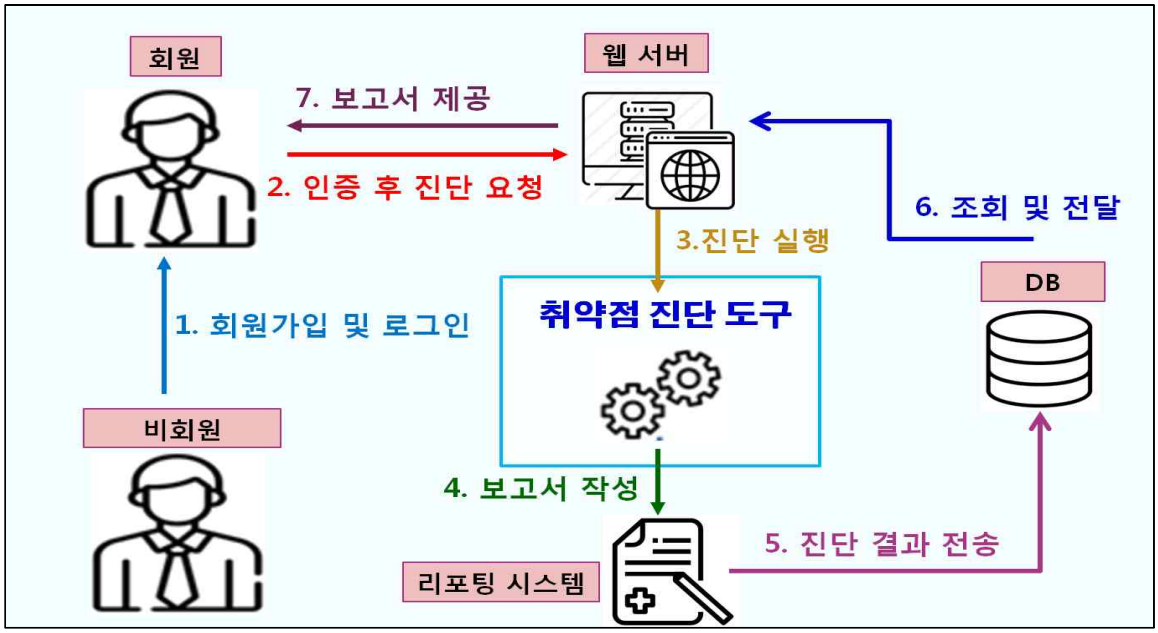
php는 C언어를 기반으로 만들어진 서버 측에서 실행되는 서버 사이드 스크립트 언어이다. 동적 웹 페이지를 만들기 위해 설계되었으며 이를 구현하기 위해 php로 작성된 코드를 html 소스 문서 안에 넣으면 php 처리 기능이 있는 웹 서버에서 해당 코드를 인식하여 작성자가 원하는 웹 페이지를 생성한다. 근래에는 PHP 코드와 HTML을 별도 파일로 분리하여 작성하는 경우가 일반적이며, PHP 또한 웹서버가 아닌 php-fpm(PHP FastCGI Process Manager)을 통해 실행하는 경우가 늘어나고 있다. 또한 PHP는 명령 줄 인터페이스 방식의 자체 인터프리터를 제공하여 이를 통해 범용 프로그래밍 언어로도 사용할 수 있으며 그래픽 애플리케이션을 제작할 수도 있다.

2.6 Html

하이퍼 본문 표식 달기 언어라는 의미의 웹 페이지를 위한 지배적인 마크업 언어다. HTML은 제목, 단락, 목록 등과 같은 본문을 위한 구조적 의미를 나타내는 것뿐만 아니라 링크, 인용과 그 밖의 항목으로 구조적 문서를 만들 수 있는 방법을 제공한다. 그리고 이미지와 객체를 내장하고 대화형 양식을 생성하는 데 사용될 수 있다.

3. 본론

3.1 구상도 설명



< 그림 3-1 : 구상도 >

3.2 취약점 진단 도구 제작 및 실행

OWASP Top 10과 KISA에서 발간한 주요정보통신기반시설 기술적 취약점 분석·평가 방법 상세가이드를 토대로 해당 취약점 진단 도구를 개발하였다.

OWASP Top 10 - 2017
A1:2017-Injection
A2:2017-Broken Authentication
A3:2017-Sensitive Data Exposure
A4:2017-XML External Entities (XXE)
A5:2017-Broken Access Control
A6:2017-Security Misconfiguration
A7:2017-Cross-Site Scripting (XSS)
A8:2017-Insecure Deserialization
A9:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring

< 그림 3-2 : OWASP Top 10 - 2017 >

먼저 비회원이 인증절차를 통하여 회원가입을 하고, 로그인 및 진단 요청을 하게 되면 취약점 진단 도구를 통해 해당 웹 사이트에 대한 취약점을 진단한다.
이후 리포팅 시스템을 통해 취약점 진단 보고서를 작성해 데이터베이스에 저장하고, 사용자에게 보고서를 전달해준다.

Web 취약점 분석·평가 항목

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
약한 문자열 강도	상	BF
불충분한 인증	상	IA
취약한 패스워드 복구	상	PR
크로스사이트 리퀘스트 변조(CSRF)	상	CF
세션 예측	상	SE
불충분한 인가	상	IN
불충분한 세션 만료	상	SC
세션 고정	상	SF
자동화 공격	상	AU
프로세스 검증 누락	상	PV
파일 업로드	상	FU
파일 다운로드	상	FD
관리자 페이지 노출	상	AE
경로 추적	상	PT
위치 공개	상	PL
데이터 평문 전송	상	SN
쿠키 변조	상	CC

< 그림 3-3 : Web 취약점 분석·평가 항목 >

```
def sqlinjection():
    sqlinjection_mysql = ['or 1=1--',
                          'W' or 1=1--',
                          'W" or 1=1--',
                          'W' or W'1W'=W'1',
                          'W" or W"1W"=W"1']

    sqlinjection_oracle = ['W' or 1=1#',
                           'W" or 1=1#',
                           'or 1=1#',
                           'W' or W'1W'=W'1',
                           'W" or W"1W"=W"1']
    return sqlinjection_mysql[1]

def send_post(data, next_url):

    is_cve = "양호"
    header = {
        'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0',
        'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
        'Accept-Language': 'en-US,en;q=0.5',
        'Accept-Encoding': 'gzip, deflate',
    }

    resp = requests.post(next_url, data=data, headers=header)

    if "Sign Off" in resp.text:
        is_cve = "취약"
    return (is_cve)
```

< 그림 3-4 : SQL INJECTION 취약점 도구 소스 일부 >

```
def dicxss(url):
    module_name = "XSS"
    contents = "Available Payloads:Wn"
    is_cve = "양호"

    url = "http://"+url
    getLinks(url)
    lst = list(pages)

    dic ={}
    d=0
    for i in lst:
        check = parse.urlparse(lst[int(d)])
        check.geturl()

        if check.query:

            dic.update(parse.parse_qs(check.query))
        d+=1


    fname = "payloads.txt"
    with open(fname) as f:
        content = f.readlines()
    payloads = [x.strip() for x in content]
    vuln = []
    for payload in payloads:
        for t in dic.keys():
```

< 그림 3-5 : XSS 취약점 도구 소스 일부 >

```
def dicrec(url):
    module_name = "directory listing"
    contents = "***** This website is W"SAFEW" from Directory listing"
    is_cve = "양호"
    c=0
    url = "http://"+url
    getLinks(url)
    lst = list(pages)
    for i in lst:
        toryurl = url+"/"+lst[int(c)]
        path = direc.get_urldirectorypath(toryurl)
        html = direc.return_souporhtml(path, "html")

        s = direc.regex_search('Index of /', html)

        if s == None:
            contents = "Wn***** This website is W"SAFEW" from Directory listing"

        else:
            contents = "Wn***** This website is W"RISKW" from Directory listing"
            is_cve = "취약"
            c+=1
    return (module_name, contents.strip(), is_cve)
```

<그림 3-6 : 디렉토리 리스팅 취약점 도구 소스 일부 >

```
def check_cve(get_header):
    module_name = "Check cve"
    contents = "Reference:Wn"
    is_cve = "양호"

    def cve1(key, contents, is_cve):
        r = requests.get('https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword='+str
        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(
            is_cve = "취약"
        return contents, is_cve
    if dic['server'] != 'hidden':
        contents, is_cve = cve1('server', contents, is_cve)
    if dic['lang'] != 'hidden':
        contents, is_cve = cve1('lang', contents, is_cve)
    if contents == "":
        contents = "no cve foundWn"
    return (module_name, contents.strip(), is_cve)
```

<그림 3-7 : 알려진 취약점 도구 소스 일부 >

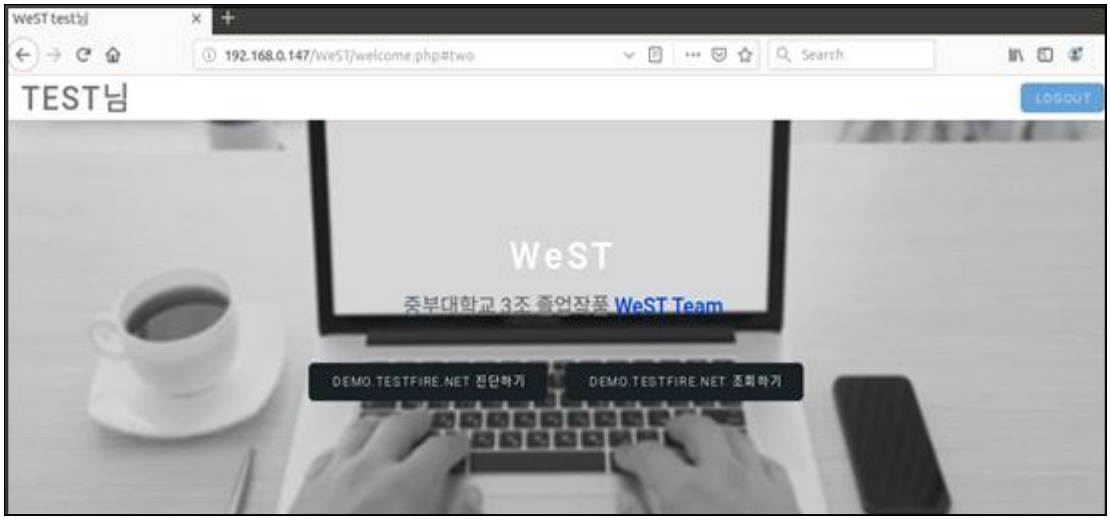
```
def adpage(domain):
    module_name = "Admin page"
    contents = ""
    is_cve = "양호"

    page= ["/admin",
           "/manager",
           "/master",
           "/system",
           "/administart"]

    url = "http://"+domain
    for pages in page:
        try:
            req = urllib.request.urlopen(url+ pages)
            contents += (pages + " server existWn")
            is_cve = "취약"
        except :
            continue
    if is_cve == "양호":
        contents += "no admin page found"
```

< 그림 3-8 : 관리자 페이지 노출 취약점 도구 소스 일부 >

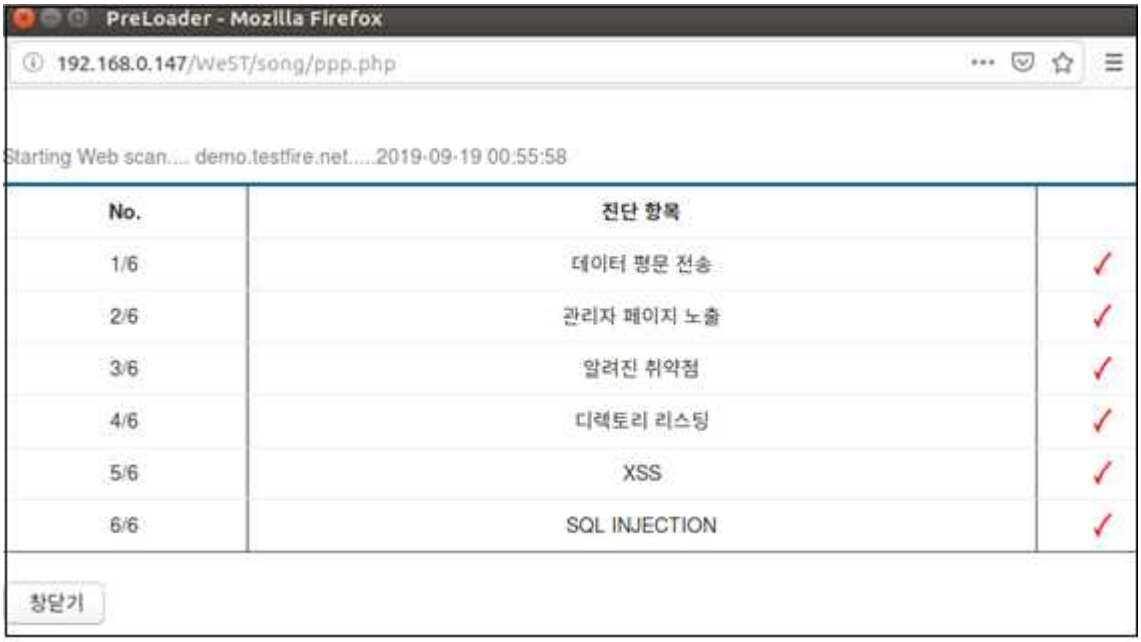
```
def scan(domain):
    module_name = "Port Scan"
    contents = ""
    is_cve = "양호"

    comport = {"FTP":21, "SMTP":25,"HTTP":80}
    ad = domain
    adip = socket.gethostbyname(ad)
    for PN, port in comport.items():
        try:
            s= socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            result = s.connect_ex((adip, port))
            banner = s.recv(1024)
            if result == 0:
                contents += str(port)+"/tcp   Open         "+str(PN) + "Wn"
                is_cve = "취약"

            elif banner==b'':

                contents += str(port)+"/tcp   noservice        "+str(PN) + "Wn"
            s.close()
        except:
            contents += str(port)+"/tcp   Closed        "+str(PN) + "Wn"
    return (module_name, contents.strip(), is_cve)
```

< 그림 3-9 : 데이터 평문 전송 취약점 도구 소스 일부 >

웹 사이트를 개발하여 웹 상에서 취약점 진단을 할 수 있도록 하였다.



< 그림 3-10 : 웹 사이트 메인 화면 >



< 그림 3-11 : 취약점 진단 화면 >

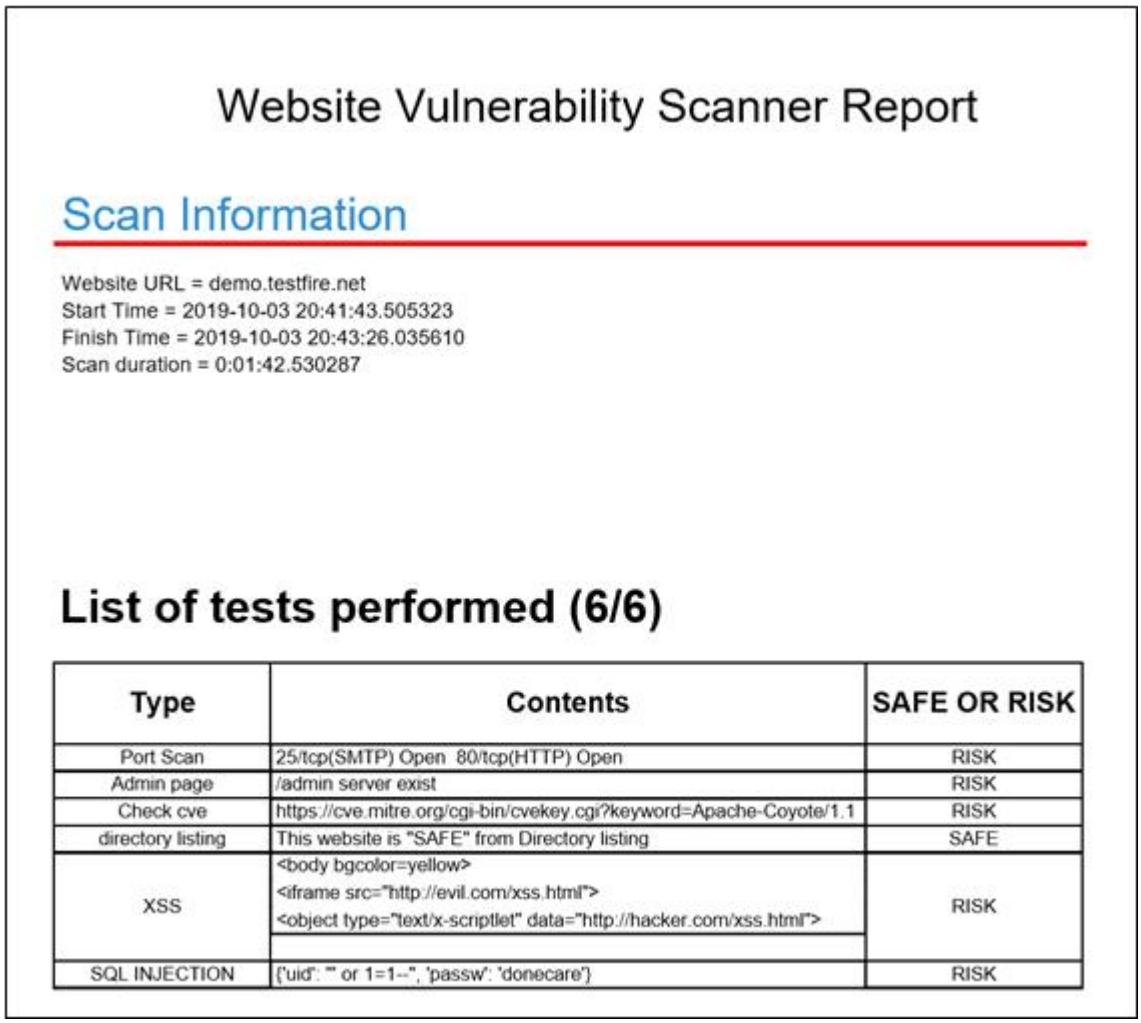
3.3 리포팅 시스템 개발

취약점 진단 이후 사용자에게 보고서를 제공하기 위하여 보기 쉽게 PDF파일로 생성하여 가독성을 높였다.

```
title = 'Web Scan Report'
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)
```

< 그림 3-12 : 리포팅 시스템 소스 일부 >

< 그림 3 – 13 : 진단 결과 화면 >



< 그림 3 – 14 : 리퍼팅 시스템으로 만든 PDF 파일 >

4. 결론

웹 취약점 진단 시스템을 만들어 XSS과 SQL Injection등의 취약점을 이용한 웹 공격에 대한 진단 시스템을 완성하는 것으로 목표로 설정하고 서비스를 제공하기 위해 인가된 사용자만 이용 가능 하도록 웹사이트를 개발하였다.
진단 이후 리포팅 시스템을 통해 관리자에게 대응 방안과 취약점들을 제공해 안정적이고 효율적으로 웹 사이트를 운영할 수 있도록 지원을 한다.
기대효과로는 웹 서비스를 운영하기 전이나 후에 취약점 진단 도구를 이용하여 관리자에게 대응 방안과 취약한 부분에 대한 보고서를 제공하고 온라인에서 자동으로 웹 사이트 취약점 진단을 진행하여 빠르고 간결하게 웹 취약점진단을 할 수 있다.

5. 참고자료

[1] verizon data breach investigations report

[2] KISA – 주요정보통신기반시설 취약점 진단 가이드 2014

[3] KISA – 홈페이지 취약점 진단 제거 가이드

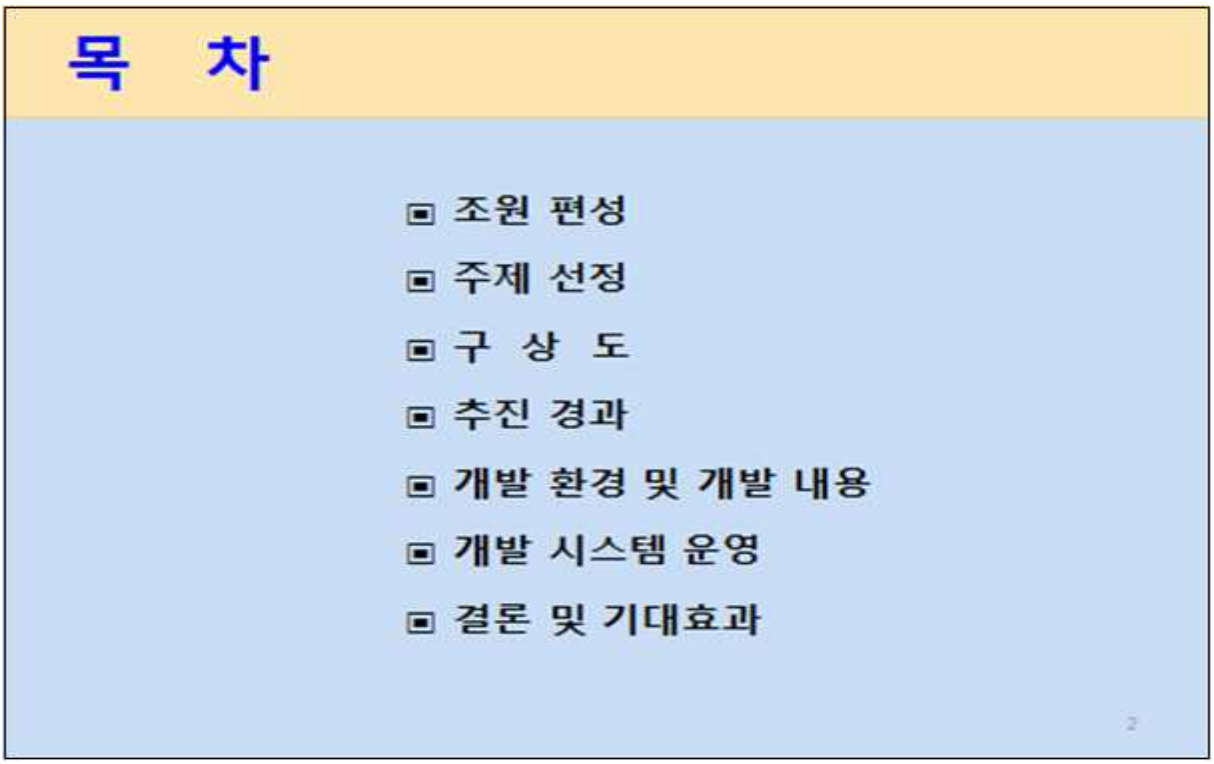
6. 별첨

6.1 발표 PPT

6.2 소스코드

6.1 발표 PPT


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
▣ 주제 선정
▣ 구 상 도
▣ 추진 경과
▣ 개발 환경 및 개발 내용
▣ 개발 시스템 운영
▣ 결론 및 기대효과

조원 편성

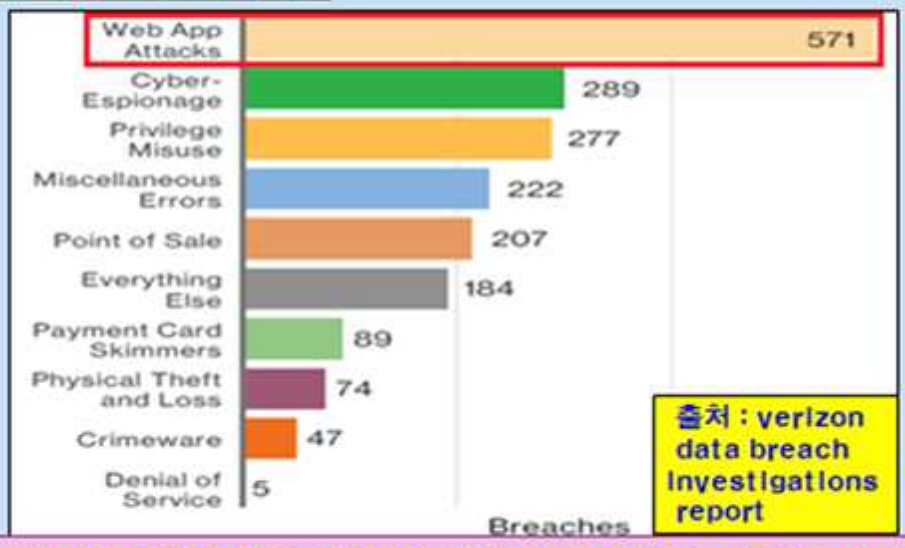
이 름	역 할
박의명 (팀장)	DB 구축 및 연동, PPT 작성, 보고서 작성
강보경	DB, 웹페이지 구축
송요섭	DB, 웹페이지 구축, 보고서 작성
심명섭	리포팅 시스템 개발
오경준	리포팅 시스템 개발, PPT 작성
조예림	리포팅 시스템 개발, 보고서 작성
공 통	OWASP10을 기반으로한 도구 개발

※ OWASP : The Open Web Application Security Project

3

주제 선정(1/2)

보안 취약점 분석



웹 애플리케이션을 공격하는 방식이 가장 많은 비중을 차지

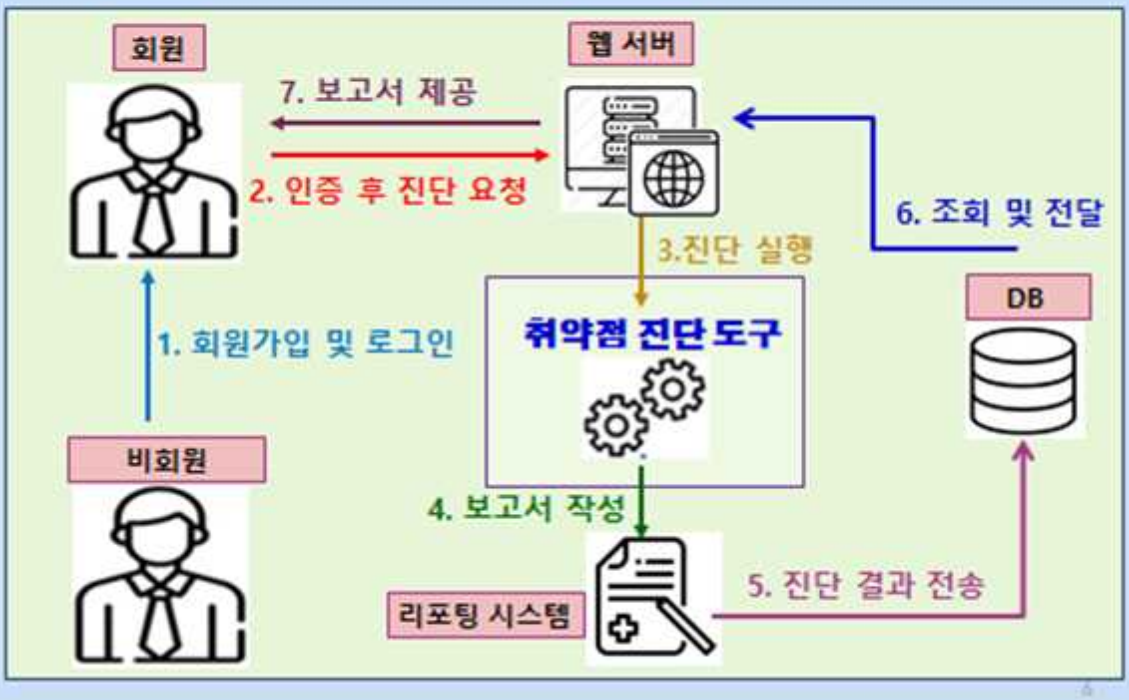
4

주제 선정(2/2)

보도 자료

웹 취약점을 진단하는 보안시스템 개발 추진

구 상 도



추진 경과

7

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

8

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

9

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

10

개발 환경 및 개발 내용(4/14)

웹 사이트 기본 화면 설계

메인 화면

로그인 화면

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

```
<h1>MeST</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)+"
        sys.exit(1)
print("일치하지 않습니다.
```

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

인증 성공

인증 실패

Crawling(웹 페이지의 데이터를 추출)을 이용한 도메인 인증

개발 환경 및 개발 내용(6/14)

메일서버 인증체계 개발

인증 메일 전송

```
try{
    $mail_content = "회원가입을 위해 이메일 인증이 필요합니다.<br/>가입을 완료하시려면 아
요.<br/><a href='http://".$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

13

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

14

개발 환경 및 개발 내용(8/14)

취약점 진단 시스템 개발(1/7)

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

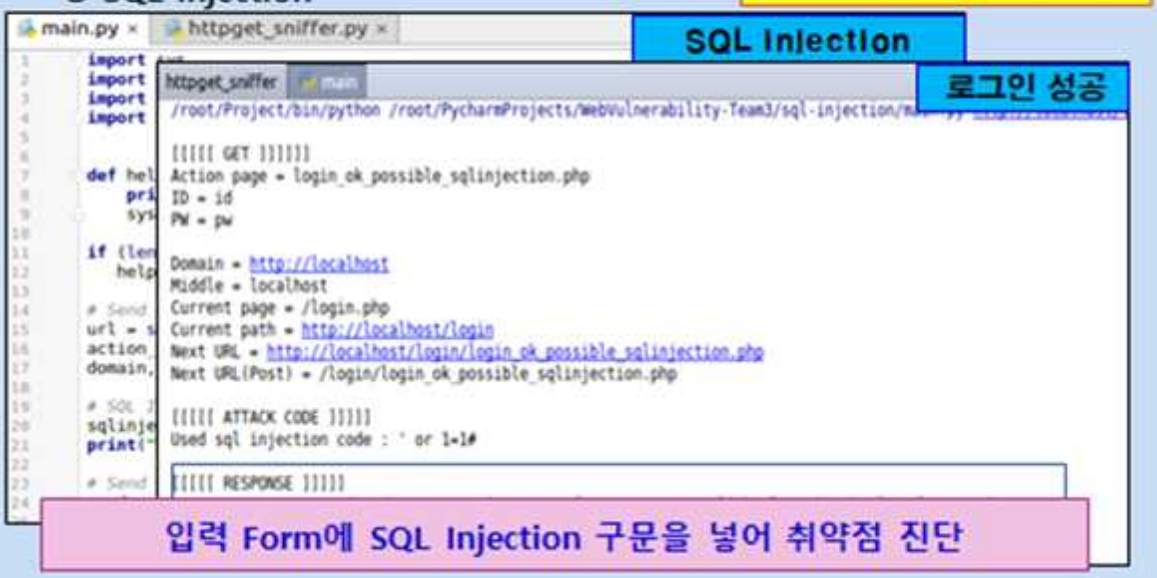
15

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



입력 Form에 SQL Injection 구문을 넣어 취약점 진단

16

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

URL 변경을 이용한 디렉토리 리스팅 취약점 진단

17

개발 환경 및 개발 내용(11/14)

취약점 진단 시스템 개발(4/7)

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

❍ XSS(Cross Site Scripting)

게시판에 Script 구문을 이용한 XSS 취약점 진단

18

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("\n Starting port scan ... at "+str(now))
    print(" Port scan report for "+ad+"\n")
    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

19

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

20

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
```

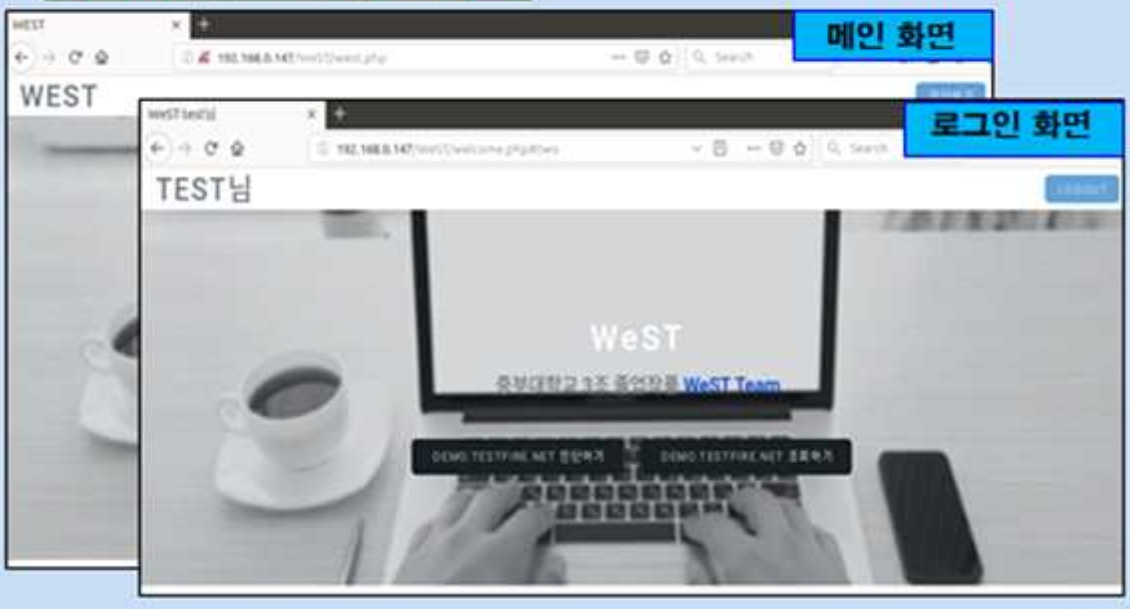
```
def check_cve(get_header):
    module_name = 'Check CVE'
    contents = ''
    is_cve = 'Safe'

    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

21

개발 시스템 운영(1/4)

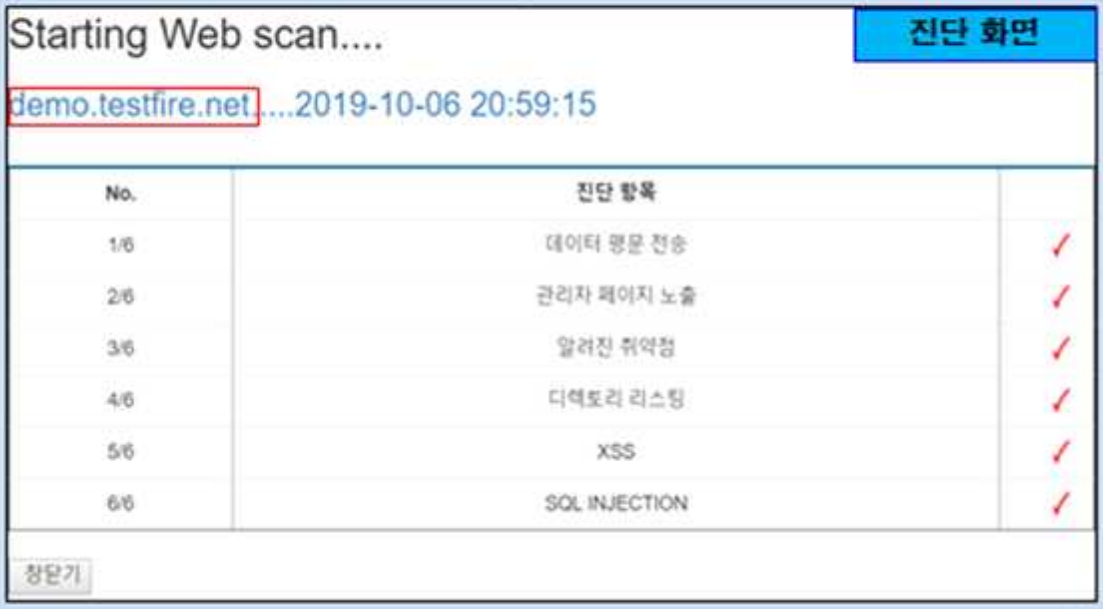
웹 사이트 기본 화면



22

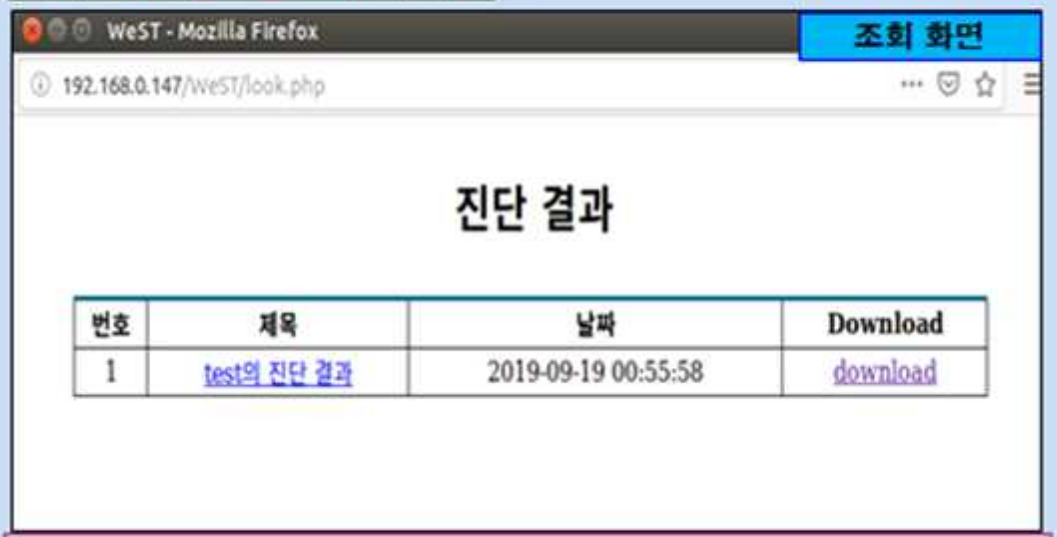
개발 시스템 운영(2/4)

웹 사이트 진단하기



개발 시스템 운영(3/4)

웹 사이트 조회하기



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', 'dorecare')	Risk

25

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

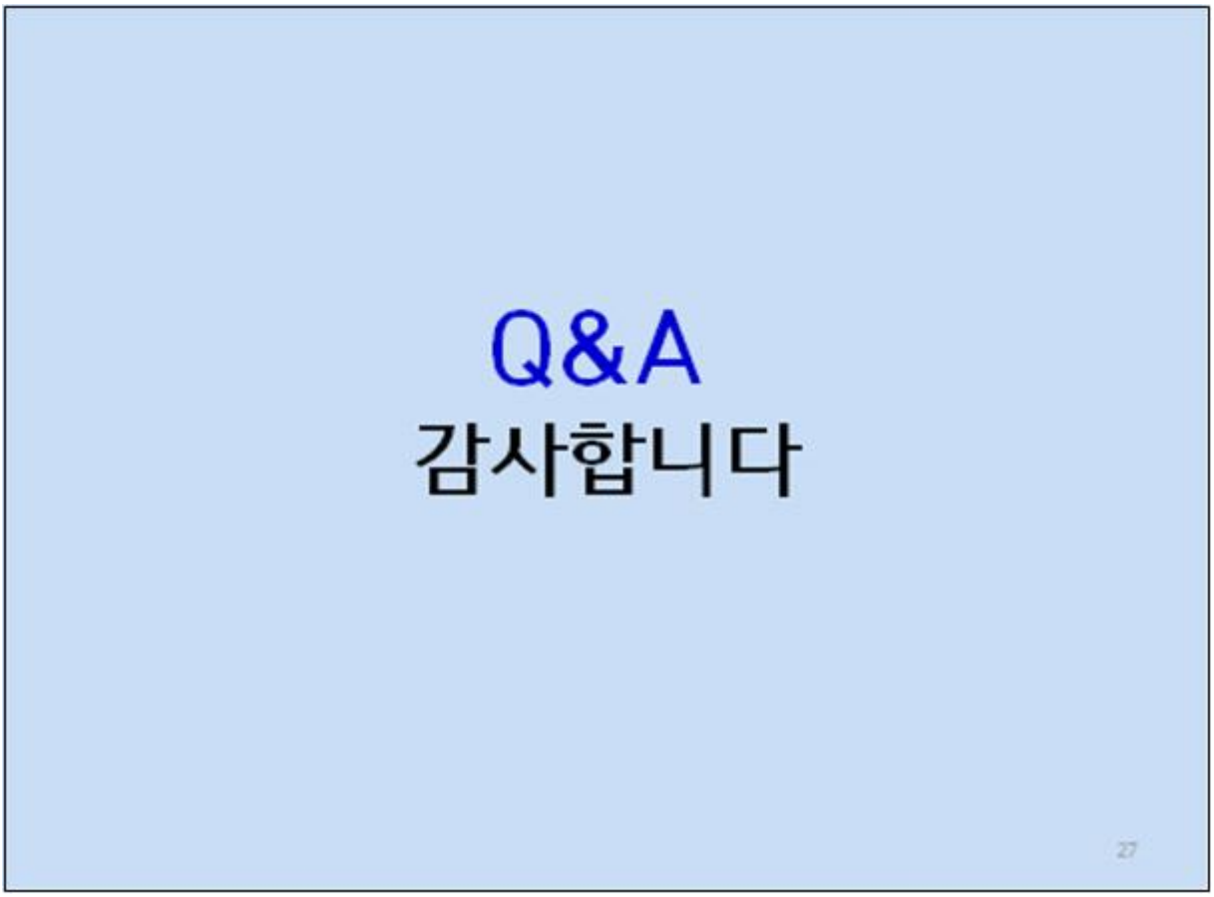
○ 기대효과

- 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 -

26



6.2 소스코드

<db.php>

```
<?php

    $host = 'localhost';
    $username = 'root';
    $password = 'song0401';
    $dbname = 'WeST';

    $options = array(PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES utf8');

    try {

        $con = new PDO("mysql:host={$host};dbname={$dbname};charset=utf8",$username,
$password);
    } catch(PDOException $e) {

        die("Failed to connect to the database: " . $e->getMessage());
    }
```

```
    $con->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $con->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);

    if(function_exists('get_magic_quotes_gpc') && get_magic_quotes_gpc()) {
        function undo_magic_quotes_gpc(&$array) {
            foreach($array as &$value) {
                if(is_array($value)) {
                    undo_magic_quotes_gpc($value);
                }
                else {
                    $value = stripslashes($value);
                }
            }
        }

        undo_magic_quotes_gpc($_POST);
        undo_magic_quotes_gpc($_GET);
        undo_magic_quotes_gpc($_COOKIE);
    }

    header('Content-Type: text/html; charset=utf-8');
    session_start();
?>
```

<dbinit.php>

```
<?php
error_reporting(E_ALL);
ini_set('display_errors',1);
include('check.php');


$databaseName = 'WeST';
$databaseUser = 'root';
$databasePassword = 'song0401';

데이터베이스 생성
$pdoDatabase = new PDO('mysql:host=localhost', $databaseUser, $databasePassword);
$pdoDatabase->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$pdoDatabase->exec('DROP DATABASE IF EXISTS WeST;');
$pdoDatabase->exec('CREATE DATABASE IF NOT EXISTS WeST DEFAULT CHARSET=utf8
COLLATE=utf8_general_ci');

테이블생성
$pdo = new PDO('mysql:host=localhost;dbname='.$databaseName, $databaseUser,
$databasePassword);
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
$pdo->exec('DROP TABLE IF EXISTS yo;');

$pdo->exec('CREATE TABLE `yo` (
 `uid` int(11) NOT NULL AUTO_INCREMENT,
 `username` VARCHAR(255) NOT NULL,
 `password` VARCHAR(255) NOT NULL,
 `email` VARCHAR(255),
 `domain` VARCHAR(255),
 `salt` VARCHAR(255) NOT NULL,
 `regtime` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,

 PRIMARY KEY (`uid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci')


echo "데이터베이스 초기화에 성공했습니다.₩n";
```

<check.php>

```
<?php
    error_reporting(E_ALL);
    ini_set('display_errors',1);

function is_login(){

    global $con;

    if (isset($_SESSION['user_id']) && !empty($_SESSION['user_id']) ){

        $stmt = $con->prepare("select username from yo where username=:username");
        $stmt->bindParam(':username', $_SESSION['user_id']);

        $stmt->execute();
        $count = $stmt->rowcount();

        if ($count == 1){

            return true; //로그인 상태
        }else{
            //사용자 테이블에 없는 사람
            return false;
        }
    }else{

        return false; //로그인 안된 상태
    }
}

//https://stackoverflow.com/a/46872528
```

```
function encrypt($plaintext, $salt) {
    $method = "AES-256-CBC";
    $key = hash('sha256', $salt, true);
    $iv = openssl_random_pseudo_bytes(16);

    $ciphertext = openssl_encrypt($plaintext, $method, $key, OPENSSL_RAW_DATA, $iv);
    $hash = hash_hmac('sha256', $ciphertext, $key, true);

    return $iv . $hash . $ciphertext;
}

function decrypt($ivHashCiphertext, $salt) {
    $method = "AES-256-CBC";
    $iv = substr($ivHashCiphertext, 0, 16);
    $hash = substr($ivHashCiphertext, 16, 32);
    $ciphertext = substr($ivHashCiphertext, 48);
    $key = hash('sha256', $salt, true);

    if (hash_hmac('sha256', $ciphertext, $key, true) !== $hash) return null;

    return openssl_decrypt($ciphertext, $method, $key, OPENSSL_RAW_DATA, $iv);
}


?>
```

< main.php >

```
<?php include('db.php');
        include('check.php');

        if(is_login()){

          if(isset($_SESSION['user_id'])) {
            header("Location: welcome.php");
                }
           }
?>

<!DOCTYPE HTML>
<html>
        <head>
                <title>WEST</title>
                <meta http-equiv="content-type" content="text/html; charset=utf-8" />
                <meta name="description" content="" />
                <meta name="keywords" content="" />

                <script src="js/jquery.min.js"></script>
                <script src="js/skel.min.js"></script>
                <script src="js/skel-layers.min.js"></script>
                <script src="js/init.js"></script>
```

```
        </head>
<script >
         function check() {
                  alert(' 로그인 먼저하세요.')
         }
</script>
<body id="top">
         <header id="header" class="skel-layers-fixed">
              <h1><a href="#">WeST</a></h1>
               <nav id="nav">
                  <ul>
                  <li><a href="#" class="button special" onclick="check()">진단하기</a></li>
                  </ul>
              </nav>
         </header>

         <section id="banner">
                <div class="inner">
                  <h2>WeST</h2>
                  <p><b>중부대학교 3조 졸업작품</b> <a href="#">WeST Team</a></p>
                  <ul class="actions">
                  <li><a href="#two" class="button big special">login</a></li>
                  <li><a href="#one" class="button big alt">vulnerabilities</a></li>
</ul>

 </div>
         </section>

         <section id="one" class="wrapper style1">
                <header class="major">
                  <h2>Kind Of Vulnerabilities</h2>
                     <p>XSS, Adminstrator Page, Known Vulnerabilities, Port Scan, SQL,
                  Directory Listing</p>
                </header>
         <div class="container">
              <div class="row">
                 <div class="4u">
                    <section class="special box">
                       <i class="icon fa-area-chart major"></i>
                       <h3>XSS</h3>
                        <p>Xss(Cross Site Script)는 웹 상에서 가장 기초적인 취약점 공격 방법의
일종으로, 악의적인 사용자기 공격하려는 사이트에 스크립트를 넣는 기법을 말합니다. 공격에 성
공하면 사이트에 접속한 사용자는 삽입된 코드를 실행하게 되며, 보통 의도치 않는 행동을 수행시
키거나 쿠키나 세션 토큰 등의 민감한 정보를 탈취합니다.</p>
                     </section
                  </div>
                  <div class="4u">
                     <section class="special box">
                        <i class="icon fa-refresh major"></i>
                        <h3>Adminstrator Page</h3>
                        <p>admin, adminstartor등 관리자페이지가 인터넷을 통해 추측성 접근이
```

가능할 경우, 공격자의 주 타겟이 될 수 있으며, 이는 공격자의 SQL 인젝션, Brute-Force공격 등 다양한 형태의 공격의 빌미를 제공하는 취약점입니다.
</section>

<section class="special box">

<h3>Known Vulnerabilities</h3>
대게 라이브러리, 프레임워크 및 다른 소프트웨어 모듈 같은 컴포넌트는 애플리케이션과 같은 권한으로 실행됩니다. 알려진 취약점이 있는 컴포넌트를 사용한 애플리케이션과 API는 애플리케이션 방어를 약화시키거나 다양한 공격과 영향을 주는 취약점입니다.
</section>

</section>

<section id="one" class="wrapper style1_2">

<section class="special box">

<h3>Port Scan</h3>

포트 스캔이란, 대상의 어떤 포트가 열려 있는지 확인하는 작업으로 침입 전 취약점을 분석하기 위한 사전 작업이라 볼 수 있다, 스캐닝을 통해 대상의 네트워크 보안장비 사용 현황, 우회 가능 네트워크 구조, 해당 네트워크 내의 시스템 플랫폼 형태, 시스템 운영체제의 커널 버전, 제동하는 서비스 등에 대한 정보를 알 수 있어 포트가 열려 있을 시 취약할수 있는 취약점이다.
</section>

<section class="special box">

<h3>SQL</h3>
SQL Injection이란, 대표적인 웹 어플리케이션 취약점 중 하나입니다.
애플리케이션에서 서버로 전달되는 명령, 쿼리, 스크립트등의 값을 변도하여 비정상적인 방법으로 시스템에 접근하는 공격기법입니다. 또한 웹 어플리케이션에서만 국한되지 않고 데이터베이스와 연결된 모든 어플리케이션에서 고려해 볼 수 있는 공격 기법입니다.
</section>

<section class="special box">

<h3>Directory Listing</h3>
Directorying 취약점은 브라우징 하는 모든 파일을 보여줍니다. 원래의 모적은 문서의 공유로 파일 탐색기처럼 원하는 문서로 바로 찾아갈 수 있게 하는 용도였지만 최근에는 문서의 저장 및 열람이 가능하다면 문서의 취약점(백업파일 및 소스 코드, 스크립트 파일의 유출로 인한 계정 정보 유출 등)을 이용해 악의적인 목적을 갖고 있는 사람들에게 탈취 및 웹 서버

```
공격이 이루어질 수 있는 취약점입니다.</p>
            </section>
          </div>
        </div>
</section>


<!-- Two -->
    <section id="two" class="wrapper style2">

        <header class="major">
                    <h2>login</h2>
        </header>

     <div class="content">
          <div class="signin-cont cont">
            <form  method="post">
             <input  id="login__username"  name="user_name"  type="text"  class="inpt"
placeholder="Username"required  autocomplete="off"  readonly
onfocus="this.removeAttribute('readonly');" />

              <input id="login__password" name="user_password" type="password" class="inpt"
placeholder="Password" required autocomplete="off" readonly
onfocus="this.removeAttribute('readonly');" />

              <div class="submit-wrap">
                  <input type="submit" name="login" class="submit" value="Login">


                  <a href="new_login.php" class="more">New_Login</a>
              </div>
            </form>
          </div>


        </div>
      </section>
</body>
</html>

<?php

    $login_ok = false;

    if ( ($_SERVER['REQUEST_METHOD'] == 'POST') and isset($_POST['login']) )
    {
                $username=$_POST['user_name'];
                $userpassowrd=$_POST['user_password'];

                if(empty($username)){
                        $errMSG = "아이디를 입력하세요.";
                }else if(empty($userpassowrd)){
                        $errMSG = "패스워드를 입력하세요.";
```

```
                }else{


                        try {

                                $stmt = $con->prepare('select * from yo where
username=:username');

                                $stmt->bindParam(':username', $username);
                                $stmt->execute();

                        } catch(PDOException $e) {
                                die("Database error. " . $e->getMessage());
                        }

                        $row = $stmt->fetch();
                        $salt = $row['salt'];
                        $password = $row['password'];

                        $decrypted_password = decrypt(base64_decode($password), $salt);

                        if ( $userpassowrd == $decrypted_password) {
                                $login_ok = true;
                        }
                }




                  if(isset($errMSG)
                 echo "<script>alert('$errMSG')</script>";

        if ($login_ok){
                $_SESSION['user_id'] = $username;
                 header('Location:welcome.php');
                 session_write_close();
        }else{
                 echo "<script>alert('$username 인증 오류')</script>";
                 }
        }
 ?>
```

<New_login.php>

```
<?php
    include('db.php');
    include('check.php');

function validatePassword($password){
```

```
    if(strlen($password) < 8 || empty($password)) {
        return 0;
    }
    if((strlen($password) > 48)) {
        return 0;
    }

    if(preg_match('/[A-Z]/',$password) == (0 || false)){
        return 1;
    }
    if(!preg_match('/[₩d]/',$password) != (0 || false)){
        return 2;
    }
    if(preg_match('/[₩W]/',$password) == (0 || false)){
        return 3;
    }
    return true;
}

  if( ($_SERVER['REQUEST_METHOD'] == 'POST') && isset($_POST['submit']))
    {

          foreach ($_POST as $key => $val)
          {
               if(preg_match('#^__autocomplete_fix_#', $key) === 1){
                    $n = substr($key, 19);
                    if(isset($_POST[$n])) {
                         $_POST[$val] = $_POST[$n];


              }
          }
          }

        $username=$_POST['newusername'];
        $password=$_POST['newpassword'];
        $confirmpassword=$_POST['newconfirmpassword'];
        $email=$_POST['newemail'];
        $domain=$_POST['newdomain'];

       if ($_POST['newpassword'] != $_POST['newconfirmpassword']) {
  $errMSG = "<script>alert('패스워드가 일치하지 않습니다.')</script>";
        }

        if(empty($username)){
            $errMSG = "<script>alert('아이디를 입력하세요')</script>";
        }
        else if(empty($password)){
            $errMSG = "<script>alert('패스워드을 입력하세요.')</script>";
        }
        else if(empty($email)){
            $errMSG = "<script>alert('email을 입력하세요.')</script>";
        }
```

```
        else if(empty($domain)){
            $errMSG = "<script>alert('domain을 입력하세요.')</script>";
        }

        try {
            $stmt = $con->prepare('select * from yo where username=:username');
            $stmt->bindParam(':username', $username);
            $stmt->execute();

            } catch(PDOException $e) {
                die("Database error: " . $e->getMessage());
            }

            $row = $stmt->fetch();
            if ($row){
                $errMSG = "<script>alert('이미 존재하는 아이디입니다')</script>";
            }
    if(!isset($errMSG))
      {
        try{
         $stmt = $con->prepare('INSERT INTO yo(username, password, email, salt,
                  domain) VALUES(:username, :password, :email, :salt, :domain)');
         $stmt->bindParam(':username',$username);
         $salt = bin2hex(openssl_random_pseudo_bytes(32));
         $encrypted_password = base64_encode(encrypt($password, $salt));
         $stmt->bindParam(':password', $encrypted_password);
         $stmt->bindParam(':email',$email);
         $stmt->bindParam(':salt',$salt);
         $stmt->bindParam(':domain',$domain);


        if($stmt->execute())
              {
                  $successMSG = "<script>alert('새로운 사용자를 추가했습니다.')</script>";
                   header("refresh:1;west.php");
         }
         else
         {
            $errMSG = "사용자 추가 에러";
         }
                    } catch(PDOException $e) {
                        die("Database error: " . $e->getMessage());
                    }
      }
   }
?>

<!DOCTYPE HTML>
<html>
        <head>
                <title>WEST</title>
                <meta http-equiv="content-type" content="text/html; charset=utf-8" />
                <meta name="description" content="" />
```

```
                <meta name="keywords" content="" />
                <script src="js/jquery.min.js"></script>
                <script src="js/skel.min.js"></script>
                <script src="js/skel-layers.min.js"></script>
                <script src="js/init.js"></script>
        </head>
<script >
function email_ck(){
    var email = document.getElementById("em").value;
    var domain = document.getElementById("dom").value;
    url = "email_ck.php?domain="+domain+"&email="+email;
window.open(url,        "window_name",'width=800,height=500,location=no,status=no,scrollbars=no,
resizable=no');
}
function check(){
   var ck = document.getElementById("aa").value;
   che = "email_ok.php";
window.open(url,        "window_name",'width=800,height=500,location=no,status=no,scrollbars=no,
resizable=no');
}
   function check1() {
     alert(' 로그인 먼저하세요.')
   }
</script>


<body id="top">
    <!-- Header -->
      <header id="header" class="skel-layers-fixed">
        <h1><a href="#">WeST</a></h1>
        <nav id="nav">
          <ul>
            <li><a href="#" class="button special" onclick="check1()">진단하기</a></li>
          </ul>
        </nav>
      </header>

     <section id="two" class="wrapper style2">
          <header class="major">
              <h2>새로운 사용자 추가</h2>
          </header>
          <div class="tabs">
                <span class="tab signin active"><a href="west.php"><b>Back</b></a></span>
          </div>
          <?php
           if(isset($errMSG)){
                    ?>
                       <div class="alert alert-danger">
                       <span class="glyphicon glyphicon-info-sign"></span>
                       <strong><?php echo $errMSG; ?></strong>
                       </div>
                       <?php
                }
```

```
                else if(isset($successMSG)){
                    ?>
                      <div class="alert alert-success">
                      <strong><span class="glyphicon glyphicon-info-sign"></span>
                      <?php echo $successMSG; ?></strong>
                       </div>
                <?php
                }
                ?>

    <div class="content">
            <div class="signin-cont cont">
                      <form method="post" enctype="multipart/form-data">
                      <? $r1 = rmd5(rand().mocrotime(TRUE)); ?>
<input type="text" name="<? echo $r1; ?>" class="inpt" placeholder="아    이디를 입력하세요."
autocomplete="off" readonly onfocus="this.removeAttribute('readonly');" />
<input type="hidden" name="__autocomplete_fix_<? echo $r1; ?>" value="newusername" />

                        <? $r2 = rmd5(rand().mocrotime(TRUE)); ?>
<input type="password" name="<? echo $r2; ?>" class="inpt" placeholder="패스워드를 입력하
세요" autocomplete="off" readonly onfocus="this.removeAttribute('readonly');" />
<input type="hidden" name="__autocomplete_fix_<? echo $r2; ?>" value="newpassword" />




<? $r3 = rmd5(rand().mocrotime(TRUE)); ?>
<input type="password" name="<? echo $r3; ?>" class="inpt" placeholder="패스워드를 다시 한
번 입력하세요" autocomplete="off" readonly onfocus="this.removeAttribute('readonly');" />
<input type="hidden" name="__autocomplete_fix_<? echo $r3; ?>" value="newconfirmpassword"
/>

<? $r4 = rmd5(rand().mocrotime(TRUE)); ?>
<input type="text" id="em"    name="newemail" name="<? echo $r4; ?>" class="inpt"
placeholder="이메일를 입력하세요" autocomplete="off" readonly
onfocus="this.removeAttribute('readonly');" />
<input type="hidden" name="__autocomplete_fix_<? echo $r4; ?>" value="newemail" />

<? $r5 = rmd5(rand().mocrotime(TRUE)); ?>
<input type="text" id="dom" name="newdomain" name="<? echo $r5; ?>" class="inpt"
placeholder="도메인을 입력하세요" autocomplete="off" readonly
onfocus="this.removeAttribute('readonly');" />
<input type="hidden" name="__autocomplete_fix_<? echo $r5; ?>" value="newdomain" />

                      <div class="do">
                              <input type="submit"  name="submit"  value="Register"
                               onclick="check()">

                              <input     type="submit"          name="aa"     value="진단하기"
                                onclick="email_ck()">
                      </div>
                      </form>
          </div>
```

```
        </div>
</section>
```

<welcome.php>

```
<?php
    include('db.php');
    include('check.php');

    if (is_login()){
        ;
    }else
        header("Location: west.php");


?>
<?php
  $user_id = $_SESSION['user_id'];


  try {
    $stmt = $con->prepare('select * from yo where username=:username');
    $stmt->bindParam(':username', $user_id);
    $stmt->execute();

   } catch(PDOException $e) {
    die("Database error: " . $e->getMessage());
   }

   $row = $stmt->fetch();
 ?>


<!DOCTYPE HTML>

<html>
  <head>
    <title>WeST <?php echo($row['username']);?>님</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  </head>
    <body id="top">

    <!-- Header -->
      <header id="header" class="skel-layers-fixed">
        <h1><a href="#"><?php echo $user_id; ?>님</a></h1>
        <nav id="nav">
          <ul>

            <li><a href="logout.php" class="button special">logout</a></li>
          </ul>
```

```
          </nav>
        </header>

      <!-- Banner -->
      <section id="banner">
          <div class="inner">
            <h2>WeST</h2>
       <p><b>중부대학교 3조 졸업작품</b> <a href="http://templated.co">WeST Team</a></p>
          <ul class="actions">
        <li> <input type="submit" value="<?php echo($row['email']);?> 진단하기" class="button big special" onclick="window.open('./song/ppp.php','window_name','width=800,height=500,location=no,status=no,scrollbars=no');"></li>
        <li><input type="submit" value="<?php echo($row['email']);?> 조회하기" class="button big alt" onclick="window.open('look.php','window_name','width=800,height=500,location=no,status=no,scrollbars=no');"></li>
          </ul>
          </div>
      </section>
```

<result.php>

```
<?php
      include('/var/www/html/WeST/db.php');
      include('/var/www/html/WeST/check.php');
      ?>

<?php
   $user_id = $_SESSION['user_id'];


   try {
      $stmt = $con->prepare('select * from yo where username=:username');
      $stmt->bindParam(':username', $user_id);
      $stmt->execute();

    } catch(PDOException $e) {
      die("Database error: " . $e->getMessage());
    }

    $row = $stmt->fetch();
?>

<!DOCTYPE html>
<html>
<head>
<title>Result</title>
</head>
<table class="table" border="1" >

<caption>Starting Web scan.... <a><?php echo($row['email']); ?></a>.....<?php
```

```
echo($row['regtime']); ?></caption>

<tr><th> No.  </th><th> 진단 항목 </th><th>  </th></tr>

<tr><td> 1/6 </td><td> 데이터 평문 전송</td><td> <div ><img id="l05" src="song.gif" width="20px" height="20px"></td></tr>

<tr><td> 2/6 </td><td> 관리자 페이지 노출</td><td > <div id ="l06" > <img id="l01" src="song.gif" width="20px" height="20px ">
</td></tr>

<tr><td> 3/6 </td><td> 알려진 취약점</td><td><div ><img id="l02" src="song.gif" width="20px" height="20px"></div>
</td></tr>

<tr><td> 4/6 </td><td> 디렉토리 리스팅</td><td> <div ><img id="l03" src="song.gif" width="20px" height="20px" ></td></tr>

<tr><td> 5/6 </td><td> XSS</td><td ><div ><img id="l04" src="song.gif"  width="20px" height="20px" ></div><div id="l06">
</div></td></tr>

<tr><td> 6/6 </td><td> SQL INJECTION</td><td> <div ><img id="loading2" src="a.jpg" width="20px" height="20px"><img id="loading" src="song.gif" width="20px" height="20px "></div>
<iframe src="result.php" id="iframe100"></iframe>
</td></tr>

<script>
var ld= document.getElementById("loading");
window.addEventListener("load", function(){
    ld.style.display="none";
    });
</script>
</table>
</div>
<input type='button' value=" 창닫기" onclick='self.close()'>
</body>
</html>
```

<lnquiry.php>

```
<?php
      include('db.php');
      include('check.php');

      if (is_login()){
            ;
      }else
            header("Location: west.php");
```

```
?>
<?php
  $user_id = $_SESSION['user_id'];

  try {
    $stmt = $con->prepare('select * from yo where username=:username');
    $stmt->bindParam(':username', $user_id);
    $stmt->execute();
   } catch(PDOException $e) {
    die("Database error: " . $e->getMessage());
   }
   $row = $stmt->fetch();
?>
<!doctype html>
<html lang="kr">
  <head>
  <meta charset="UTF-8">
  <title>WeST</title>
</head>

<body>
       <p>진단 결과</p>
       <table class="table" border="1">
       <tr> <th>번호</th><th>제목</th><th>날짜</th><th>Download</th> </tr>
       <tbody>
       <tr> <td>1</td>
       <td><?php echo $row['username']?>의 진단 결과</td>
       <td><?php echo $row['regtime']?></td>
       <td><a  href="download.php">download</a></td> </tr>
        </tbody>
        </table>
</body>
</html>
```

<download.php>

```
<?php
    include('db.php');
    include('check.php');

    if (is_login()){
        ;
    }else
        header("Location: west.php");
?>
<?php
  $user_id = $_SESSION['user_id'];

  try {
    $stmt = $con->prepare('select * from yo where username=:username');
    $stmt->bindParam(':username', $user_id);
```

```
    $stmt->execute();

  } catch(PDOException $e) {
    die("Database error: " . $e->getMessage());
  }

  $row = $stmt->fetch();
?>
<?php
$user = $_SESSION['user_id'];
$filename = "cve_log_$user.pdf";
$file = "./song/" . $filename;

if (is_file($file)) {

    if (preg_match("/MSIE*/", $_SERVER['HTTP_USER_AGENT'])) {
        header("Content-type: application/octet-stream");
        header("Content-Length: ".filesize("$file"));
        header("Content-Disposition: attachment; filename=$filename"); // 다운로드되는 파일명
(실제 파일명과 별개로 지정 가능)
        header("Content-Transfer-Encoding: binary");
        header("Cache-Control: must-revalidate, post-check=0, pre-check=0");
        header("Pragma: public");
        header("Expires: 0");
    }
    else {
        header("Content-type: file/unknown");
        header("Content-Length: ".filesize("$file"));
        header("Content-Disposition: attachment; filename=$filename"); // 다운로드되는 파일명
(실제 파일명과 별개로 지정 가능)
        header("Content-Description: PHP3 Generated Data");
        header("Pragma: no-cache");
        header("Expires: 0");
    }

    $fp = fopen($file, "rb");
    fpassthru($fp);
    fclose($fp);
}
else {
    echo "해당 파일이 없습니다.";
}
?>
```

<Vulnerability.py>

```
# -*- coding: utf-8 -*-
import socket
import datetime
import urllib.request
import requests
from bs4 import BeautifulSoup
import re
```

```
import sys
from urllib import parse
from urllib.request import urlopen
import time
import os
from fpdf import FPDF

def scan(domain):
    module_name = "Port Scan"
    contents = ""
    is_cve = "Safe"

    comport = {"FTP":21, "SMTP":25,"HTTP":80}
    ad = domain
    adip = socket.gethostbyname(ad)
    for PN, port in comport.items():
        try:
            s= socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            result = s.connect_ex((adip, port))
            banner = s.recv(1024)
            if result == 0:
                contents += str(port)+"/tcp("+str(PN)+") Open   "
                is_cve = "Risk"
            elif banner==b'':
                contents += str(port)+"/tcp("+str(PN)+")    noservice₩n"
            s.close()
        except:
            continue
    return (module_name, contents.strip(), is_cve)

def adpage(domain):
    module_name = "Admin Page"
    contents = ""
    is_cve = "Safe"

    page= ["/admin",
           "/manager",
           "/master",
           "/system",
           "/administart"]

    url = "http://"+domain
    for pages in page:
        try:
            req = urllib.request.urlopen(url+ pages)
            contents += (pages + " server exist₩n")
            is_cve = "Risk"
        except :
            continue
    if is_cve == "Safe":
        contents += "no admin page found"
    return (module_name, contents.strip(), is_cve)
```

```
def get_header(domain):
    global req, header, dic, cve
    req = requests.get('http://'+domain)
    header = req.headers
    dic = {'server' : 'hidden', 'os' : 'hidden', 'lang' : 'hidden'}
    cve = {'server' : '', 'lang' : ''}
    if 'Server' in header:
        server=header['Server']
        s = server.split(' ')
        for i, a in enumerate(dic.keys()):
            dic[a] = s[i]
            if (len(s) < len(dic)):
                break
    else:
        pass

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])
            is_cve = "Risk"
        return contents, is_cve
    if dic['server'] != 'hidden':
        contents, is_cve = cve1('server', contents, is_cve)
    if dic['lang'] != 'hidden':
        contents, is_cve = cve1('lang', contents, is_cve)
    if contents == "":
        contents = "no cve found"
    return (module_name, contents.strip(), is_cve)

def help():
    print('Usage: ./main url')
    sys.exit(1)

def sqlinjection():
    sqlinjection_mysql = ['or 1=1--',
                          '₩' or 1=1--',
                          '₩" or 1=1--',
                          '₩' or ₩'1₩'=₩'1',
                          '₩" or ₩"1₩"=₩"1']
```

```
    sqlinjection_oracle = ['₩' or 1=1#',
                           '₩" or 1=1#',
                           'or 1=1#',
                           '₩' or ₩'1₩'=₩'1',
                           '₩" or ₩"1₩"=₩"1']
    return sqlinjection_mysql[1]

def send_post(data, next_url):

    is_cve = "Safe"
    header = {
        'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101
Firefox/66.0',
        'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
        'Accept-Language': 'en-US,en;q=0.5',
        'Accept-Encoding': 'gzip, deflate',
    }

    resp = requests.post(next_url, data=data, headers=header)

    if "Sign Off" in resp.text:
        is_cve = "Risk"
    return (is_cve)

def get_domain(url):
    domainp = '^(https?:₩/₩/)?([₩da-z₩.-]+)'
    domain = re.compile(domainp).match(url).group()
    return domain
def sqltest(url):
        module_name = "SQL Injection"
        is_cve = "Safe"
        url = "http://"+ url + "/login.jsp"
        r = requests.get(url).text
        soup = BeautifulSoup(r, 'html.parser')
        tags = soup.select("form input")

        idp = re.compile("id=₩"[a-zA-Z]*id[a-z]*₩"")
        for tag in tags:
            try:
                result = idp.search(str(tag)).group()
                id_value = result.replace("id=", "").replace("₩"", "")

                break
            except AttributeError:
                result = None

        pw_value = soup.select('form input[type=password]')[0]['name']
        submit = soup.select('form input[type=submit]')
        submitp = re.compile("₩"[a-zA-Z]*[L|l]ogin[a-zA-Z]*₩"")
        subnetname = submitp.search(str(submit)).group().replace("₩"", "")
        tags = soup.select("form")
        formp = re.compile("<form action=₩"[a-zA-Z]*[L|l]ogin₩"")
        actionvaluep = re.compile("₩"[a-zA-Z]*₩"")
```

```
        for tag in tags:
            result = formp.search(str(tag))
            if result != None:
                action_value = actionvaluep.search(result.group()).group().replace("₩"", "")

        domain = get_domain(url)
        next_url = domain + "/" + action_value

        data = {id_value:sqlinjection(),
                pw_value:'donecare'}
        contents = str(data)
        send_post(data, next_url)
        is_cve = send_post(data, next_url)
        return (module_name, contents.strip(), is_cve)



pages = set()
def getLinks(pageUrl):
    global pages
    html = urlopen(pageUrl)
    soup = BeautifulSoup(html, "html.parser")
    for link in soup.findAll("a"):
        if 'href' in link.attrs:
            pages.add(link.attrs['href'])
            if link.attrs['href'] not in pages:
                newPage = link.attrs['href']

                pages.add(newPage)
                getLinks(newPage)


def dicxss(url):
    module_name = "XSS"
    contents = ""
    is_cve = "Safe"

    url = "http://"+url
    getLinks(url)
    lst = list(pages)

    dic ={}
    d=0
    for i in lst:
        check = parse.urlparse(lst[int(d)])
        check.geturl()

        if check.query:

            dic.update(parse.parse_qs(check.query))
        d+=1
```

```
    fname = "payloads.txt"
    with open(fname) as f:
        content = f.readlines()
    payloads = [x.strip() for x in content]
    vuln = []
    for payload in payloads:
        for t in dic.keys():
            payload = payload
            xss_url = url+"?"+t+"="+payload
            r = requests.get(xss_url)
            if payload.lower() in r.text.lower():
                if(payload not in vuln):
                    vuln.append(payload)
            else:
                continue
    if vuln:
        tmp_contents = "₩n".join(vuln)
        contents += str(tmp_contents)
        is_cve = "Risk"

    return (module_name, contents.strip(), is_cve)



def get_urldirectorypath(url):

    current_pagep = '₩/[a-zA-Z0-9]*₩.[a-zA-Z0-9]*$'

    path = re.sub(current_pagep, "", url)

    return path



def return_souporhtml(url, str):

    r = requests.get(url).text

    soup = BeautifulSoup(r, 'html.parser')

    if(str=="soup"):

        return soup

    elif(str=="html"):

        return soup.text



def regex_search(regex, str):

    p = re.compile(regex)
```

```
    s = p.search(str)
    return s

def dicrec(url):
    module_name = "Directory Listing"
    contents = ""
    is_cve = "Safe"
    c=0
    url = "http://"+url
    getLinks(url)
    lst = list(pages)
    for i in lst:
        toryurl = url+"/"+lst[int(c)]
        path = get_urldirectorypath(toryurl)
        html = return_souporhtml(path, "html")

        s = regex_search('Index of /', html)

        if s == None:
            contents = "This website is ₩"SAFE₩" from Directory listing"

        else:
            contents = path
            is_cve = "Risk"
            c+=1
    return (module_name, contents.strip(), is_cve)


start = datetime.datetime.now()

def Westall(domain):
    results = []

    results.append(scan(domain))
    results.append(adpage(domain))
    get_header(domain)
    results.append(check_cve(get_header))
    results.append(dicrec(domain))
    results.append(dicxss(domain))
    results.append(sqltest(domain))
    return results
data = []
url = str(sys.argv[1])
uid = str(sys.argv[2])
data.extend(Westall(url))

filename = "cve_log_"+uid+".pdf"
title = 'Web Scan Report'
finish = datetime.datetime.now()
duration = finish - start

class PDF(FPDF):
```

```
def header(self):
        self.set_font("Arial", size=24)
        self.cell(200, 20, txt="Website Vulnerability Scanner Report", ln=1, align="C")
        self.set_text_color(46,138,204)
        self.cell(20, 20, txt="Scan Information", ln=1)
        self.set_line_width(1)
        self.set_draw_color(255, 0, 0)
        self.line(10, 45, 200, 45)
        self.set_line_width(1)
        self.set_draw_color(0, 0, 0)
        self.set_text_color(0,0,0)
        self.set_font("Arial", size=11)
        self.cell(20, 5, txt="Website URL = "+url, ln=1)
        self.cell(10,5,txt="Start Time = "+str(start), ln=1)
        self.cell(10,5,txt="Finish Time = "+str(finish), ln=1)
        self.cell(10,5,txt="Scan duration = "+str(duration), ln=1)

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):


        global data

        self.cell(10, 20, ln=1, align="c")
        self.set_font("Arial", 'B', size=24)
        self.cell(10, 20, txt="List of tests performed (6/6)",ln=1 ,align="L")
        self.set_font("Arial", 'B', size=15)
        self.set_draw_color(0, 0, 0)
        self.set_line_width(0.5)
        col_width = self.w / 3.3
        row_height = self.font_size
        header = ('Type', 'Contents', 'Resulte')

        cellwidth =110
        cellHeight= 5
        self.cell(40, cellHeight*3, txt=header[0], border=1, align="C")
        self.cell(cellwidth, cellHeight*3, txt=header[1], border=1, align="C")
        self.cell(40, cellHeight*3, txt=header[2], border=1, ln=1, align="C")
        self.set_font("Arial", size=10)

        for i in range(6):
```

```
                        line =1
                        if self.get_string_width(data[i][1]) < cellwidth:
                                line =1

                        else:

                                textLength=len(data[i][1])
                                errMargin = 44
                                startChar = 0
                                maxChar = 0
                                textArray = []
                                tmpString=""
                                st = data[i][1]
                                while (startChar < textLength):
                                        while (self.get_string_width(tmpString) < (cellwidth -
errMargin) and (startChar+maxChar) < textLength):
                                                maxChar +=1
                                                tmpString = st[startChar : maxChar]
                                        startChar= startChar + maxChar
                                        textArray.append(tmpString)
                                        line +=1
                                        maxChar=0
                                        tmpString=""
                        self.cell(40, line*cellHeight, txt=data[i][0], border=1, ln=0,align = "C")
                        x = self.get_x()
                        y = self.get_y()
                        self.multi_cell(cellwidth, cellHeight, txt=data[i][1], border=1)
                        self.set_xy(x+cellwidth, y)
                        self.cell(40, line*cellHeight, txt=data[i][2], border=1, ln=1,align = "C")

    def print_chapter(self, num, title, name):
                self.add_page()
                self.chapter_title(num, title)
                self.chapter_body(name)

pdf = PDF()
pdf.set_title(title)
pdf.add_page()
pdf.chapter_body()
pdf.output(filename, 'F')
```