|  |
| --- |
| **Confidential** |

**웹 모의해킹**

**진단 보고서**

* Coremall.com -

**2020. 08. 18.**

개 정 이 력

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **버전** | **작성일** | **변경내용** | **작성자** | **비고** |
| 1.0 | 2020. 08. |  | 정세영 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목차

[1 개요 7](#_Toc48837366)

[**1.1** **목적** 7](#_Toc48837368)

[**1.2** **진단 방법** 7](#_Toc48837369)

[**1.3** **진단 일정** 7](#_Toc48837370)

[**1.4** **진단 담당자** 7](#_Toc48837371)

[**1.5** **진단대상** 7](#_Toc48837372)

[**1.6** **진단항목** 8](#_Toc48837373)

[2 총평 10](#_Toc48837381)

[**2.1** **총평 요약** 10](#_Toc48837384)

[3 상세 진단 결과 12](#_Toc48837385)

[**3.1** **SQL 인젝션** 12](#_Toc48837389)

[**3.1.1** **취약점 정의** 12](#_Toc48837390)

[**3.1.2** **상세분석** 12](#_Toc48837391)

[**3.1.3** **보안대책** 13](#_Toc48837392)

[**3.2** **디렉터리 인덱싱** 14](#_Toc48837393)

[**3.2.1** **취약점 정의** 14](#_Toc48837394)

[**3.2.2** **상세분석** 14](#_Toc48837395)

[**3.2.3** **보안대책** 14](#_Toc48837396)

[**3.3** **정보누출** 15](#_Toc48837397)

[**3.3.1** **취약점 정의** 15](#_Toc48837398)

[**3.3.2** **상세분석** 15](#_Toc48837399)

[**3.3.3** **보안대책** 16](#_Toc48837400)

[**3.4** **악성 콘텐츠** 17](#_Toc48837401)

[**3.4.1** **취약점 정의** 17](#_Toc48837402)

[**3.4.2** **상세분석** 17](#_Toc48837403)

[**3.4.3** **보안대책** 18](#_Toc48837404)

[**3.5** **크로스 사이트 스크립팅** 19](#_Toc48837405)

[**3.5.1** **취약점 정의** 19](#_Toc48837406)

[**3.5.2** **상세분석** 19](#_Toc48837407)

[**3.5.3** **보안대책** 20](#_Toc48837408)

[**3.6** **불충분한 인증** 21](#_Toc48837409)

[**3.6.1** **취약점 정의** 21](#_Toc48837410)

[**3.6.2** **상세분석** 21](#_Toc48837411)

[**3.6.3** **보안대책** 22](#_Toc48837412)

[**3.7** **취약한 패스워드 복구** 23](#_Toc48837413)

[**3.7.1** **취약점 정의** 23](#_Toc48837414)

[**3.7.2** **상세분석** 23](#_Toc48837415)

[**3.7.3** **보안대책** 23](#_Toc48837416)

[**3.8** **크로스사이트 리퀘스트 변조(CSRF)** 24](#_Toc48837417)

[**3.8.1** **취약점 정의** 24](#_Toc48837418)

[**3.8.2** **상세분석** 24](#_Toc48837419)

[**3.8.3** **보안대책** 25](#_Toc48837420)

[**3.9** **불충분한 인가** 26](#_Toc48837421)

[**3.9.1** **취약점 정의** 26](#_Toc48837422)

[**3.9.2** **상세분석** 26](#_Toc48837423)

[**3.9.3** **보안대책** 27](#_Toc48837424)

[**3.10** **불충분한 세션 만료** 28](#_Toc48837425)

[**3.10.1** **취약점 정의** 28](#_Toc48837426)

[**3.10.2** **상세분석** 28](#_Toc48837427)

[**3.10.3** **보안대책** 28](#_Toc48837428)

[**3.11** **세션 고정** 29](#_Toc48837429)

[**3.11.1** **취약점 정의** 29](#_Toc48837430)

[**3.11.2** **상세분석** 29](#_Toc48837431)

[**3.11.3** **보안대책** 29](#_Toc48837432)

[**3.12** **자동화 공격** 30](#_Toc48837433)

[**3.12.1** **취약점 정의** 30](#_Toc48837434)

[**3.12.2** **상세분석** 30](#_Toc48837435)

[**3.12.3** **보안대책** 31](#_Toc48837436)

[**3.13** **파일 업로드** 32](#_Toc48837437)

[**3.13.1** **취약점 정의** 32](#_Toc48837438)

[**3.13.2** **상세분석** 32](#_Toc48837439)

[**3.13.3** **보안대책** 33](#_Toc48837440)

[**3.14** **파일 다운로드** 34](#_Toc48837441)

[**3.14.1** **취약점 정의** 34](#_Toc48837442)

[**3.14.2** **상세분석** 34](#_Toc48837443)

[**3.14.3** **보안대책** 35](#_Toc48837444)

[**3.15** **관리자 페이지 노출** 36](#_Toc48837445)

[**3.15.1** **취약점 정의** 36](#_Toc48837446)

[**3.15.2** **상세분석** 36](#_Toc48837447)

[**3.15.3** **보안대책** 37](#_Toc48837448)

[**3.16** **데이터 평문 통신** 38](#_Toc48837449)

[**3.16.1** **취약점 정의** 38](#_Toc48837450)

[**3.16.2** **상세분석** 38](#_Toc48837451)

[**3.16.3** **보안대책** 38](#_Toc48837452)

**표 목차**

[<표1- 1> 진단일정 8](#_Toc461106828)

[<표1- 2> 수행인원 8](#_Toc461106829)

[<표1- 3> WEB 진단대상 8](#_Toc461106830)

[<표1- 4> 웹 어플리케이션 소스 보안 진단 체크리스트 9](#_Toc461106831)

[<표2- 5> 취약점 진단 결과 10](#_Toc461106832)

**그림 목차**

[[그림 3-1] SQL 인젝션 공격으로 관리자 권한 획득 12](#_Toc48896295)

[[그림 3-2] SQL 인젝션 공격으로 다른 사용자 권한 획득 12](#_Toc48896296)

[[그림 3-3] icons 디렉터리 내 구조 확인 14](#_Toc48896297)

[[그림 3-4] 잘못된 정보를 입력했을 때 DB 정보가 누출 된 모습 15](#_Toc48896298)

[[그림 3-5] 주민등록번호 뒷자리 마스킹 부재 15](#_Toc48896299)

[[그림 3-6] 홈페이지 구성하는 확장자명 노출 16](#_Toc48896300)

[[그림 3-7] 스크립트 내에 불필요한 정보 삽입 16](#_Toc48896301)

[[그림 3-8] 게시글에 악성 스크립트 삽입 17](#_Toc48896302)

[[그림 3-9] 악성 스크립트 삽입된 게시글 클릭 시 이동 18](#_Toc48896303)

[[그림 3-10] Stored XSS 공격 결과 19](#_Toc48896304)

[[그림 3-11] 회원정보 수정에 추가적인 인증 절차 없이 접근 21](#_Toc48896305)

[[그림 3-12] 패스워드 찾기 화면 23](#_Toc48896306)

[[그림 3-13] 취약한 임시 패스워드, 패스워드 홈페이지 노출 23](#_Toc48896307)

[[그림 3-14] 게시판에 자동으로 글 생성되는 스크립트 삽입 24](#_Toc48896308)

[[그림 3-15] 해당 게시글을 읽을 경우 자동 생성되는 게시글 25](#_Toc48896309)

[[그림 3-16] 비인가 사용자의 공지사항 게시 시도 26](#_Toc48896310)

[[그림 3-17] 비인가 사용자의 공지사항 게시 27](#_Toc48896311)

[[그림 3-18] 로그인 세션이 만료되지 않은 상태 28](#_Toc48896312)

[[그림 3-19] 세션 ID가 유지되는 상황 29](#_Toc48896313)

[[그림 3-20] Burp suite를 통한 repeater 기능 활용 30](#_Toc48896314)

[[그림 3-21] 게시판 내 자동생성된 게시글 31](#_Toc48896315)

[[그림 3-22] PHP파일로 된 웹 쉘 업로드 시도 32](#_Toc48896316)

[[그림 3-23] Content-Type과 Filename 변경 32](#_Toc48896317)

[[그림 3-24] PHP 파일 업로드 33](#_Toc48896318)

[[그림 3-25] 업로드된 웹쉘 실행 33](#_Toc48896319)

[[그림 3-26] PHP Webshell을 이용하여 boot.ini 다운로드 시도 34](#_Toc48896320)

[[그림 3-27] 다운로드된 boot.ini 파일 35](#_Toc48896321)

[[그림 3-28] 관리자 로그인 페이지 노출과 관리자 페이지 접근 36](#_Toc48896322)

[[그림 3-29] 데이터 평문 통신으로 ID, PASSWORD 노출 38](#_Toc48896323)

1 개요

1. 1. **목적**

* 운영 중인 코어몰 사이트의 보안진단을 통해 고객사의 개인정보 유출 및 보안 사고 방지 등 보안 강화에 그 목적
  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 | 2018. 08. 18. |
| 기술적 점검 | 취약점 점검 수행 | 2018. 08. 18. - 2018. 08. 21. |
| 결과분석 / 보고 | 취약점 결과보고서 및 보안대책 작성 | 2018. 08. 21. |
| 보안진단결과보고서 제출 | 2018. 08. 21. |

<표1- 1> 진단일정

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

|  |  |  |
| --- | --- | --- |
| **수행인원** | **수행업무** | **연락처** |
| 정세영 | 취약점 진단 및 보고서 작성 | 010-0000-0000 |

<표1- 2> 수행인원

* 1. **진단대상**

|  |  |  |  |
| --- | --- | --- | --- |
| **구분** | **IP** | **WEB** | **비고** |
| 1 | 70.12.113.49 | coremall.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**

코어몰 사이트 진단 결과 16개의 취약점이 발견 되었으며, SQL인젝션, 디렉터리 인덱싱, 정보누출, 악성 콘텐츠, 크로스 사이트 스크립팅, 불충분한 인증, 취약한 패스워드 복구, 크로스사이트 리퀘스트 변조, 불충분한 인가, 불충분한 세션만료, 세션고정, 자동화공격, 파일업로드, 파일 다운로드, 관리자 페이지 노출, 데이터 평문 전송 등 취약점이 발견됨. 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 | **취약** |
| 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 | **취약** |
| 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 | **양호** |
| 22 | FU | 파일 업로드 | 상 | **취약** |
| 23 | FD | 파일 다운로드 | 상 | **취약** |
| 24 | AE | 관리자 페이지 노출 | 상 | **취약** |
| 25 | PT | 경로 추적 | 상 | **양호** |
| 26 | PL | 위치 공개 | 상 | **양호** |
| 27 | SN | 데이터 평문 전송 | 상 | **취약** |
| 28 | CC | 쿠키 변조 | 상 | **양호** |

<표2- 5> 취약점 진단 결과

3 상세 진단 결과

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 방법임.

* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | SQL 인젝션 공격에 취약한 파라미터가 존재하며, 이 취약점을 통해 DB의 중요한 정보(관리자 권한)를 공격자가 탈취 가능함. |

**[Step #1]**



[그림 3-] SQL 인젝션 공격으로 관리자 권한 획득



[그림 3-] SQL 인젝션 공격으로 다른 사용자 권한 획득

* + 1. **보안대책**

- 사용자 입력 값 중 DB 쿼리에 사용될 수 있는 ‘, “, -와 같은 문자의 필터링을 권고함. 또한 아래코드 형태의 BlackList 필터링 기법이 아닌 허용된 문자만 입력 받는 WhiteList방식으로 필터링 될 것을 권고함.

**[ 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") ]**

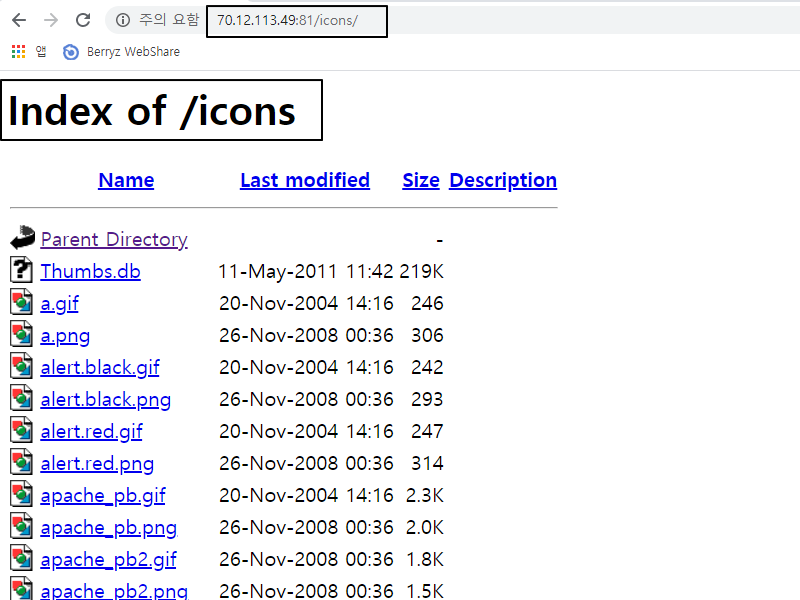
* 1. **디렉터리 인덱싱**
     1. **취약점 정의**

해당 취약점이 존재할 경우 브라우저를 통해 특정 디렉터리 내 파일 리스트를 노출하여 응용시스템의 구조를 외부에 허용할 수 있고, 민감한 정보가 포함된 설정 파일 등이 노출될 경우 보안상 심각한 위험을 초래할 수 있음.

* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | /icons/을 URL에 입력할 경우 디렉터리 내용이 출력됨을 볼 수 있음. |

**[Step #1]**



[그림 3-] icons 디렉터리 내 구조 확인

* + 1. **보안대책**

- 웹 서버 환경 설정에서 디렉터리 인덱싱 기능 제거.

- 설정 > 제어판 > 관리도구 > “인터넷 서비스 관리자” 선택 후 해당 웹 사이트에서 우클릭 후 등록정보 > [홈 디렉터리] 탭 > [디렉터리 검색] 체크 해제

* 1. **정보누출**
     1. **취약점 정의**

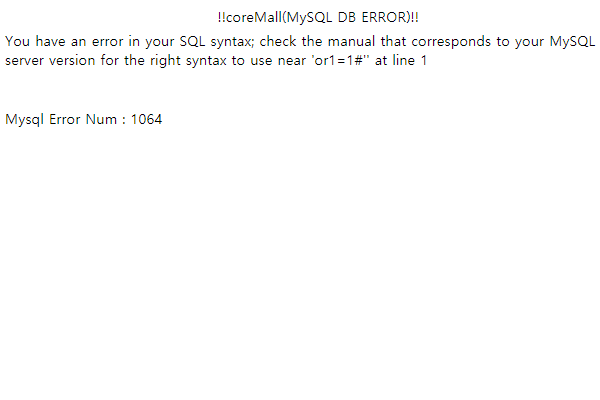
웹 어플리케이션의 민감한 정보가 개발자의 부주의로 인해 노출되는 것으로 중요 정보(관리자 계정 및 테스트 계정 등)를 주석구문에 포함시켜 의도하지 않게 정보가 노출되는 취약점임.

또한 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**

|  |  |
| --- | --- |
| 설명 | DB에러페이지에서 DB정보 노출, 주민등록 번호 마스킹 부재, 홈페이지 구성 확장자명 노출되는 등 해당 홈페이지 사용자가 접근하면 안되는 정보들이 누출됨. 보고된 자료 외에도 많은 정보 누출이 있음. |

**[Step #1]**

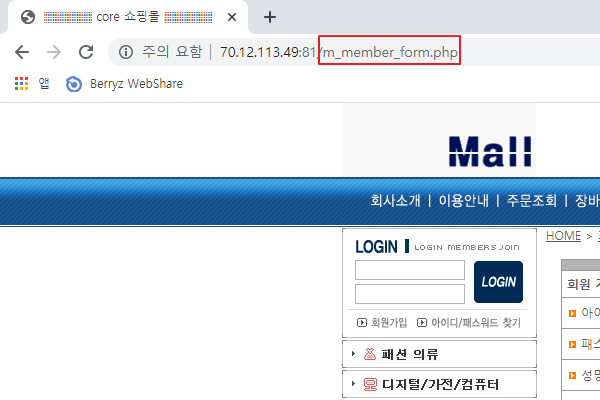


[그림 3-] 잘못된 정보를 입력했을 때 DB 정보가 누출 된 모습

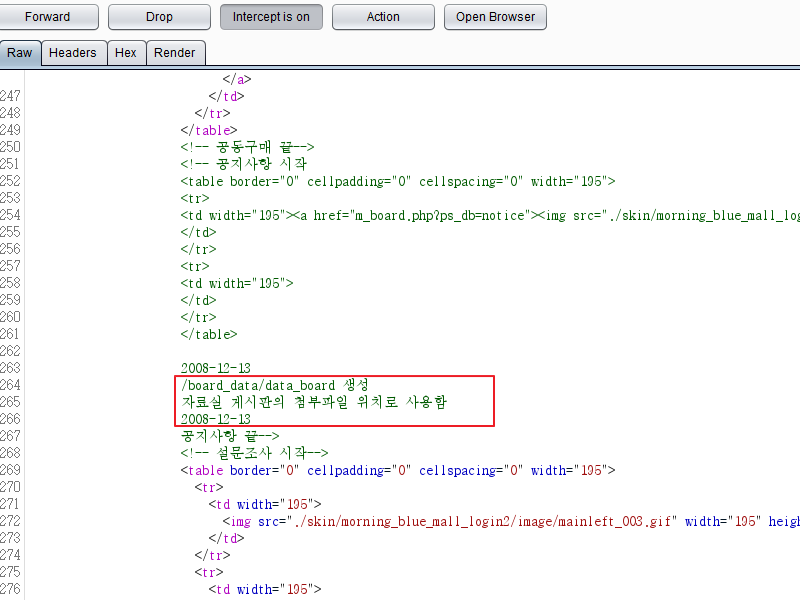
**[Step #2]**



[그림 3-] 주민등록번호 뒷자리 마스킹 부재



[그림 3-] 홈페이지 구성하는 확장자명 노출



[그림 3-] 스크립트 내에 불필요한 정보 삽입

* + 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 설정함.

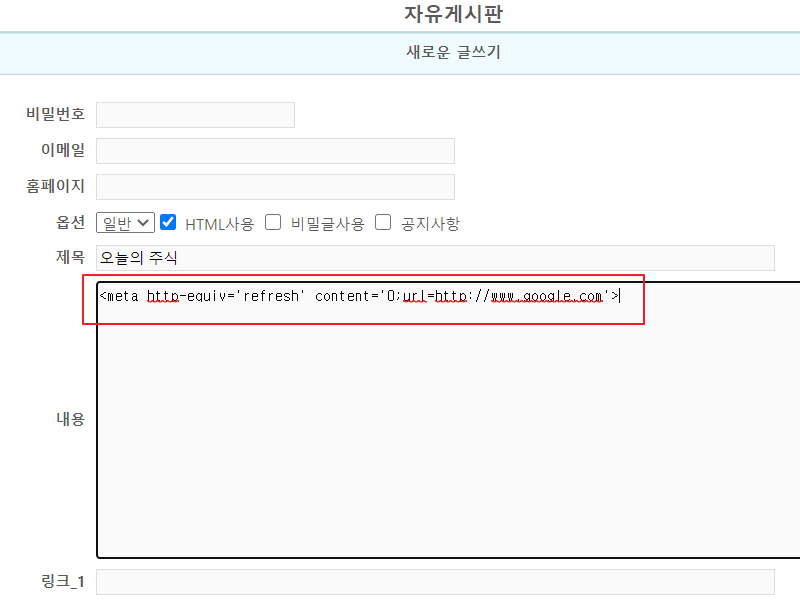
* 1. **악성 콘텐츠**
     1. **취약점 정의**

웹 사이트 게시판, 댓글, 자료실 등에 정상적인 콘텐츠 대신에 악성 콘텐츠를 주입하여 실행될 경우 사용자가 원본 콘텐츠 대신 악성코드 감염 및 웹 페이지 변조 등 사용자에게 악의적인 영향을 미칠 수 있는 악성 콘텐츠를 열람할 수 있음.

* + 1. **상세분석**

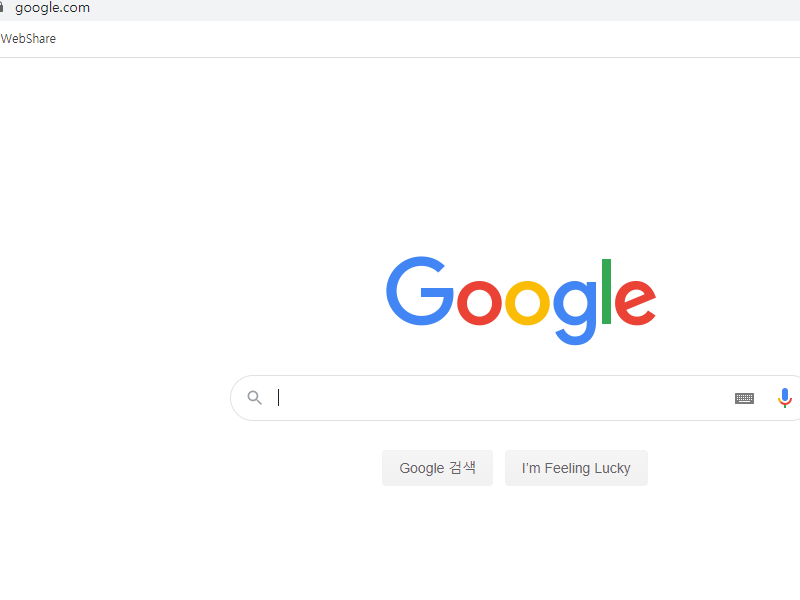
|  |  |
| --- | --- |
| 설명 | 해당 홈페이지는 게시판 내에 HTML을 허용하여 게시글에 악성스크립트를 삽입해 악성 홈페이지로 자동으로 이동하게 만들 수 있음.. |

**[Step #1]**



[그림 3-] 게시글에 악성 스크립트 삽입





[그림 3-] 악성 스크립트 삽입된 게시글 클릭 시 이동

* + 1. **보안대책**

- 콘텐츠 삽입 및 파일 업로드 제한 필터링 적용 여부 점검.

- 게시판 등의 페이지에서 강제적으로 이뤄지는 악의적인 프로그램 다운로드 및 콘텐츠 자동 실행이나 악의적인 사이트로의 이동이 발생하는지 확인

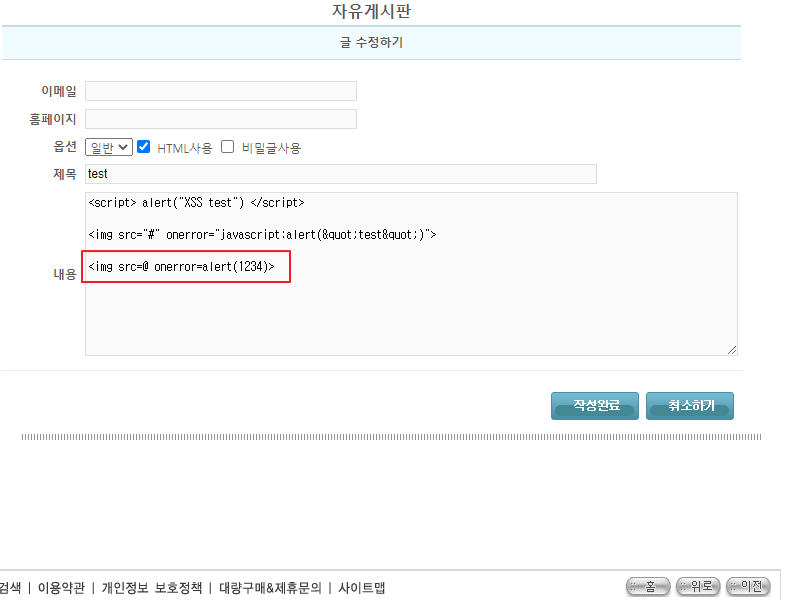
* 1. **크로스 사이트 스크립팅**
     1. **취약점 정의**

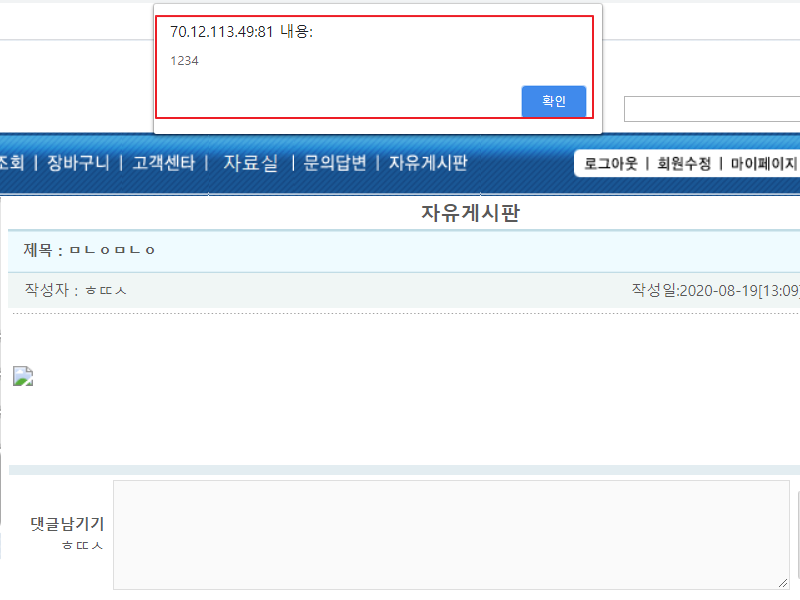
웹 어플리케이션에서 사용자 입력 인수 값에 대한 필터링이 제대로 이루어지지 않을 경우, 사용자 인수 값을 받는 웹 사이트 게시판, URL 등에 악의적인 스크립트(자바스크립트, VB스크립트, ActiveX, 플래시 등)를 삽입하여 게시글이나 이메일을 읽는 사용자의 쿠키(세션)를 도용하거나 악성코드(URL 리다이렉트)를 유포할 수 있음.

* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | 게시판 작성시 HTML로 작성이 가능하여 내용에 스크립트 작성 시 해당 스크립트가 실행되는 상황임. |

**[Step #1]**





[그림 3-] Stored XSS 공격 결과

* + 1. **보안대책**
* 게시물에 HTML이나 자바 스크립트에 해당되는 태그 사용을 사전에 제한하고, 사용자가 입력한 인수 값에 대한 필터링 작업 필요.
* 게시물의 본문 뿐만 아니라 제목, 댓글, 검색어 입력 창, 그 외 사용자 측에서 넘어오는 값을 신뢰하는 모든 form과 인수 값에 대해서 필터링을 수행함.
* 웹 방화벽에 모든 사용자 입력 폼을 대상으로 특수문자, 특수 구문 필터링하도록 룰셋 적용.
  1. **불충분한 인증**
     1. **취약점 정의**

중요 정보(회원정보 등) 페이지에 대한 인증 절차가 불충분할 경우 발생하는 취약점으로 권한이 없는 사용자가 중요 정보 페이지에 접근하여 정보를 유출하거나 변조할 수 있으므로 중요 정보 페이지에는 추가적인 인증 절차를 구현하여야 함.

* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | 로그인 후 회원정보를 수정하는 페이지에 접근하는데 추가적인 인증 절차가 없음. 해당 페이지 접근 시에 추가적인 인증 절차가 필요함. |

**[Step #1]**



[그림 3-] 회원정보 수정에 추가적인 인증 절차 없이 접근

* + 1. **보안대책**

- 중요 정보 페이지와 같은 중요 정보를 표시하는 페이지에서는 본인 인증을 재확인하는 로직을 구현하고, 인증 후 사용자가 이용가능 페이지에 접근할 때마다 승인을 얻은 사용자인지 페이지마다 검증을 하여야 함.

- 접근 통제 정책을 구현하고 있는 코드는 구조화, 모듈화가 되어 있어야 함.

- 인증 과정을 처리하는 부분에 Client Side Script(JS, VBScript등)를 사용하면 사용자가 임의로 수정할 수 있으므로 Server Side Script(JSP, ASP, PHP 등)를 통하여 인증 및 필터링 과정을 수행함.

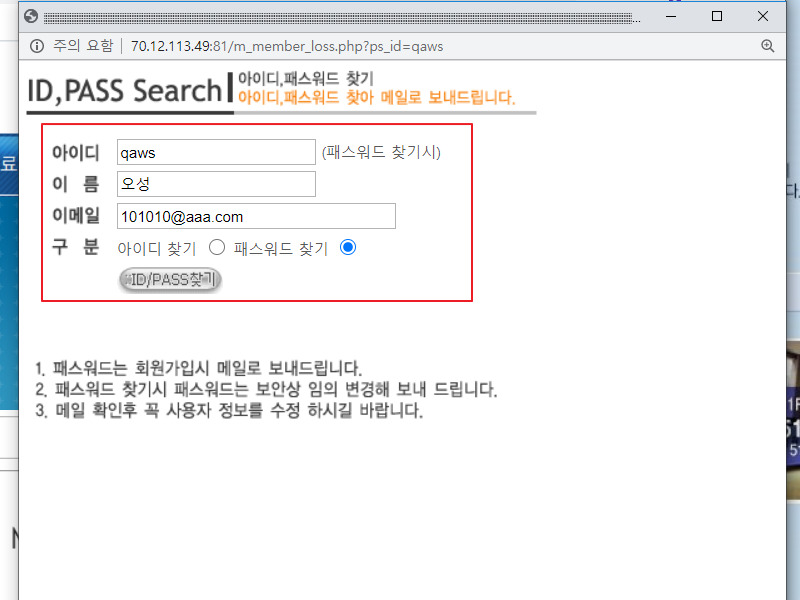
* 1. **취약한 패스워드 복구**
     1. **취약점 정의**

취약한 패스워드 복구 로직(패스워드 찾기 등)으로 인하여 공격자가 불법적으로 다른 사용자의 패스워드를 획득, 변경할 수 있음.

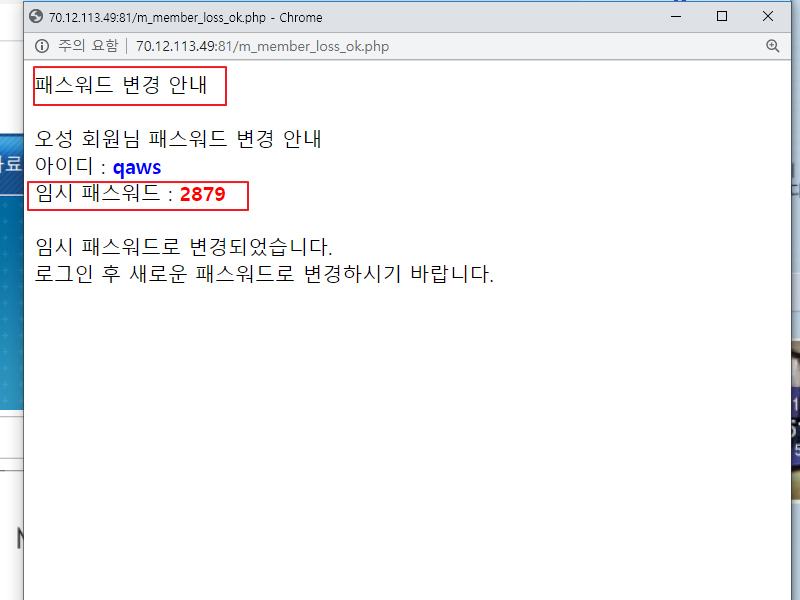
* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | 패스워드 찾기를 할 경우, 임시 패스워드가 웹페이지에 공개되고 취약한 패스워드로 보여짐. |

**[Step #1]**



[그림 3-] 패스워드 찾기 화면



[그림 3-] 취약한 임시 패스워드, 패스워드 홈페이지 노출

* + 1. **보안대책**

- 사용자 패스워드를 발급해주거나 확인해줄 때, 웹사이트 화면에 바로 출력하는 것이 아니라 인증된 사용자 메일이나 SMS로 전송해주어야 함.

- 난수를 이용한 불규칙적이고 최소 길이(6자 이상 권고) 이상의 패턴이 없는 패스워드 발급하여야 함.

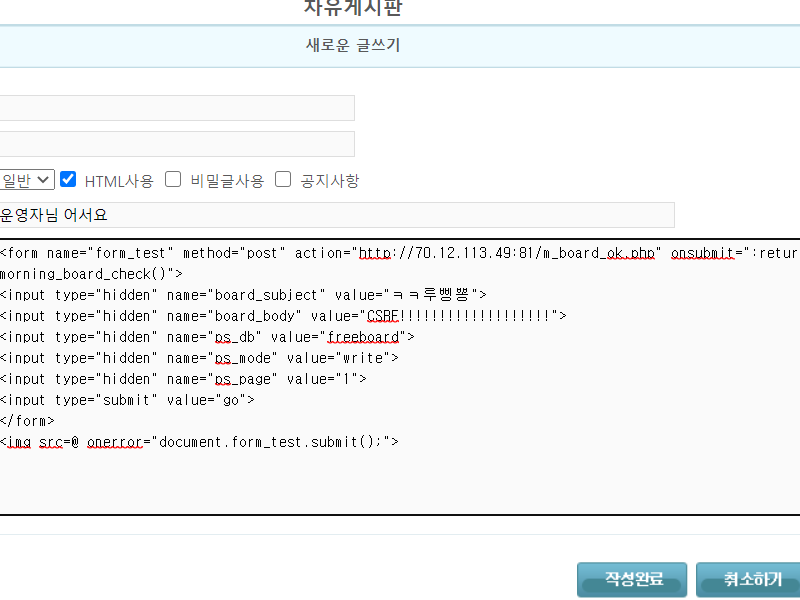
* 1. **크로스사이트 리퀘스트 변조(CSRF)**
     1. **취약점 정의**

사용자의 신뢰 정보 내에서 사용자의 요청을 변조함으로써 해당 사용자의 권한으로 악의적인 공격을 수행할 수 있음.

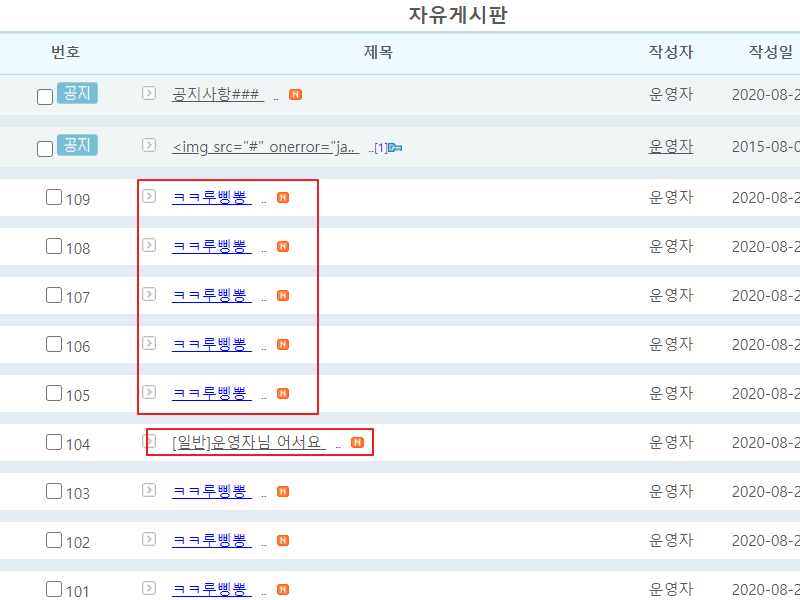
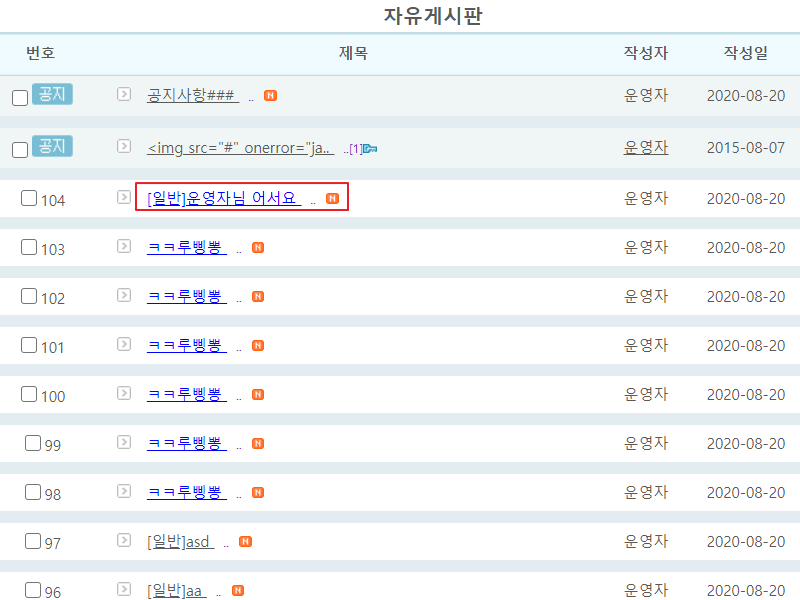
* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | HTML을 지원하는 게시글로 인해 스크립트를 삽입해 해당 게시글을 읽을 경우 자동적으로 게시판에 글쓰기가 되는 상황임. |

**[Step #1]**



[그림 3-14] 게시판에 자동으로 글 생성되는 스크립트 삽입



[그림 3-15] 해당 게시글을 읽을 경우 자동 생성되는 게시글

* + 1. **보안대책**

- 사용자가 입력하는 값에 대한 검증 로직 구현.

- 정상적인 리퀘스트와 비정상적인 리퀘스트를 구분할 수 있도록 Form/URL에서 임의의 토큰을 추가하고 이 토큰을 검증하도록 설계

- HTML이나 자바스크립트에 해당되는 태그 사용을 사전에 제한하고, 서버 단에서 사용자 입력 값에 대한 필터링 구현

- XSS 조치 방안 참조

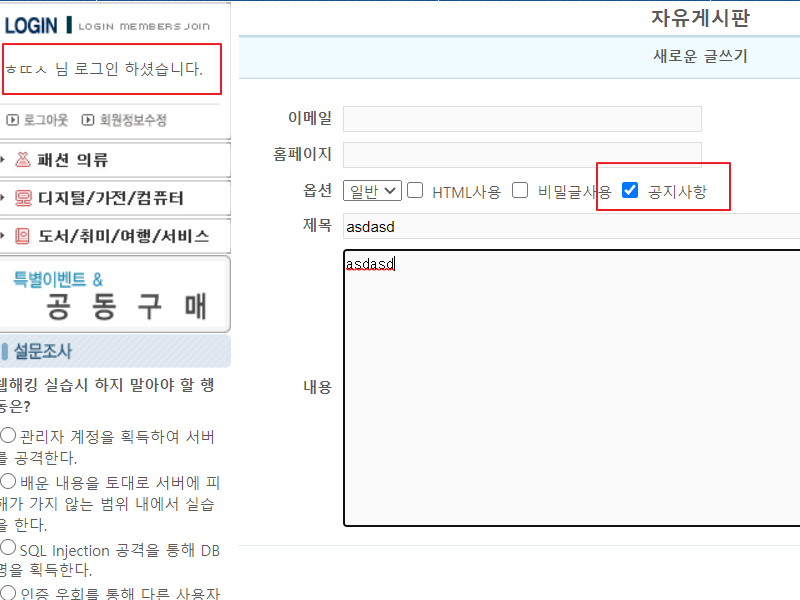
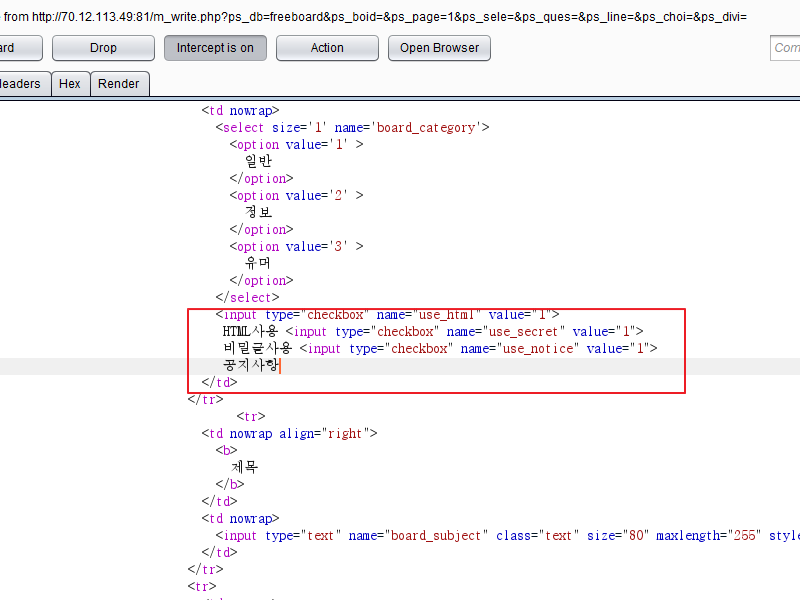
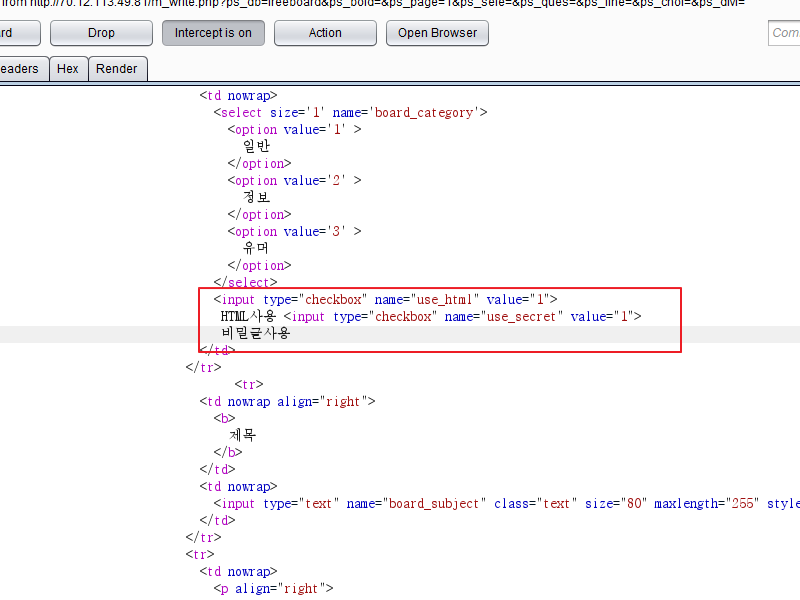
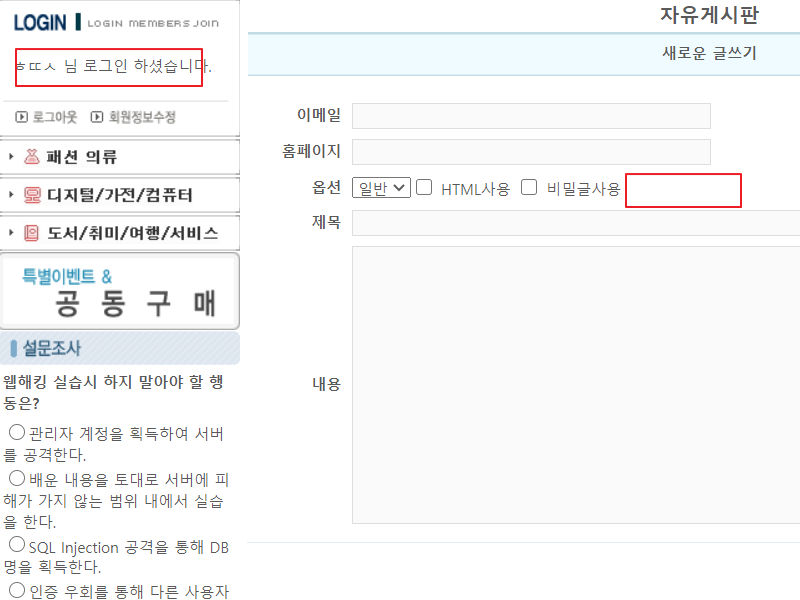
* 1. **불충분한 인가**
     1. **취약점 정의**

중요정보 페이지 접근을 위한 인증 로직이 구현되지 않을 경우, 비인가 사용자의 페이지에 접근 및 중요 정보의 열람 및 변조가 가능함.

* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | 일반 사용자가 게시판에 공지사항을 게시할 수 있는 권한에 접근할 수 있는 상황임. |

**[Step #1]**



[그림 3-] 비인가 사용자의 공지사항 게시 시도



[그림 3-] 비인가 사용자의 공지사항 게시

* + 1. **보안대책**

- 민감한 중요 데이터에 대한 접근 페이지에서 인증을 위한 로직이 구현되지 않았다면, 세션을 통한 인증 및 인증 값 입력을 통한 인증 절차를 구현하여 페이지에 접근하도록 하여야 함.

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

세션의 만류 기간을 정하지 않거나, 만료기한을 너무 길게 설정한 경우 악의적인 사용자가 만료되지 않은 세션을 활용하여 불법적인 접근이 가능할 수 있음.

* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | 사용하지 않은지 1시간이 넘은 세션의 로그인이 유지되고 있는 상황임. |

**[Step #1]**

****



[그림 3-] 로그인 세션이 만료되지 않은 상태

* + 1. **보안대책**

- 세션 타임아웃 기능을 구현해야 함. 타임아웃 시간은 10분으로 설정할 것을 권고함.

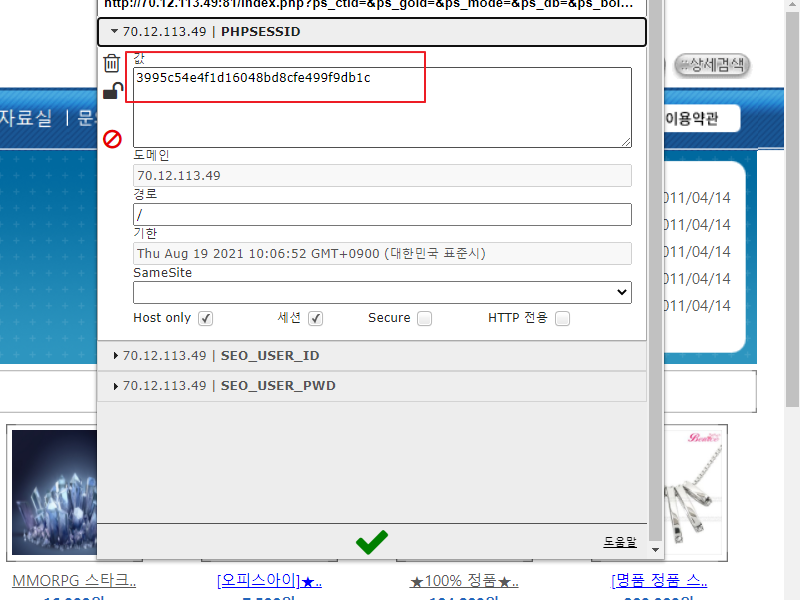
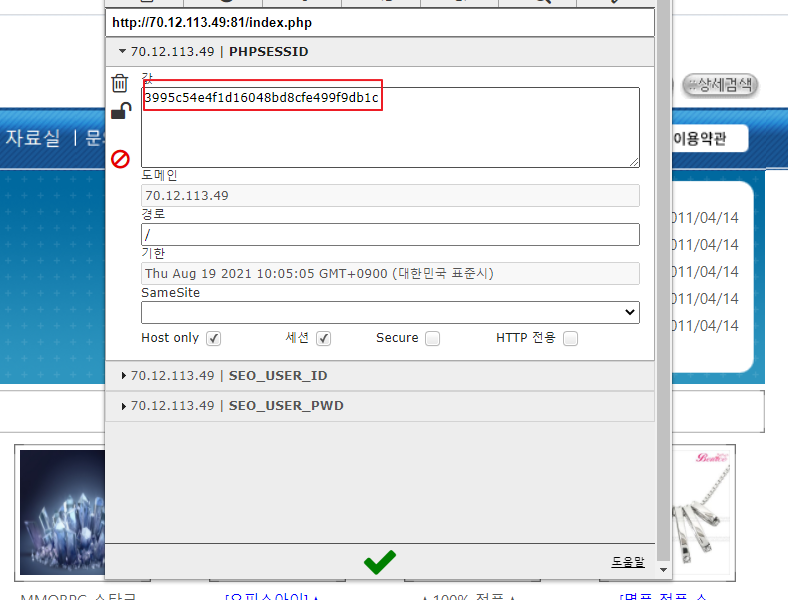
* 1. **세션 고정**
     1. **취약점 정의**

사용자 로그인 시 항상 일정하게 고정된 세션ID가 발행되는 경우 세션 ID를 도용한 비인가자의 접근 및 권한 우회가 가능함.

* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | 5분의 격차를 두고 다시 로그인 했을 때, 해당 세션ID가 유지되는 것을 볼 수 있음. |

**[Step #1]**



[그림 3-] 세션 ID가 유지되는 상황

* + 1. **보안대책**

- 로그인 할 때마다 추측 불가능한 새로운 세션 ID를 발급받도록 해야 하고 기존 세션 ID는 파기해야 함.

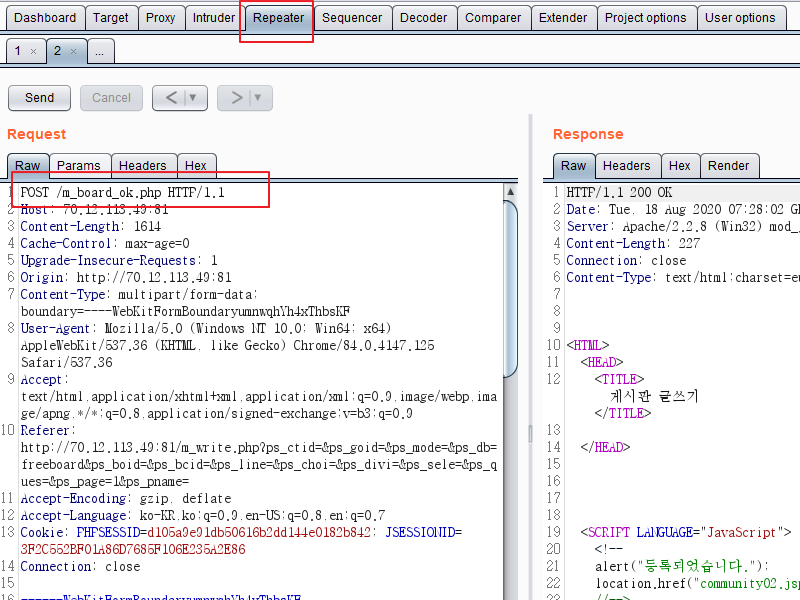
* 1. **자동화 공격**
     1. **취약점 정의**

웹 어플리케이션의 특정 프로세스에 대한 접근 시도 횟수 제한을 설정하지 않고 자동화 공격을 방치하면, 웹사이트를 다운시키거나 무차별 대입 공격으로 인해 사용자 계정을 탈취할 수 있고, 데이터 등록 또는 메일 발송 기능 등을 이용하여 악의적인 활용이 가능함..

* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | 리피터를 통한 자유게시판 게시글 자동 생성이 가능함. 같은 내용의 HTTP POST 패킷에 대한 필터링 없이 무한정 생성됨. |

**[Step #1]**



[그림 3-] Burp suite를 통한 repeater 기능 활용



[그림 3-] 게시판 내 자동생성된 게시글

* + 1. **보안대책**

- 데이터 등록 및 메일 발송 기능에서 사용자 등록이 일회성이 될 수 있도록, 캡차(Captcha) 등 일회성 확인 로직을 구현하여야 함.

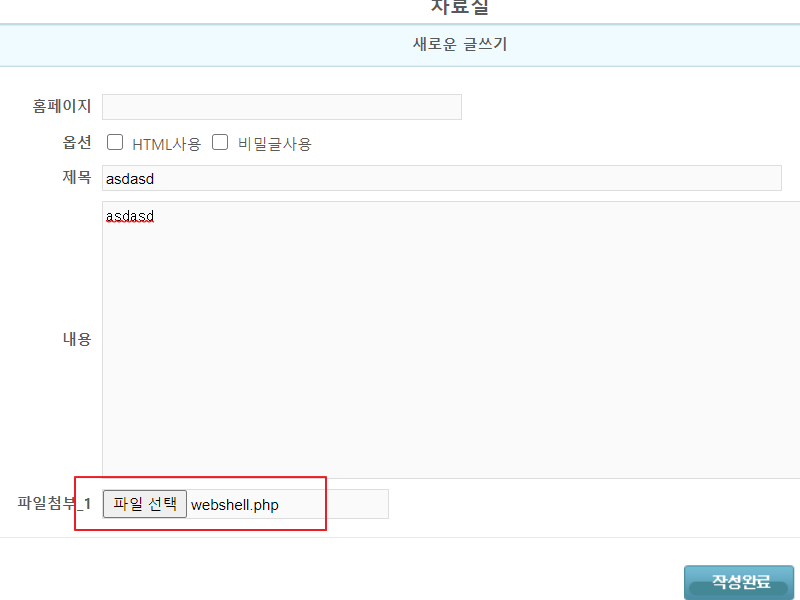
* 1. **파일 업로드**
     1. **취약점 정의**

해당 취약점 존재 시 공격자가 조작된 Server Side Script 파일을 업로드하고 실행하여, 쉘 권한 획득 후 홈페이지를 통해 시스템 명령어를 실행하고, 웹 브라우저를 통해 그 결과 값을 보며, 시스템 관리자 권한 획득 또는 인접 서버에 대한 침입을 시도할 수 있음.

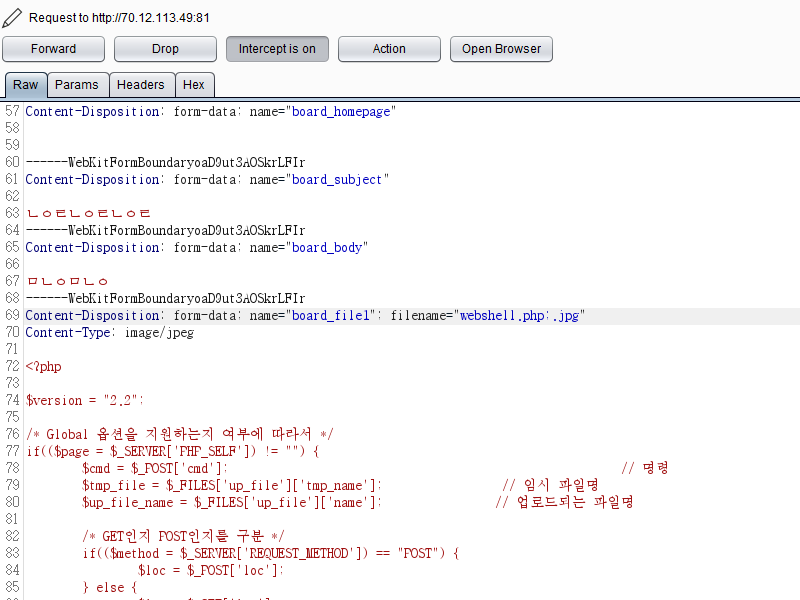
* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | 해당 자료실 게시판은 업로드 가능한 파일을 제한하고 있으나, URL인코딩 우회와 Contents type 우회를 통해 해당 홈페이지에 PHP파일이 업로드 가능함. |

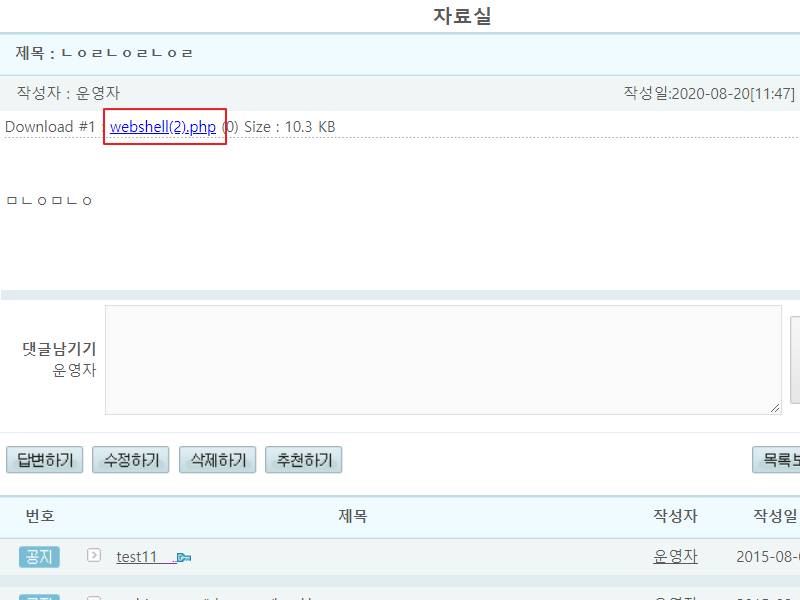
**[Step #1]**



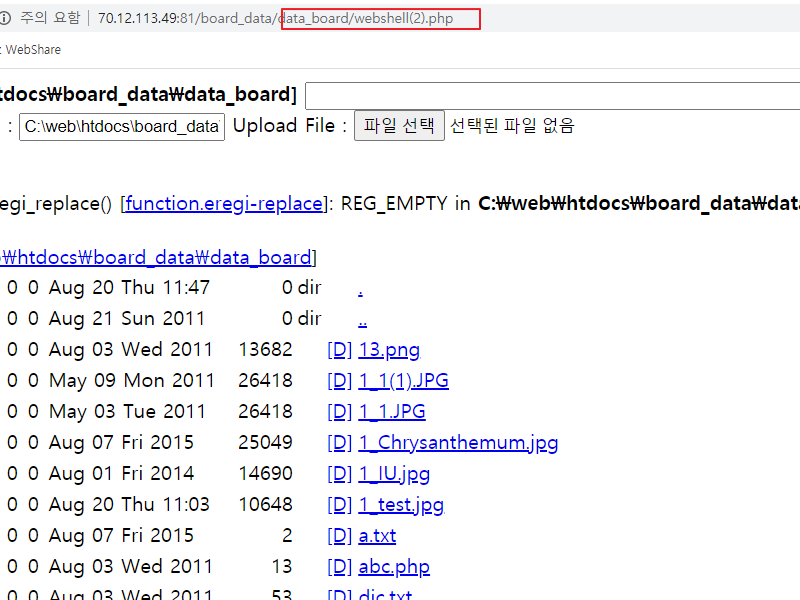
[그림 3-] PHP파일로 된 웹 쉘 업로드 시도



[그림 3-] Content-Type과 Filename 변경



[그림 3-] PHP 파일 업로드



[그림 3-] 업로드된 웹쉘 실행

* + 1. **보안대책**

- 화이트 리스트 방식으로 허용된 확장자만 업로드 허용해야 함.

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

- 업로드 파일을 위한 전용 디렉터리를 별도로 생성하여 httpd.conf와 같은 웹 서버 데몬 설정파일에서 실행 설정을 제거함으로써, Server Side Script가 업로드 되더라도 웹 엔진이 실행하지 않는 환경을 설정해야 함.

- 파일 업로드 필드를 대상으로 특수문자 필터링하도록 웹 방화벽 룰셋을 적용해야 함.

* 1. **파일 다운로드**
     1. **취약점 정의**

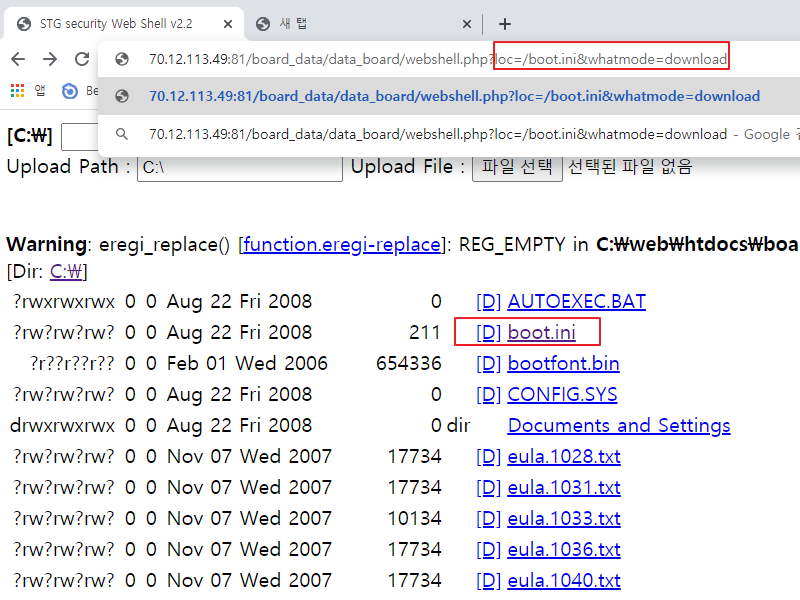
웹 사이트 상에서 파일을 다운받게 해주는 php, jsp, php3 등의 어플리케이션에서 입력되는 인수 값의 유효성을 검증하지 않는 경우 임의의 문자나 주요 파일명의 입력을 통해 웹 서버의 홈 디렉터리를 벗어나서 임의의 위치에 있는 파일을 열람하거나 다운 받는 것이 가능함.

해당 취약점이 존재하는 경우, 공격자가 웹 사이트의 파일 다운로드 관련 어플리케이션의 인수 값을 조작하여 웹 사이트의 중요한 파일(DB커넥션 파일, 어플리케이션 파일 등)이나 웹 사이트를 운용 중인 웹 서버 루트에 있는 중요한 설정 파일(passwd, shadow, boot.ini 등)을 다운로드 할 수 있음..

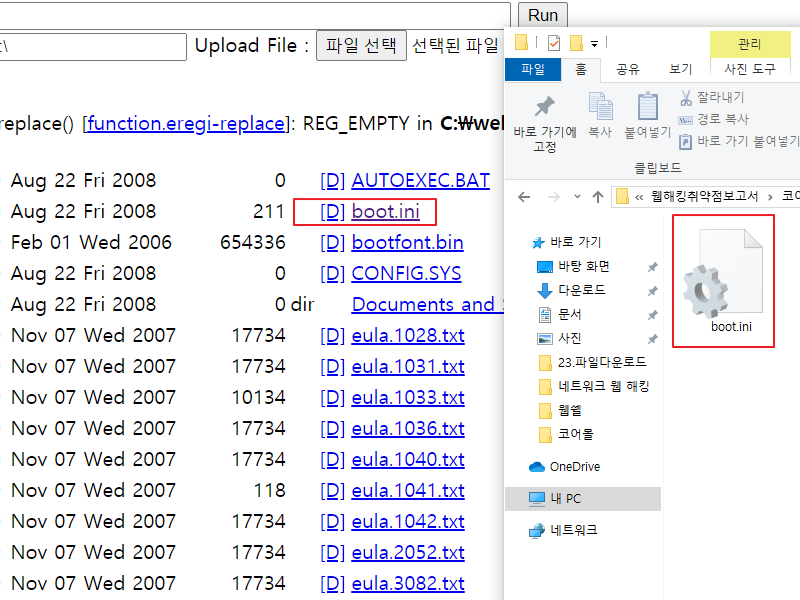
* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | 해당 홈페이지는 업로드된 PHP 웹쉘을 통하여 C:\에 접근이 가능하며 해당 디렉터리에서 boot.ini이나 혹은 다른 중요 정보 파일, 설정 파일을 다운로드 가능함. |

**[Step #1]**



[그림 3-] PHP Webshell을 이용하여 boot.ini 다운로드 시도



[그림 3-] 다운로드된 boot.ini 파일

* + 1. **보안대책**

- 파일 다운로드의 취약성은 주로 파일의 이름을 조작하는데서 비롯되므로 다운로드 파일 이름을 DB에 저장하고 다운로드 수행 시 요청 파일 이름과 비교하여 적절한지 확인하여 사용자가 조작할 수 있는 변수를 제거함.

- PHP를 사용하는 경우 php.ini에서 magic\_quotes\_gpc를 On으로 설정하여 .\./와 같은 역 슬러시 문자 입력 시 치환되도록 설정

- 다운로드 이전에 Webshell과 같은 파일이 업로드 되지 않도록 필터링해야함.

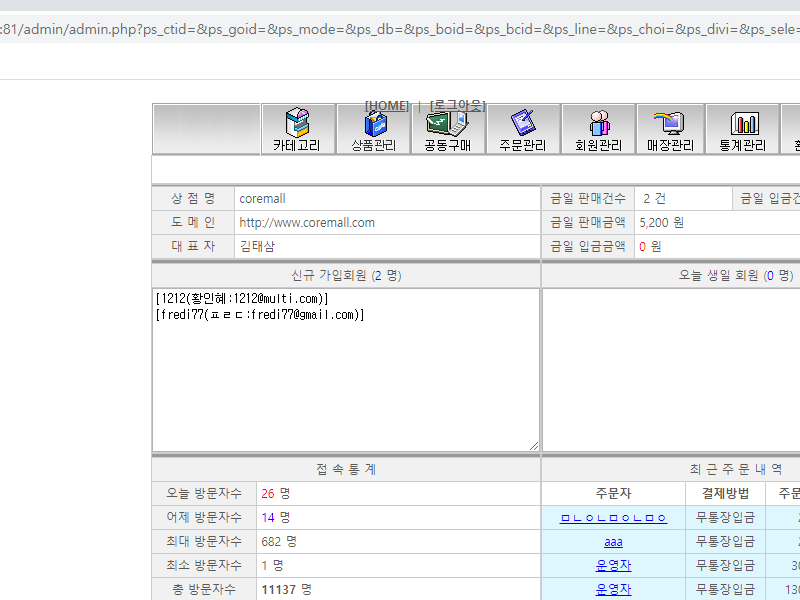
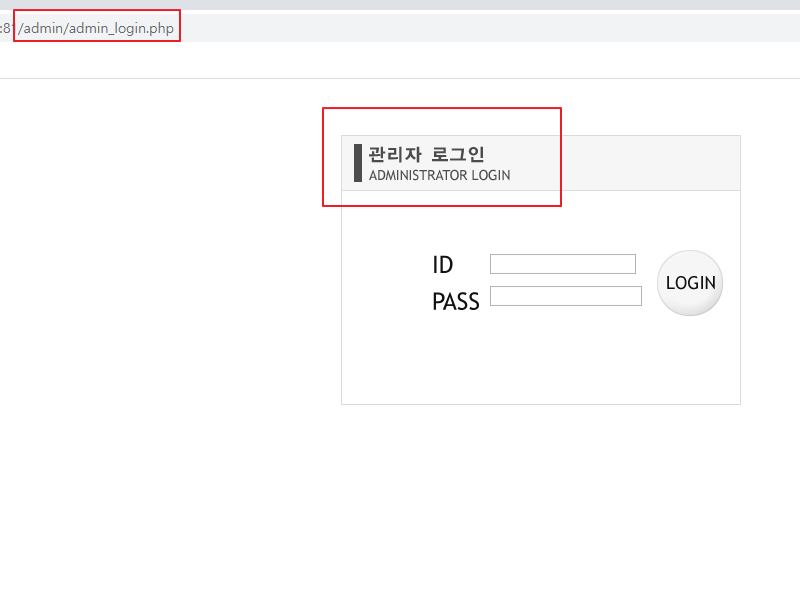
* 1. **관리자 페이지 노출**
     1. **취약점 정의**

웹 관리자의 권한이 노출될 경우 홈페이지의 변조뿐만 아니라 취약성 정도에 따라서 웹 서버의 권한까지도 노출될 수 있음.

* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | 해당 홈페이지는 SQL인젝션을 통해 관리자로 로그인 될 뿐만 아니라 로그아웃할 경우 관리자 홈페이지 로그인 페이지가 나오는 상황임. 관리자 ID의 노출 뿐만 아니라 관리자 페이지의 노출이 쉬운 상황임. |

**[Step #1]**



[그림 3-] 관리자 로그인 페이지 노출과 관리자 페이지 접근

* + 1. **보안대책**

- 일반 사용자의 접근이 불필요한 관리자 로그인 페이지 주소를 유추하기 어려운 이름으로 변경하고 포트도 변경해야 함.

- 관리자 인증을 위한 세션 관리, 웹 방화벽을 이용해 특정 IP만 접근 가능하도록 제한.

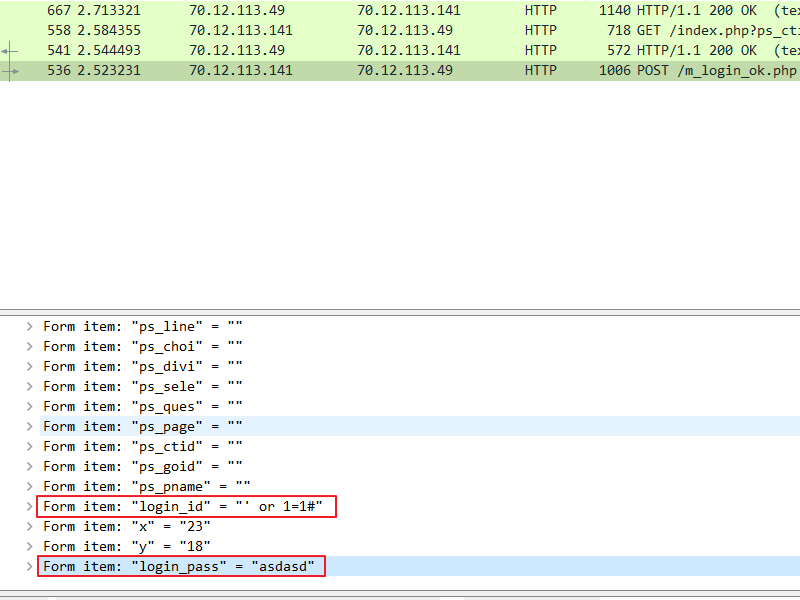
* 1. **데이터 평문 통신**
     1. **취약점 정의**

웹 상의 데이터 통신은 대부분 텍스트 기반으로 이루어지기 때문에 서버와 클라이언트 간에 암호화 프로세스를 구현하지 않으면 간단한 도청을 통해 정보를 탈취 및 도용할 수 있음.

* + 1. **상세분석**

|  |  |
| --- | --- |
| 설명 | 해당 홈페이지는 HTTPS를 지원하지 않기 때문에 Wireshark로 확인한 결과 해당 패킷 내에 로그인 정보가 평문으로 담겨서 데이터 전송이 이루어짐. |

**[Step #1]**



[그림 3-] 데이터 평문 통신으로 ID, PASSWORD 노출

* + 1. **보안대책**

- 웹 상에서의 전송 정보를 제한하여 중요 정보의 전송을 최소화하여야 하며, 중요 정보에 대해서는 반드시 SSL 등의 암호화 통신을 해야 함.

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 |