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

**웹 모의해킹**

**진단 보고서**

Coremall.com

**이현제**

**2020. 08. 21**

개 정 이 력

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 버전 | 작성일 | 변경내용 | 작성자 | 비고 |
| 1.0 | 2020.08.21 | 최초작성 | 이현제 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목차

[1 개요 8](#_Toc48901044)

[**1.1** **목적** 8](#_Toc48901046)

[**1.2** **진단 방법** 8](#_Toc48901047)

[**1.3** **진단 일정** 8](#_Toc48901048)

[**1.4** **진단 담당자** 8](#_Toc48901049)

[**1.5** **진단대상** 8](#_Toc48901050)

[**1.6** **진단항목** 9](#_Toc48901051)

[2 총평 11](#_Toc48901059)

[**2.1** **총평 요약** 11](#_Toc48901062)

[3 상세 진단 결과 13](#_Toc48901063)

[**3.1** **SQL 인젝션** 13](#_Toc48901067)

[**3.1.1** **취약점 정의** 13](#_Toc48901068)

[**3.1.2** **상세분석** 13](#_Toc48901069)

[**3.1.3** **보안대책** 14](#_Toc48901070)

[**3.2** **디렉터리 인덱싱** 15](#_Toc48901071)

[**3.2.1** **취약점 정의** 15](#_Toc48901072)

[**3.2.2** **상세분석** 15](#_Toc48901073)

[**3.2.3** **보안대책** 15](#_Toc48901074)

[**3.3** **정보 누출** 16](#_Toc48901075)

[**3.3.1** **취약점 정의** 16](#_Toc48901076)

[**3.3.2** **상세분석** 16](#_Toc48901077)

[**3.3.3** **보안대책** 18](#_Toc48901078)

[**3.4** **악성 콘텐츠** 18](#_Toc48901079)

[**3.4.1** **취약점 정의** 18](#_Toc48901080)

[**3.4.2** **상세분석** 18](#_Toc48901081)

[**3.4.3** **보안대책** 19](#_Toc48901082)

[**3.5** **크로스사이트 스크립트** 19](#_Toc48901083)

[**3.5.1** **취약점 정의** 19](#_Toc48901084)

[**3.5.2** **상세분석** 19](#_Toc48901085)

[**3.5.3** **보안대책** 21](#_Toc48901086)

[**3.6** **불충분한 인증** 21](#_Toc48901087)

[**3.6.1** **취약점 정의** 21](#_Toc48901088)

[**3.6.2** **상세분석** 21](#_Toc48901089)

[**3.6.3** **보안대책** 23](#_Toc48901090)

[**3.7** **크로스사이트 리퀘스트 변조(CSRF)** 24](#_Toc48901091)

[**3.7.1** **취약점 정의** 24](#_Toc48901092)

[**3.7.2** **상세분석** 24](#_Toc48901093)

[**3.7.3** **보안대책** 24](#_Toc48901094)

[**3.8** **불충분한 인가** 25](#_Toc48901095)

[**3.8.1** **취약점 정의** 25](#_Toc48901096)

[**3.8.2** **상세분석** 25](#_Toc48901097)

[**3.8.3** **보안대책** 26](#_Toc48901098)

[**3.9** **불충분한 세션 만료** 26](#_Toc48901099)

[**3.9.1** **취약점 정의** 26](#_Toc48901100)

[**3.9.2** **상세분석** 26](#_Toc48901101)

[**3.9.3** **보안대책** 27](#_Toc48901102)

[**3.10** **자동화 공격** 27](#_Toc48901103)

[**3.10.1** **취약점 정의** 27](#_Toc48901104)

[**3.10.2** **상세분석** 27](#_Toc48901105)

[**3.10.3** **보안대책** 28](#_Toc48901106)

[**3.11** **파일 업로드 취약점** 28](#_Toc48901107)

[**3.11.1** **취약점 정의** 28](#_Toc48901108)

[**3.11.2** **상세분석** 28](#_Toc48901109)

[**3.11.3** **보안대책** 29](#_Toc48901110)

[**3.12** **파일 다운로드 취약점** 29](#_Toc48901111)

[**3.12.1** **취약점 정의** 29](#_Toc48901112)

[**3.12.2** **상세분석** 29](#_Toc48901113)

[**3.12.3** **보안대책** 30](#_Toc48901114)

[**3.13** **관리자 페이지 노출** 30](#_Toc48901115)

[**3.13.1** **취약점 정의** 30](#_Toc48901116)

[**3.13.2** **상세분석** 30](#_Toc48901117)

[**3.13.3** **보안대책** 31](#_Toc48901118)

[**3.14** **데이터 평문 전송** 31](#_Toc48901119)

[**3.14.1** **취약점 정의** 31](#_Toc48901120)

[**3.14.2** **상세분석** 31](#_Toc48901121)

[**3.14.3** **보안대책** 33](#_Toc48901122)

**표 목차**

[<표1- 1> 진단일정 8](#_Toc48899468)

[<표1- 2> 수행인원 8](#_Toc48899469)

[<표1- 3> WEB 진단대상 8](#_Toc48899470)

[<표1- 4> 웹 어플리케이션 소스 보안 진단 체크리스트 10](#_Toc48899471)

[<표2- 5> 취약점 진단 결과 12](#_Toc48899472)

**그림 목차**

[[그림 3- 1] SQL 인젝션 공격으로 관리자 계정 획득 13](#_Toc48901123)

[[그림 3- 2] SQL 인젝션 공격으로 DB 정보 덤프 14](#_Toc48901124)

[[그림 3- 3] icons 경로로 해당 디렉토리 인덱싱이 됨 15](#_Toc48901125)

[[그림 3- 4] 파일이 어느 경로로 저장되는지 확인 16](#_Toc48901126)

[[그림 3- 5] 서버의 정보 확인 17](#_Toc48901127)

[[그림 3- 6] DB 이름을 알아냄 17](#_Toc48901128)

[[그림 3- 7] 악성 콘텐츠 업로드 18](#_Toc48901129)

[[그림 3- 8] phpinfo 함수를 통해 서버 정보 누출 19](#_Toc48901130)

[[그림 3- 9] video태그가 에러가 발생하도록 하고 에러이벤트로 스크립트 실행 20](#_Toc48901131)

[[그림 3- 10] 저장해둔 스크립트를 응답으로 받음 20](#_Toc48901132)

[[그림 3- 11] 네이버로 리다이렉트 21](#_Toc48901133)

[[그림 3- 12] 웹서비스 관리자 로그인 창 22](#_Toc48901134)

[[그림 3- 13] phpMyAdmin 데이터베이스 관리 로그인 창 22](#_Toc48901135)

[[그림 3- 14] 회원정보 수정 페이지 창 23](#_Toc48901136)

[[그림 3- 15] 희생자가 해당 글을 읽으면 공격자가 설정한 글이 작성됨 24](#_Toc48901137)

[[그림 3- 16] 글수정 POST request 25](#_Toc48901138)

[[그림 3- 17] 글 제목이 변조됨 26](#_Toc48901139)

[[그림 3- 18] 세션id의 기간이 1년으로 설정됨 27](#_Toc48901140)

[[그림 3- 19] 데이터베이스 자원이 고갈됨 28](#_Toc48901141)

[[그림 3- 20] 웹쉘 파일 업로드 29](#_Toc48901142)

[[그림 3- 21] 웹서버에서 webshell.php 파일을 전송하면서 실행이 됨 30](#_Toc48901143)

[[그림 3- 22] 누구든지 관리자페이지 접속 가능 31](#_Toc48901144)

[[그림 3- 23] 패킷에서 id와 password 확인 32](#_Toc48901145)

[[그림 3- 24] 검색내용 확인 32](#_Toc48901146)

[[그림 3- 25] 작성된 글 내용 확인 33](#_Toc48901147)

1 개요

1. 1. **목적**

* 운영 중인 Core쇼핑몰 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 같습니다.

|  |  |  |
| --- | --- | --- |
| **구분** | **내용** | **일정** |
| 대상선정 | 대상 협의 / 환경 분석 | 2020.08.18 |
| 기술적 점검 | 취약점 점검 수행 | 2020.08.19-2020.08.20 |
| 결과분석 / 보고 | 취약점 결과보고서 및 보안대책 작성 | 2020.08.20 |
| 보안진단결과보고서 제출 | 2020.08.21 |

<표1- 1> 진단일정

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

|  |  |  |
| --- | --- | --- |
| **수행인원** | **수행업무** | **연락처** |
| 이현제 | 취약점 점검 및 분석 | 010-4052-7130 |
|  |  |  |
|  |  |  |
|  |  |  |

<표1- 2> 수행인원

* 1. **진단대상**

|  |  |  |  |
| --- | --- | --- | --- |
| **구분** | **IP** | **WEB** | **비고** |
| 1 | 70.12.113.49:81 | 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 존재 여부를 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**

Core쇼핑몰사이트 진단 결과 14개의 취약점이 발견되었으며,

불충분한 인증 및 인가 취약점, 파일 업로드 등은 빠른 조처를 할 것을 권고합니다.

또한, SQL 인젝션, 서버 정보 노출, 악성 콘텐츠, 크로스사이트 스크립트, 자동화 공격, 데이터 평문 전송, 불출분한 세션 관리, 파라미터 변조, 관리자 페이지 노출 등 취약점이 발견되었습니다.

서버의 보안 설정과 시큐어 코딩 등과 같은 보안 대책을 권고 합니다.

| **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 | **양호** |
| 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 | **양호** |
| 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

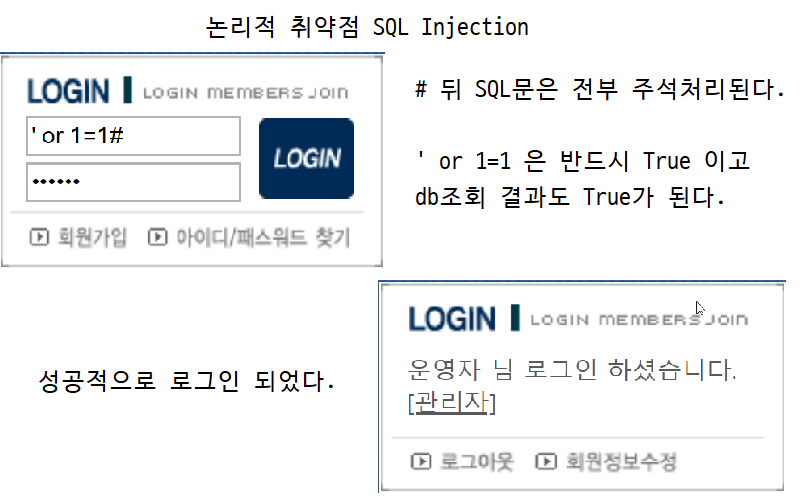
3. 1. **SQL 인젝션**
      1. **취약점 정의**

SQL 인젝션은 응용프로그램 보안상의 허점을 의도적으로 이용해, 개발자가 생각지 못한 SQL문을 실행되게 함으로써 데이터베이스를 비정상적으로 조작하는 공격 방법입니다.

* + 1. **상세분석**

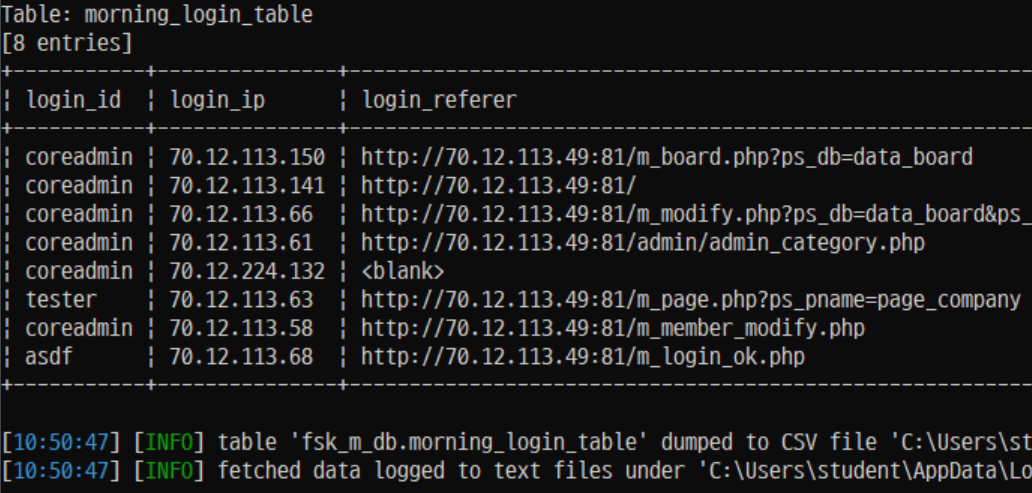
|  |  |
| --- | --- |
| 취약한 경로 | /admin/admin\_login.php   * POST 파라미터: login\_id   /m\_login\_ok.php   * POST 파라미터: login\_id   /m\_board.php   * GET 파라미터: ps\_db   /m\_mall\_detail.php   * GET 파라미터 ps\_ctid, ps\_goid |
| 설명 | SQL 인젝션 공격에 취약한 파라미터가 존재하며, 이 취약점을 통해 DB의 중요한 정보를 공격자가 탈취 가능함 |

**[Step #1] 메인화면에서 SQL 인젝션을 시도한 결과 관리자로 로그인됨**



[그림 3- 1] SQL 인젝션 공격으로 관리자 계정 획득

**[Step #2] SQL 인젝션 자동화 도구 사용**



[그림 3- 2] SQL 인젝션 공격으로 DB 정보 덤프

* + 1. **보안대책**

SQL문의 특수문자들을 필터링을하는 블랙리스트 방식을 사용하지 않고 SQL문을 질의하기 전에 입력값들을 mysql\_real\_escape\_string() 함수를 사용해서 데이터 로써 들어온 입력값들이 SQL문에서의 특별한 문자로 인식되지 않도록 처리하도록 합니다.

mysql\_real\_escape\_string() 함수가 멀티바이트 문자에서는 제대로 공격이 필터링 되지 않고 공격이 그대로 이루어질 수도 있으니 입력된 값들을 mb\_convert\_encoding() 함수를 사용해서 인코딩하는걸 권장합니다.

* 1. **디렉터리 인덱싱**
     1. **취약점 정의**

웹사이트가 사용하고 있는 웹서버의 설정 실수로 인해 인덱싱 기능이 활성화되어 있을 경우, 공격자가 강제 브라우징을 통해 서버 내의 모든 디렉터리 및 파일에 대해 인덱싱이 가능하여 웹 애플리케이션 및 서버의 주요 정보가 노출될 수 있는 취약점

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로 | /icons/. |
| 설명 | /icons/ 경로로 디렉터리 인덱싱이 활성화 |

**[Step #1]** <http://70.12.113.49:81/icons/> 경로로 접속



[그림 3- 3] icons 경로로 해당 디렉토리 인덱싱이 됨

* + 1. **보안대책**

아파치 웹서버의 설정파일인 httpd.conf 파일에서 DocumentRoot 항목의 Options 에서 indexes를 제거하여 디렉토리 인덱싱을 비활성화한다.

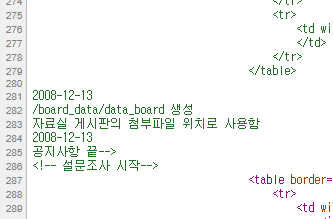
* 1. **정보 누출**
     1. **취약점 정의**

웹사이트의 중요한 데이터가 노출되는 것으로 개발과정의 주석이나 에러 메시지 등에서 중요한 데이터가 노출되어 공격자에게 2차 공격을 하기 위한 중요한 정보를 제공할 수 있는 취약점

* + 1. **상세분석**

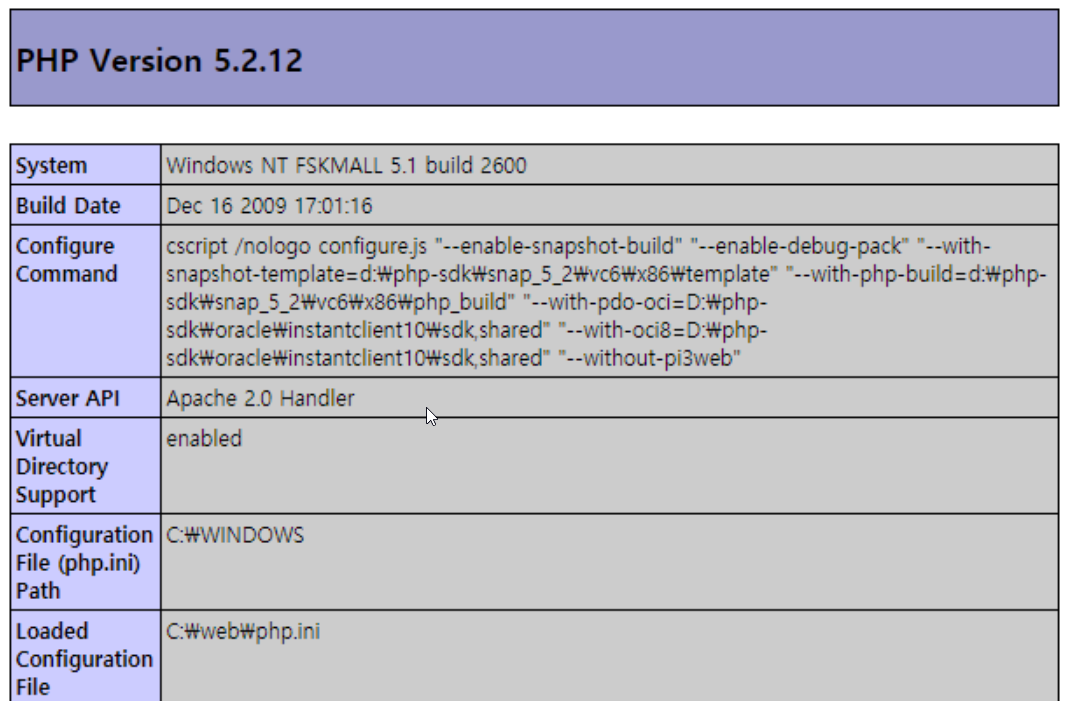
|  |  |
| --- | --- |
| 취약한 경로 | /  /test.php%20  /m\_login\_ok.php   * POST 파라미터 login\_id |
| 설명 | / - 홈페이지 소스코드 에서 파일다운로드 경로가 적힌 주석 확인  /test.php%20 - 서버 테스트 페이지 phpinfo() 발견  /m\_login\_ok.php - 로그인 request의 login\_id 파라미터에 거짓 SQL구문을 넣을경우 .DB의 정보 누출 확인 |

**[Step #1]** 홈페이지 소스코드 안에 있는 개발도중 적힌 주석 확인



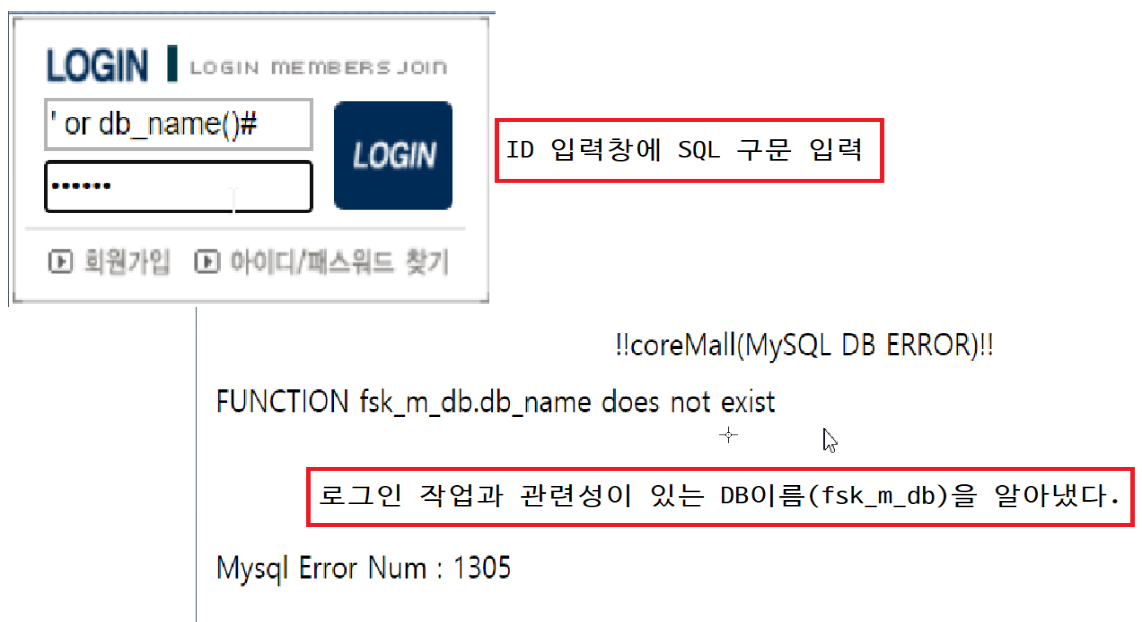
[그림 3- 4] 파일이 어느 경로로 저장되는지 확인

**[Step #2]** /test.php%20 경로로 접속



[그림 3- 5] 서버의 정보 확인

**[Step #3]** 에러기반 SQL Injection 을 이용한 정보노출



[그림 3- 6] DB 이름을 알아냄

* + 1. **보안대책**

개발 당시 추가했던 주석들을 제거한다.

서버 테스트 페이지를 삭제한다.

SQL 질의문에 예외처리를 추가하여 부적절한 에러페이지를 표시하지 않도록 한다.

* 1. **악성 콘텐츠**
     1. **취약점 정의**

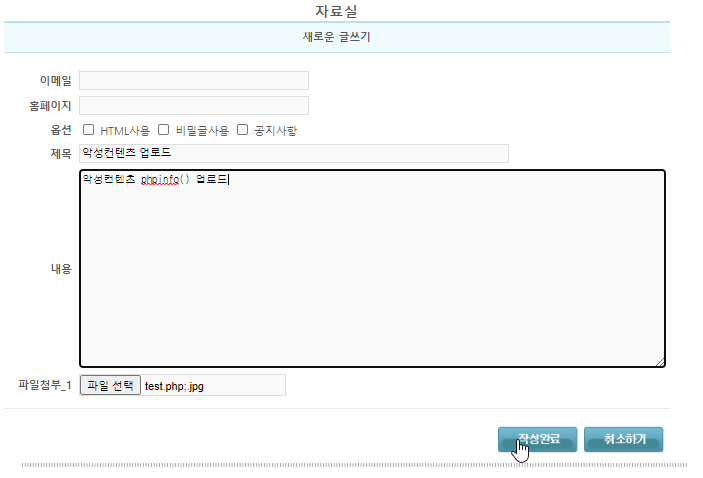
웹서비스에서 사용자 입력값에 대한 필터링이 제대로 이루어지지 않을 경우 공격자가 아성 콘텐츠를 삽입할 수 있다.

악성 콘텐츠가 삽입된 페이지에 접속한 사용자는 악성코드 유포 사이트가 자동으로 호출되어 악성코드에 감염될 수 있는 취약점

* + 1. **상세분석**

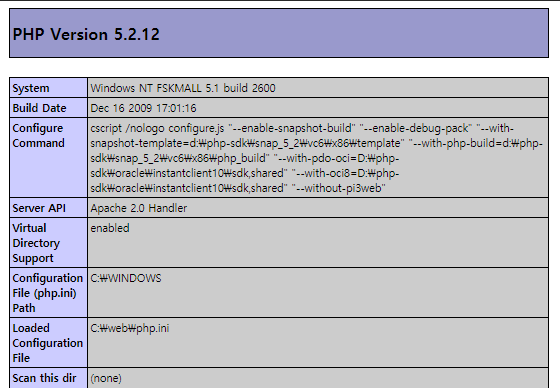
|  |  |
| --- | --- |
| 취약한 경로 | /m\_board\_ok.php   * POST 파라미터 board\_file1 |
| 설명 | 파일업로드 기능을 활용해 악성 콘텐츠를 업로드 한다. |

**[Step #1]** 파일 필터링을 우회하여 test.php 파일 업로드



[그림 3- 7] 악성 콘텐츠 업로드

**[Step #2]** 악성 콘텐츠 실행



[그림 3- 8] phpinfo 함수를 통해 서버 정보 누출

* + 1. **보안대책**

기본적으로 악성 콘텐츠는 SQL Injection, XSS, 파일 업로드 취약점을 이용하여 업로드를 한 뒤 되기 때문에 앞서 이야기한 취약점들을 보안한다.

업로드된 파일들의 이름을 바꾸어 파일저장경로를 알더라도 쉽게 찾을 수 없게 하고 실행 권한을 제거하여 실행하지 못하게 한다.

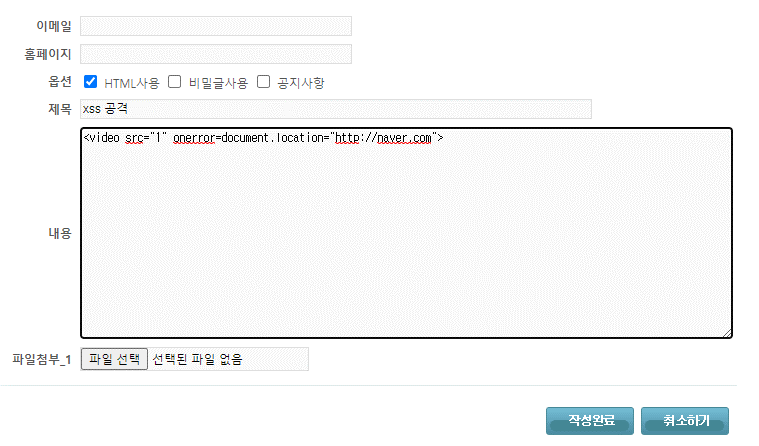
* 1. **크로스사이트 스크립트**
     1. **취약점 정의**

웹서비스에 관리자가 아닌 공격자가 악성 스크립트를 삽입할 수 있는 취약점

* + 1. **상세분석**

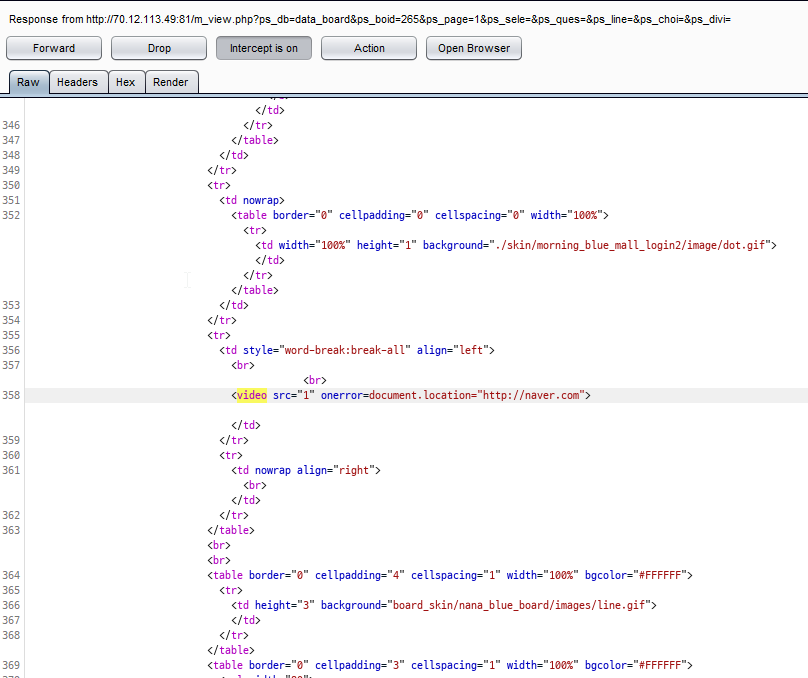
|  |  |
| --- | --- |
| 취약한 경로 | /m\_board.php?ps\_db=data\_board  /m\_board.php?ps\_db=qna |
| 설명 | HTML을 활성화하고 악의적인 스크립트를 작성하여 글 작성 |

**[Step #1]** 악의적인 스크립트가 작성된 글 저장



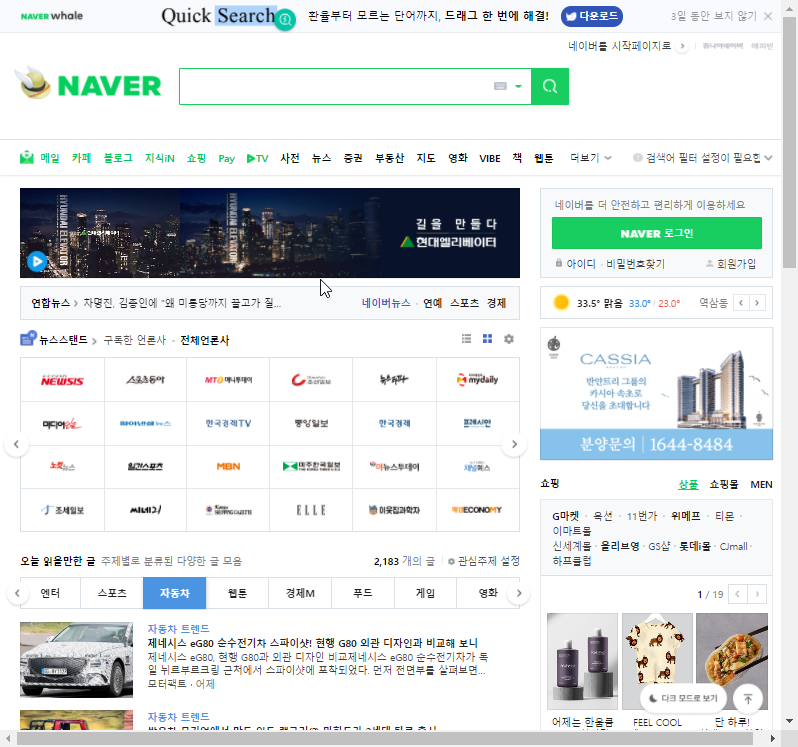
[그림 3- 9] video태그가 에러가 발생하도록 하고 에러이벤트로 스크립트 실행

**[Step #2]** 작성된 글 접속



[그림 3- 10] 저장해둔 스크립트를 응답으로 받음

**[Step #3]** 접속하면 자동으로 악성스크립트 실행됨



[그림 3- 11] 네이버로 리다이렉트

* + 1. **보안대책**

Core몰은 현재 script 태그와 cookie 단어를 필터링해서 XSS 공격과 쿠키 접근을 못 하도록 막지만, 이벤트 핸들러 영역에 접근해서 스크립트를 실행하면 필터링을 우회해서 XSS공격이 가능하고 cookie 접근도 document["coo"+"kie"] 와같이 하면 접근할 수 있다.

HTML 글 작성을 못 하도록 막거나 whitelist 방식으로 특정 태그만 HTML로 입력할 수 있도록 설정한다.

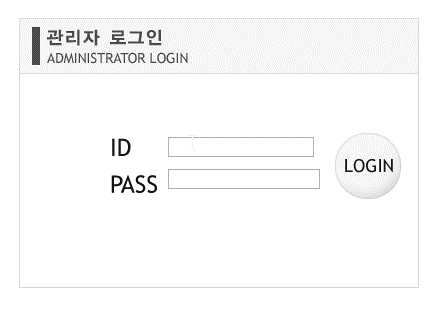
* 1. **불충분한 인증**
     1. **취약점 정의**

민감한 데이터에 접근할 수 있는 경로에 취약한 인증 매커니즘으로 구현된 취약점

* + 1. **상세분석**

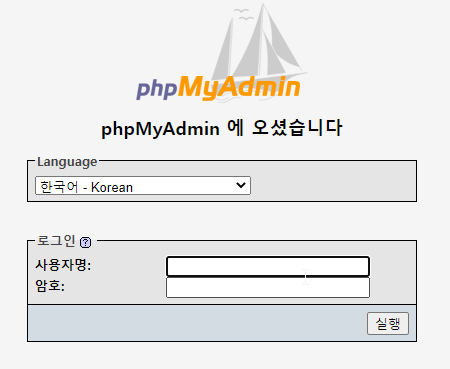
|  |  |
| --- | --- |
| 취약한 경로 | /admin/admin.php  /myadmin/  /m\_member\_modify.php |
| 설명 | 웹서비스 관리자 로그인 페이지, 데이터베이스 관리자 로그인 페이지, 프로필 수정 페이지 에 접근 가능 |

**[Step #1]** 웹서비스 관리자 로그인 페이지 접근



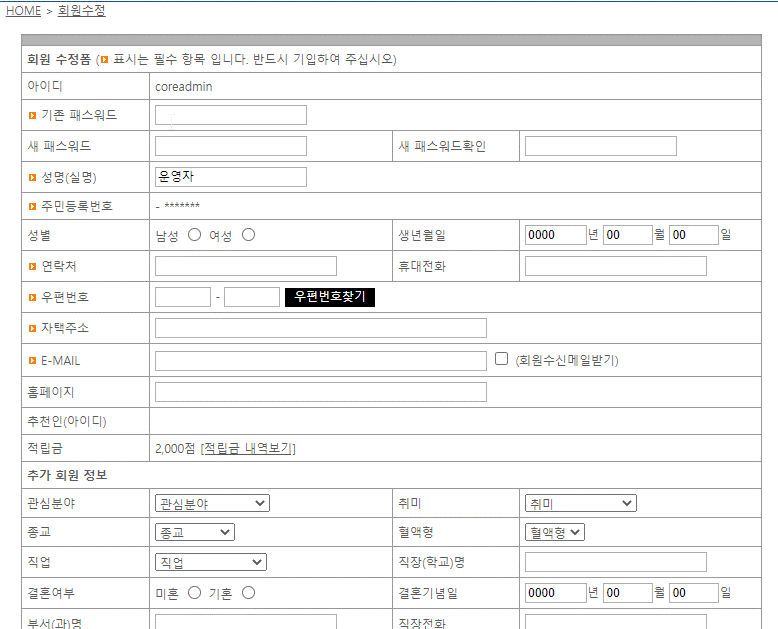
[그림 3- 12] 웹서비스 관리자 로그인 창

**[Step #2]** 데이터베이스 관리자 로그인 페이지 접근



[그림 3- 13] phpMyAdmin 데이터베이스 관리 로그인 창

**[Step #3]** 인증없이 회원정보 수정 페이지 접속



[그림 3- 14] 회원정보 수정 페이지 창

* + 1. **보안대책**

회원 정보 수정 페이지에 접속하기 전에 본인인증을 재확인하는 로직 구현

관리자 페이지를 아무나 접속할 수 없도록 접근제어를 구현해서 특정 ip만 접속할 수 있도록 설정

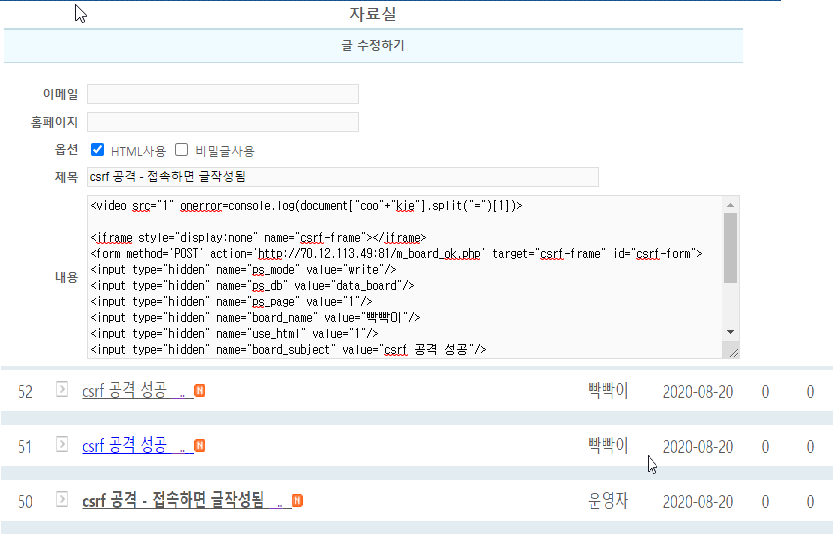
* 1. **크로스사이트 리퀘스트 변조(CSRF)**
     1. **취약점 정의**

웹서비스에서 사용자가 자신의 의지와는 무관하게 공격자가 의도한 행위를 특정 웹서버에 요청하게 만드는 공격

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로 | /m\_board.php?ps\_db=data\_board  /m\_board.php?ps\_db=qna |
| 설명 | HTML을 활성화하고 악의적인 스크립트를 작성하여 글 작성 |

**[Step #1]** html을 허용하고 CSRF공격을 포함한 글 작성



[그림 3- 15] 희생자가 해당 글을 읽으면 공격자가 설정한 글이 작성됨

* + 1. **보안대책**

CSRF는 서버를 대상으로 하는 공격이기에 쓰기/수정이 가능한 GET, POST, PATCH, DELETE 메서드에 보안을 적용하면 된다.

요청을 받은 서버 측에서 request의 referer를 확인해서 domain인 이 일치하는지 검증하고 Security Token을 사용해서 요청시마다 다른 세션 값을 설정하여 서버 측에서 요청을 받을 받을 때 세션에 저장된 토큰값과 요청 파라미터에 전달되는 토큰값이 일치하는지 검증한다.

Double Submit Cookie를 구현해서 요청의 토큰값과 헤더의 쿠키값이 일치하는지도 검사

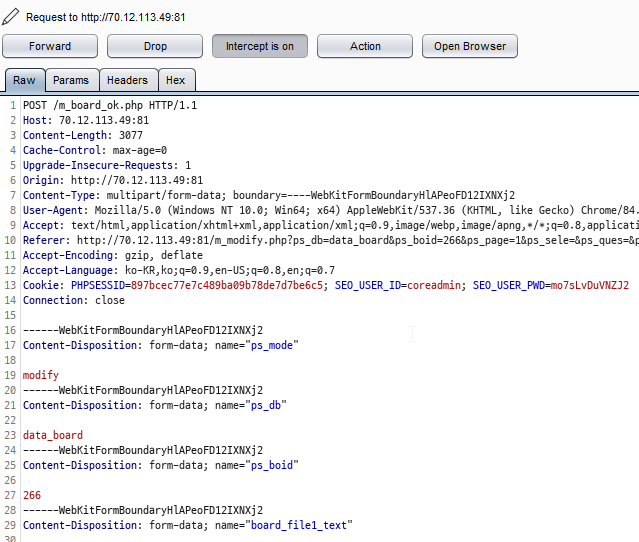
* 1. **불충분한 인가**
     1. **취약점 정의**

권한이 없는 요청에도 접근제어가 제대로 구현되지 못해서 접근이 허용되는 취약점

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로 | /m\_board\_ok.php   * POST 파라미터 ps\_boid |
| 설명 | 타인이 작성한 글도 ps\_boid값을 수정해서 요청하면 수정됨 |

**[Step #1]** burp 를 이용해서 글수정 요청을 인터셉트 한다.



[그림 3- 16] 글수정 POST request

**[Step #2]** 글수정요청에서 글번호값이 변조된 POST 요청이 전송된 결과



[그림 3- 17] 글 제목이 변조됨

* + 1. **보안대책**

중요 기능 또는 데이터 접근 시 2차 인증 프로세스를 구현하고 사용자 접근 통제 로직이 구현되지 않았다면 세션 관리를 통해 사용자 권한을 식별하고 인가된 사용자 요청을 관리한다.

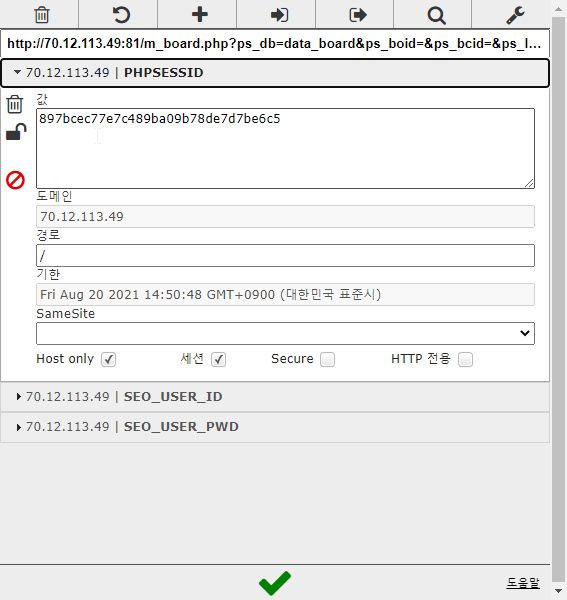
페이지에 부여된 권한의 타당성을 체크 후에 권한 매트릭스를 기준으로 하여 전 페이지에서 권한 체크가 이뤄지도록 구현한다.

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

세션의 만료 기간을 정하지 않거나, 만료 일자를 너무 길게 설정하여 공격자가 만료되지 않은 세션 활용이 가능하게 되는 취약점으로, 세션 만료는 사용자 인증 후 웹 사이트 내에서 이벤트 없이 일정 시간이 지나면 임의로 세션을 종료 시켜 서비스 접근을 제한해야 한다.

* + 1. **상세분석**

**[Step #1]** 쿠키에 저장된 세션ID 확인



[그림 3- 18] 세션id의 기간이 1년으로 설정됨

* + 1. **보안대책**

세션 타임아웃 기능을 구현하여 10분간 웹 서버로의 요청이 없을경우 세션을 삭제한다.

* 1. **자동화 공격**
     1. **취약점 정의**

웹 서버로 정해진 요청을 자동화된 도구를 통해 공격함으로써 수없이 많은 프로세스가 진행되는 취약점

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로 | 모든URL |
| 설명 | 수없이 많은 요청을 해도 전부 응답을 하려 시도함 |

**[Step #1]** 자동화 공격 시도한 결과



[그림 3- 19] 데이터베이스 자원이 고갈됨

* + 1. **보안대책**

여러 번 같은 ip로 같은 요청이 올 경우 captcha와 같은 인증수단을 추가해서 자동화 공격을 방지하고 로그인 같은 요청이 여러 번 올 경우 횟수를 제한하여 패스워드를 5번 이상 틀릴 경우 계정을 잠금 한다.

* 1. **파일 업로드 취약점**
     1. **취약점 정의**

파일 업로드 기능이 존재하는 웹사이트는 업로드되는 파일의 확장자 필터링을 제대로 검증하지 않으면 공격자가 웹쉘을 업로드하여 실행한 다음 원격으로 시스템을 제어할 수 있는 취약점이다.

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로 | /m\_board\_ok.php   * POST 파라미터 board\_file1 |
| 설명 | 파일 확장자 검사를 우회하여 파일을 업로드 |

**[Step #1]** 확장자 필터링 우회를 위해 ;.jpg를 붙여 업로드



[그림 3- 20] 웹쉘 파일 업로드

* + 1. **보안대책**

서비스에 필요한 파일만을 업로드 할 수 있는 화이트리스트 방식의 필터링을 사용한다.

파일의 종류를 확장자만 확인하는 것이 아닌 파일의 매직넘버를 읽어 들어서 확인한다.

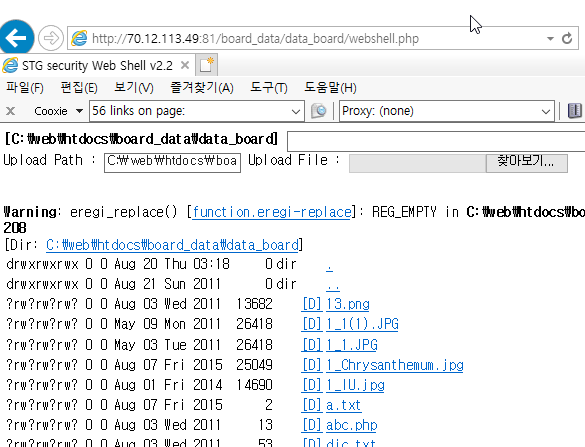
* 1. **파일 다운로드 취약점**
     1. **취약점 정의**

허용된 파일 외의 허용되지 않은 파일을 다운로드 하는 취약점

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로 | /board\_data/data\_board/파일명 |
| 설명 | 위 경로로 get 요청을 하면 업로드 되었던 파일이 다운로드 된다. |

**[Step #1]** /board\_data/data\_board/webshell.php 접속



[그림 3- 21] 웹서버에서 webshell.php 파일을 전송하면서 실행이 됨

* + 1. **보안대책**

. 이나 / 등의 특수문자를 필터링한다.

파일다운로드 모듈 구현 시 실제 파일 경로는 DB에 저장하도록 하고 사용자에게 노출되는 URI의 파라미터에는 인덱스 번호를 통해 다운로드할 파일을 표시한다.

직접 파일의 경로를 입력하여 요청하더라도 허용되지 않은 파일 외에는 접근이 안되도록 한다.

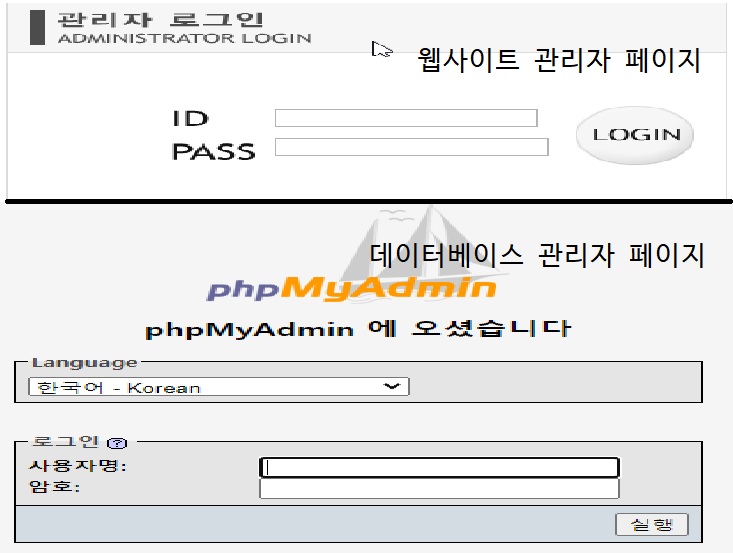
* 1. **관리자 페이지 노출**
     1. **취약점 정의**

공격자에게 관리자 페이지를 노출시켜 관리자페이지에 접근할 수 있는 취약점

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로 | /admin/admin.php  /myadmin/ |
| 설명 | 웹사이트 관리자 페이지와 데이터베이스 관리자 페이지가 노출됨 |

**[Step #1]** 관리자페이지 접속 시도



[그림 3- 22] 누구든지 관리자페이지 접속 가능

* + 1. **보안대책**

관리자페이지에는 접근권한을 가진 ip에서만 접근 가능하도록 접근 제어를 설정한다.

관리자 인증 후 접속 할 수 있는 페이지의 경우해당 페이지 주소를 직접 입력하여 접근하지 못하도록 페이지 각각에 대하여 관리자 인증을 위한 세션을 관리한다.

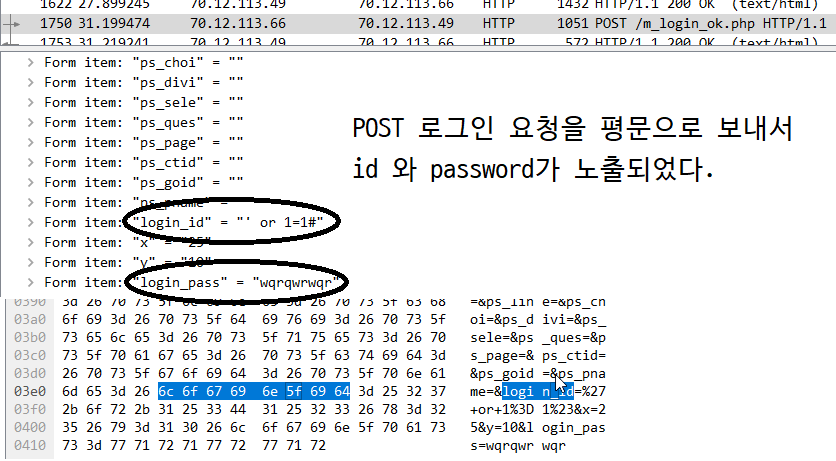
* 1. **데이터 평문 전송**
     1. **취약점 정의**

주민번호, 신용카드 번호, 계좌번호 등의 개인 식별정보와 사용자 id, 패스워드 및 회사의 영업기밀, 핵심 정보 등을 암호화하지 않고 전송하는 취약점

* + 1. **상세분석**

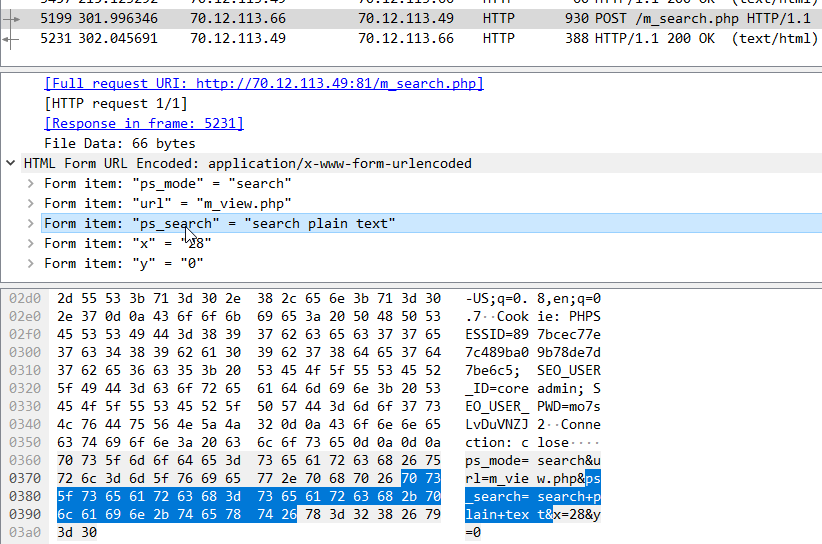
|  |  |
| --- | --- |
| 취약한 경로 | /m\_login\_ok.php  /m\_board\_ok.php  /m\_search.php |
| 설명 | 로그인, 글작성, 검색 모든 파라미터가 암호화되지않고 전송됨 |

**[Step #1]** 로그인 패킷 분석



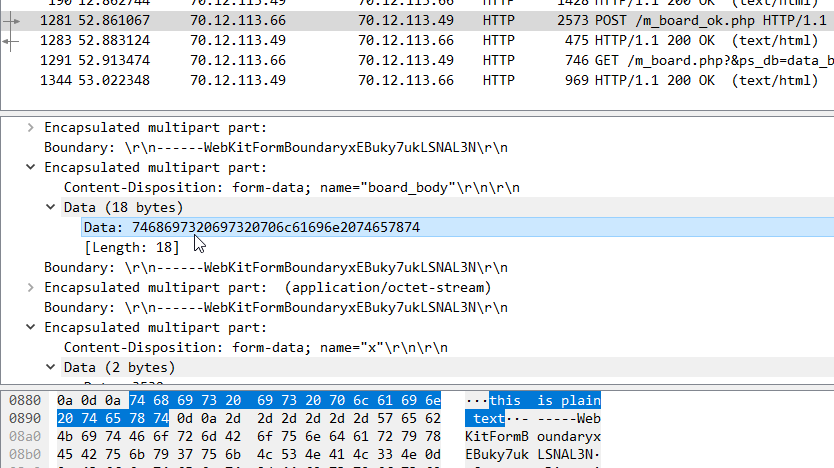
[그림 3- 23] 패킷에서 id와 password 확인

**[Step #2]** 글검색 패킷 분석



[그림 3- 24] 검색내용 확인

**[Step #3]** 글작성 패킷 분석



[그림 3- 25] 작성된 글 내용 확인

* + 1. **보안대책**

웹에서의 정보 전송을 제한하여 불필요한 비밀번호, 주민번호와 같은 중요 정보의 전송을 최소화한다.

중요 정보에 대해서는 반드시 SSL등의 암호화를 사용하여 스니핑으로부터의 위험을 없애야 하고 쿠키와 같이 클라이언트 측에서 노출되는 곳에 비밀번호, 인증값, 개인정보 등의 정보를 기록하지 말아야 한다.

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 |