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

**웹 모의해킹**

**진단 보고서**

Coremall.com

**조상우**

**2020. 08. 21.**

개 정 이 력

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 버전 | 작성일 | 변경내용 | 작성자 | 비고 |
| 1.0 | 2020. 08. 21. | - | 조상우 | - |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목차

[1 개요 7](#_Toc48901243)

[**1.1** **목적** 7](#_Toc48901245)

[**1.2** **진단 방법** 7](#_Toc48901246)

[**1.3** **진단 일정** 7](#_Toc48901247)

[**1.4** **진단 담당자** 7](#_Toc48901248)

[**1.5** **진단대상** 7](#_Toc48901249)

[**1.6** **진단항목** 8](#_Toc48901250)

[2 총평 10](#_Toc48901258)

[**2.1** **총평 요약** 10](#_Toc48901261)

[3 상세 진단 결과 12](#_Toc48901262)

[**3.1** **SQL 인젝션** 12](#_Toc48901266)

[**3.1.1** **취약점 정의** 12](#_Toc48901267)

[**3.1.2** **상세분석** 12](#_Toc48901268)

[**3.1.3** **보안대책** 14](#_Toc48901269)

[**3.2** **정보누출** 15](#_Toc48901270)

[**3.2.1** **취약점 정의** 15](#_Toc48901271)

[**3.2.2** **상세분석** 15](#_Toc48901272)

[**3.2.3** **보안대책** 16](#_Toc48901273)

[**3.3** **크로스 사이트 스크립팅 (XSS)** 17](#_Toc48901274)

[**3.3.1** **취약점 정의** 17](#_Toc48901275)

[**3.3.2** **상세분석** 17](#_Toc48901276)

[**3.3.3** **보안대책** 18](#_Toc48901277)

[**3.4** **불충분한 인증** 19](#_Toc48901278)

[**3.4.1** **취약점 정의** 19](#_Toc48901279)

[**3.4.2** **상세분석** 19](#_Toc48901280)

[**3.4.3** **보안대책** 20](#_Toc48901281)

[**3.5** **취약한 패스워드 복구** 21](#_Toc48901282)

[**3.5.1** **취약점 정의** 21](#_Toc48901283)

[**3.5.2** **상세분석** 21](#_Toc48901284)

[**3.5.3** **보안대책** 21](#_Toc48901285)

[**3.6** **크로스사이트 리퀘스트 변조(CSRF)** 22](#_Toc48901286)

[**3.6.1** **취약점 정의** 22](#_Toc48901287)

[**3.6.2** **상세분석** 22](#_Toc48901288)

[**3.6.3** **보안대책** 23](#_Toc48901289)

[**3.7** **불충분한 인가** 24](#_Toc48901290)

[**3.7.1** **취약점 정의** 24](#_Toc48901291)

[**3.7.2** **상세분석** 24](#_Toc48901292)

[**3.7.3** **보안대책** 26](#_Toc48901293)

[**3.8** **자동화 공격** 27](#_Toc48901294)

[**3.8.1** **취약점 정의** 27](#_Toc48901295)

[**3.8.2** **상세분석** 27](#_Toc48901296)

[**3.8.3** **보안대책** 28](#_Toc48901297)

[**3.9** **파일 업로드** 29](#_Toc48901298)

[**3.9.1** **취약점 정의** 29](#_Toc48901299)

[**3.9.2** **상세분석** 29](#_Toc48901300)

[**3.9.3** **보안대책** 30](#_Toc48901301)

[**3.10** **관리자 페이지 노출** 31](#_Toc48901302)

[**3.10.1** **취약점 정의** 31](#_Toc48901303)

[**3.10.2** **상세분석** 31](#_Toc48901304)

[**3.10.3** **보안대책** 32](#_Toc48901305)

[**3.11** **데이터 평문 전송** 33](#_Toc48901306)

[**3.11.1** **취약점 정의** 33](#_Toc48901307)

[**3.11.2** **상세분석** 33](#_Toc48901308)

[**3.11.3** **보안대책** 34](#_Toc48901309)

[**3.12** **쿠키 변조** 35](#_Toc48901310)

[**3.12.1** **취약점 정의** 35](#_Toc48901311)

[**3.12.2** **상세분석** 35](#_Toc48901312)

[**3.12.3** **보안대책** 36](#_Toc48901313)

**표 목차**

[<표1- 1> 진단일정 6](#_Toc48901314)

[<표1- 2> 수행인원 6](#_Toc48901315)

[<표1- 3> WEB 진단대상 6](#_Toc48901316)

[<표1- 4> 웹 어플리케이션 소스 보안 진단 체크리스트 8](#_Toc48901317)

[<표2- 5> 취약점 진단 결과 10](#_Toc48901318)

**그림 목차**

[[그림 3- 1] [SQL 인젝션] 공격으로 DB 정보 획득 11](#_Toc48901319)

[[그림 3- 2] [SQL 인젝션] 에러기반 공격으로 DB 정보 획득 12](#_Toc48901320)

[[그림 3- 3] [SQL 인젝션] SQL\_MAP 활용하여 DB 정보 획득 12](#_Toc48901321)

[[그림 3- 4] [정보누출] Web Server 내부파일 저장경로 노출 14](#_Toc48901322)

[[그림 3- 5] [정보누출] 소스파일 주석구문 중 의도치 않은 파일저장경로 노출 15](#_Toc48901323)

[[그림 3- 6] [XSS] naver.com으로 이동하는 Stored XSS 16](#_Toc48901324)

[[그림 3- 7] [불충분한 인증] 회원정보 수정 18](#_Toc48901325)

[[그림 3- 8] [취약한 패스워드 복구] 페이지에 변경된 패스워드 노출 20](#_Toc48901326)

[[그림 3- 9] [CSRF] 코드유형 21](#_Toc48901327)

[[그림 3- 10] [CSRF] 공격 상황 22](#_Toc48901328)

[[그림 3- 11] [불충분한 인가] 비인가자 게시판 접근 23](#_Toc48901329)

[[그림 3- 12] [불충분한 인가] 메일주소(Base64) 디코딩 24](#_Toc48901330)

[[그림 3- 13] [불충분한 인가] 비밀번호 변경 취약점을 통한 계정탈취 24](#_Toc48901331)

[[그림 3- 14] [자동화 공격] Repeater을 이용한 공격 26](#_Toc48901332)

[[그림 3- 15] [파일 업로드] 파일 이름 변조 28](#_Toc48901333)

[[그림 3- 16] [파일 업로드] 정보누출 취약점 결합하여 웹쉘 실행 29](#_Toc48901334)

[[그림 3- 17] [관리자 페이지 노출] 페이지에 변경된 패스워드 노출 30](#_Toc48901335)

[[그림 3- 18] [데이터 평문 전송] Wire Shark를 이용한 패킷 분석 32](#_Toc48901336)

[[그림 3- 18] [쿠키 변조] 쿠키내부 민감정보 노출 34](#_Toc48901337)

1 개요

1. 1. **목적**

* Coremall 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. 17. |
| 기술적 점검 | 취약점 점검 수행 | 2020.08.17. - 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** |
| 조상우 | 대상선정, 기술적 점검, 결과분석/보고 | - |
|  |  |  |
|  |  |  |
|  |  |  |

<표1- 2> 수행인원

* 1. **진단대상**

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

Coremall사이트 진단 결과 10개의 취약점이 발견 되었으며, **불충분한 인증 및 인가**

**취약점**, **파일 업로드** 등은 빠른 조치를 취할 것을 권고 합니다.

또한, **SQL 인젝션**, **정보누출**, **XSS**, **취약한 패스워드 복구**, **CSRF**, **자동화공격**,

**관리자 페이지 노출**, **데이터 평문전송** 등의 취약점이 발견되었습니다.

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 | **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 | **취약** |
| 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 | **양호** |
| 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 | **취약** |
| 26 | PL | 위치 공개 | 상 | **취약** |
| 27 | SN | 데이터 평문 전송 | 상 | **취약** |
| 28 | CC | 쿠키 변조 | 상 | **취약** |

<표2- 5> 취약점 진단 결과

3 상세 진단 결과

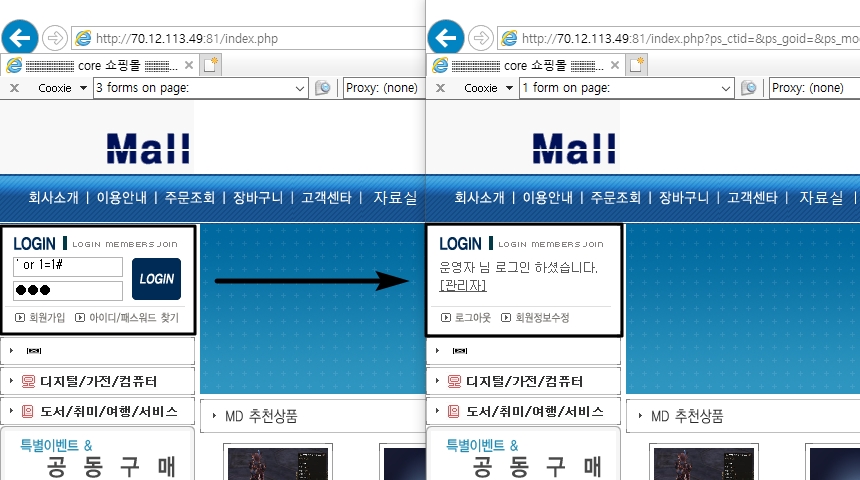
3. 1. **SQL 인젝션**
      1. **취약점 정의**

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

* + 1. **상세분석**

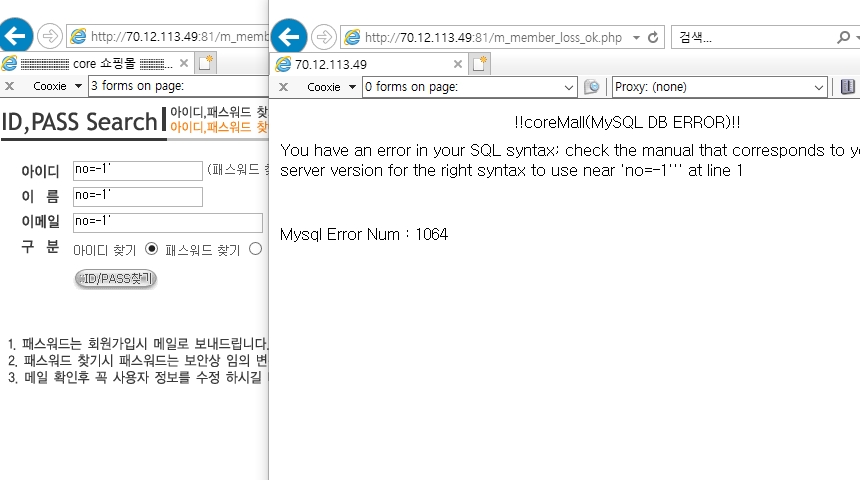
|  |  |
| --- | --- |
| 취약한 경로  (파라미터) | http://70.12.113.49:81/m\_login\_ok.php (param = login\_id)  http://70.12.113.49:81/m\_member\_loss.php (param = loss\_id)  http://70.12.113.49:81/m\_order.php (param = order\_num) |
| 설명 | SQL인젝션 공격에 취약한 파라미터가 존재함.  취약점을 통해 DB의 중요한 정보를 공격자가 탈취 가능함  발견된 경로의 파라미터 외에도 전체적인 경로 및 파라미터에 대한 점검이 필요함. |

**[Step #1]**



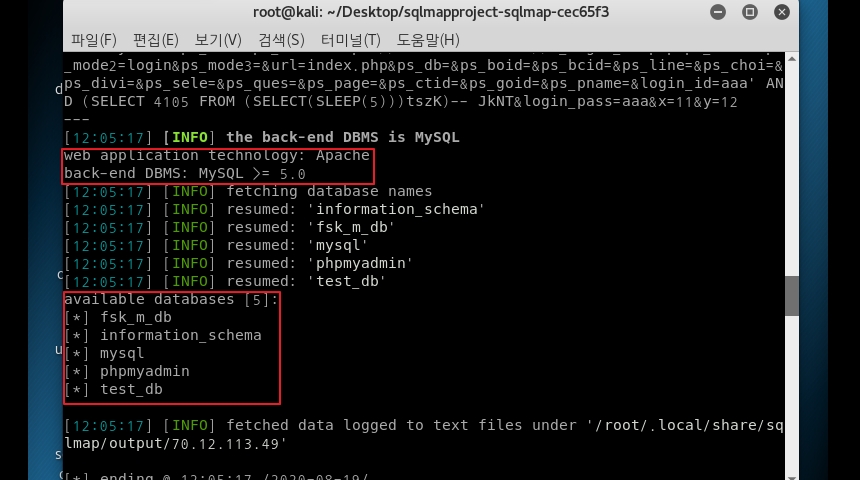
[그림 3- 1] [SQL 인젝션] 공격으로 DB 정보 획득

**[Step #2]**



[그림 3- 2] [SQL 인젝션] 에러기반 공격으로 DB 정보 획득

**[Step #3]**



[그림 3- 3] [SQL 인젝션] SQL\_MAP 활용하여 DB 정보 획득

* + 1. **보안대책**

**[Step #1] 문자열 유효성 검증 로직 구현**

SQL Query에 사용되는 문자열에 대해 유효성 검사를 실시하는 프로세스 구현

아래와 같은 특수문자를 사용자 입력값으로 지정 금지

(아래 문자들은 해당 데이터베이스에 따라 달라질 수 있음)

|  |  |
| --- | --- |
| **문자** | **설명** |
| ‘ | 문자 데이터 구분기호 |
| ; | 쿼리 구분기호 |
| --, # | 해당라인 주석 구분 기호 |
| /\* \*/ | \*와 \*/사이 구문 주석 |

**[Step #2] Dynamic SQL 구문 사용 금지**

Dynamic SQL 구문 사용을 지양하며 파라미터에 문자열 검사 필수적용

**[Step #3] 오류에 대한 예외처리**

에러 메시지는 공격자에게 많은 정보를 제공하므로 오류처리로 정보 노출을 최소화

시스템에서 제공하는 에러 메시지 및 DBMS에서 제공하는 에러 코드가 노출되지

않도록 예외처리

**[Step #4] 웹 방화벽에 인젝션 공격 관련 차단 룰셋 적용**

**[Step #5] 입력값 검증 프로세스는 Client side script가 아닌, Server 페이지로 구현**

* 1. **정보누출**
     1. **취약점 정의**

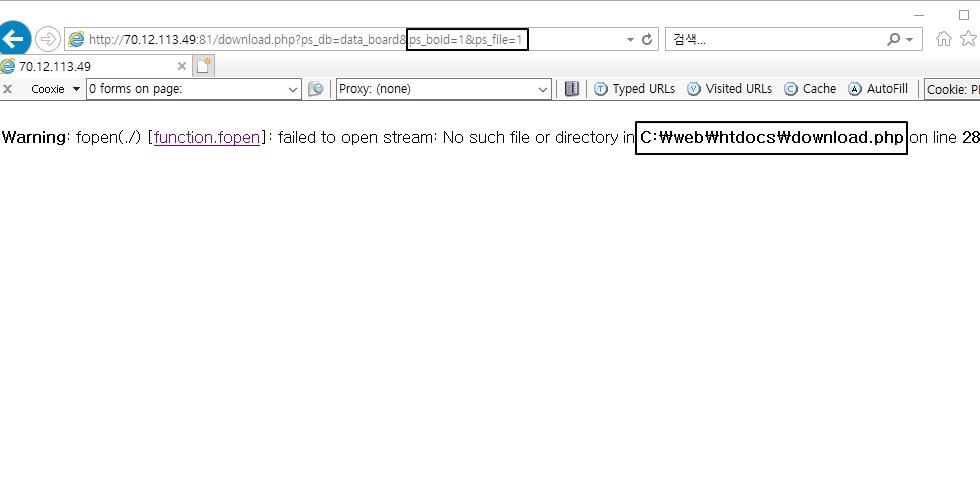
웹 어플리케이션의 민감한 정보가 개발자의 부주의로 인해 노출되는 것으로 중요 정보(관리자 계정 및 테스트 계정 등)를 주석구문에 포함시켜 의도하지 않게 정보가 노출되는 취약점 입니다.

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 있습니다.

* + 1. **상세분석**

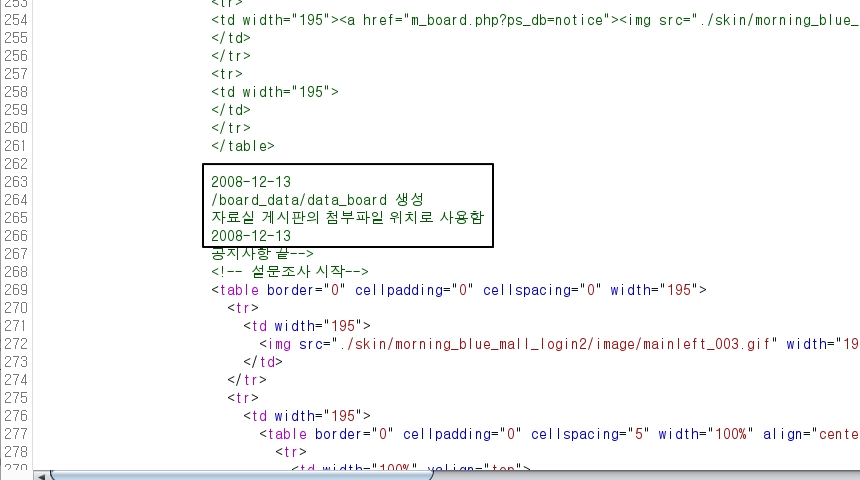
|  |  |
| --- | --- |
| 취약한 경로  (파라미터 및 내용) | http://70.12.113.49:81/download.php (param = ps\_boid)  http://70.12.113.49:81/index.php (페이지 소스 내부 주석) |
| 설명 | Web Server 내 파일 저장 경로 노출  메인페이지 소스 내부 게시판 다운로드 파일 저장경로 노출 |

**[Step #1]**



[그림 3- 4] [정보누출] Web Server 내부파일 저장경로 노출

**[Step #2]**



[그림 3- 5] [정보누출] 소스파일 주석구문 중 의도치 않은 파일저장경로 노출

* + 1. **보안대책**

**[Step #1] 주석 재확인 및 사용금지**

Html 소스 단에 기록되는 정보는 사용자가 웹 브라우저의 소스보기 기능만을 사용해도

간단히 내용을 볼 수 있으므로, html 소스레벨에서 **중요 정보 코멘트를 확인**하거나,

**hidden** 등의 값으로 기록하지 말아야함

**[Step #2] 별도의 에러 페이지로 Redirect**

일반적으로 웹에서 발생하는 에러메시지는 400, 500번대의 에러코드를 리턴하게 되는데,

이러한 에러코드에 대해 **별도의 페이지로 Redirect**하거나 적절한 에러처리 루틴을 설정

하여 처리 되도록 함

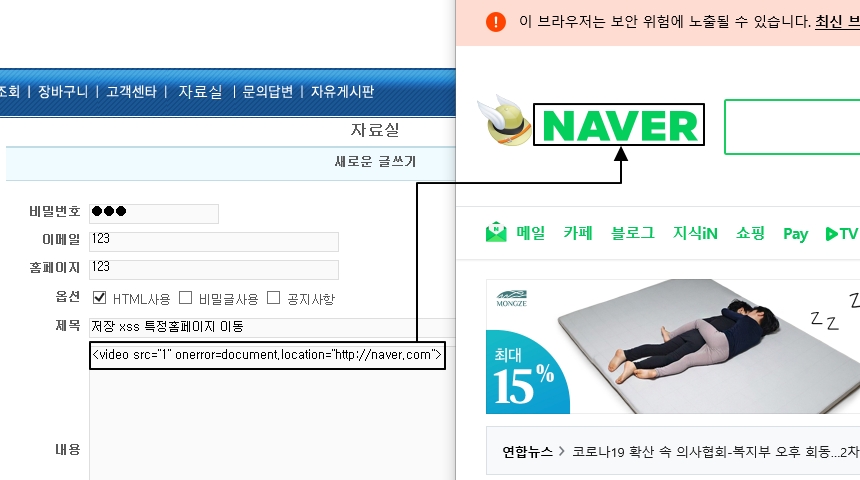
* 1. **크로스 사이트 스크립팅 (XSS)**
     1. **취약점 정의**

웹 애플리케이션에서 사용자 입력 인수 값에 대한 필터링이 제대로 이루어지지 않을 경우, 사용자 인수 값을 받는 웹 사이트 게시판의 게시글이나, 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. 공격방식은 크게 **stored** 공격방식과 **reflected** 공격 방식으로 나누어 짐

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로  (파라미터 및 내용) | http://70.12.113.49:81/m\_write.php?ps\_db=freeboard  (글쓰기 및 댓글)  http://70.12.113.49:81/m\_write.php?ps\_db=data\_board  (글쓰기 및 댓글) |
| 설명 | Stored 공격방식의 크로스 사이트 스크립팅 |

**[Step #1]**



[그림 3- 6] [XSS] naver.com으로 이동하는 Stored XSS

* + 1. **보안대책**

**[Step #1] 스크립트 태그 필터링**

게시물에 HTML이나 자바 스크립트에 해당되는 태그 사용을 사전에 제한하고,

사용자가 입력한 인수 값에 대한 **필터링** 작업 필요

**[Step #2] 모든항목 필터링**

게시물의 본문뿐만 아니라 **제목, 댓글, 검색어입력창, 그 외 사용자 측에서 넘어오는**

**값**을 신뢰하는 모든 form과 인수 값에 대해서 필터링을 수행함

**[Step #3] 서버측에서 구현된 필터링 로직**

입력 값에 대한 필터링 로직 구현 시 공백 문자를 제거하는 trim, replace 함수를

사용하여 **반드시 서버 측에서 구현되어야 함**

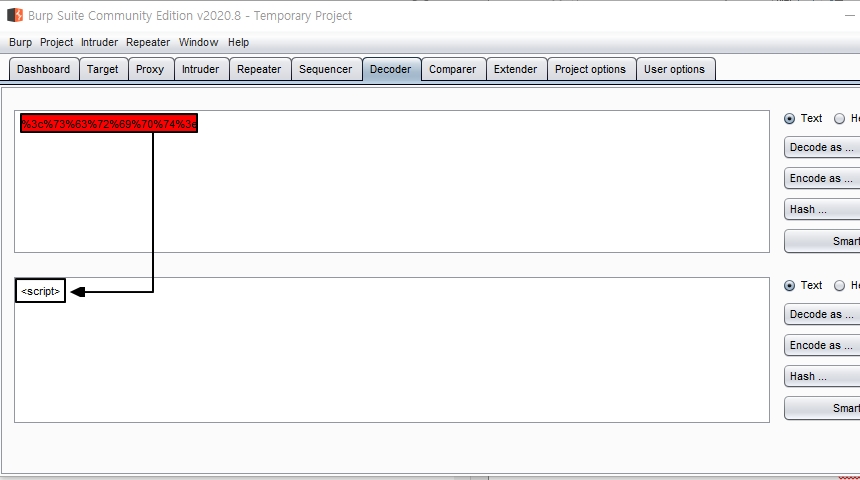
필터링 조치 대상 입력 값

|  |  |
| --- | --- |
| **필터링 조치 대상 입력 값** | |
| 스크립트 정의어 | <SCRIPT>, <OBJECT>, <APPLET>, <EMBED>, <FORM>, <IFRAME>등 |
| 특수문자 | <, >, “, ‘, &, %, %00 등 |

**[Step #4] 사용자 입력값 디코딩**

게시물의 본문뿐만 아니라 제목, 댓글, 검색어입력창, 그 외 사용자 측에서 넘어오는

값을 신뢰하는 모든 form과 인수 값에 대해서 필터링을 수행함



**[Step #5] 웹 방화벽**

웹 방화벽에 모든 사용자 입력 폼(회원정보 변경, 게시판, 댓글, 자료실, 검색, URL 등)을

대상으로 **특수문자, 특수구문 필터링 하도록 룰셋 적용함**

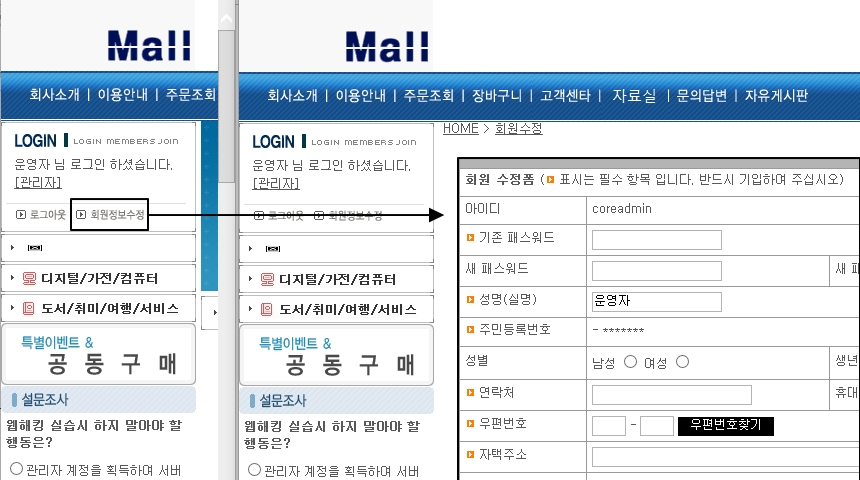
* 1. **불충분한 인증**
     1. **취약점 정의**

중요정보(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 한다.

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로  (파라미터 및 내용) | http://70.12.113.49:81/m\_member\_modify.php (회원정보수정) |
| 설명 | 추가인증 없이 회원정보수정 페이지 진입가능 |

**[Step #1]**



[그림 3- 7] [불충분한 인증] 회원정보 수정

* + 1. **보안대책**

**[Step #1] 본인인증 재확인 로직 구현**

중요 정보(회원정보 변경) 페이지와 같은 중요 정보를 표시하는 페이지에서는 본인

인증을 재확인하는 로직을 구현하고, 인증 후 사용자가 이용 가능 페이지에 접근할

때마다 **승인**을 얻은 사용자인지 페이지마다 **검증**하여야 함

**[Step #2] 구조화, 모듈화**

접근 통제 정책을 구현하고 있는 코드는 **구조화, 모듈화**가 되어있어야 함

**[Step #3] 접근제어가 필요한 모든 페이지에 통제수단 적용**

접근제어가 필요한 모든 페이지에 통제수단(로그인 체크 및 권한 체크)을 구현해야 하며

특히, 하나의 프로세스가 여러 개의 페이지 또는 모듈로 이루어져 있을 때 권한체크가

누락되는 경우를 방지하기 위해서 **공통 모듈**을 사용하는 것을 권장함

**[Step #4] Server side Script(PHP, ASP, JSP 등) 사용**

인증 과정을 처리하는 부분에 **Client Side Script(Javascript, VBScript 등)**를 사용하면

사용자가 임의로 수정할 수 있음

서버에서 처리되는 **Server side Script(PHP, ASP, JSP 등)**로 인증 및 필터링 하여야 함

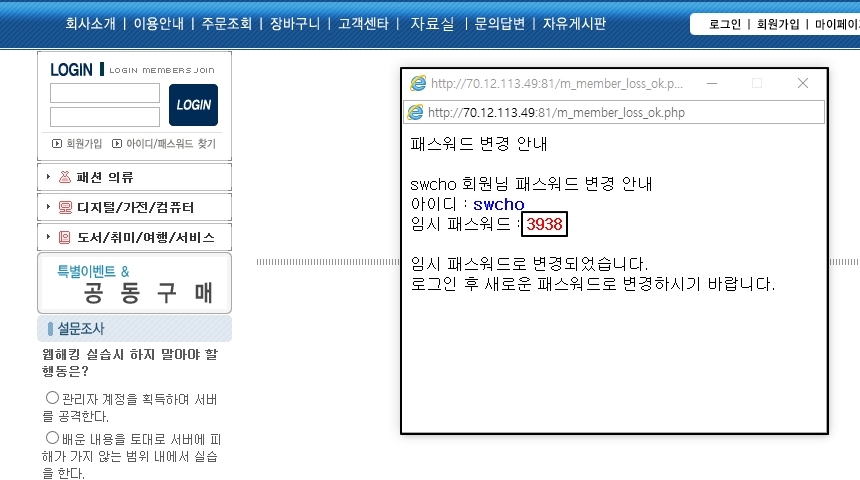
* 1. **취약한 패스워드 복구**
     1. **취약점 정의**

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

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로  (파라미터 및 내용) | http://70.12.113.49:81/m\_member\_loss\_ok.php  (패스워드 변경) |
| 설명 | 추가인증채널 없이 페이지에 바로 패스워드 노출 |

**[Step #1]**



[그림 3- 8] [취약한 패스워드 복구] 페이지에 변경된 패스워드 노출

* + 1. **보안대책**

**[Step #1] 쉬운 패스워드 생성금지**

사용자의 개인 정보(연락처, 주소, 메일 등)로 패스워드를 생성하지 말아야 하며,

난수를 이용한 불규칙적이고 최소길이 6자이상의 패턴이 없는 패스워드를 발급해야 함

**[Step #2] 인증된 채널로 패스워드 발급**

웹 사이트 화면에 바로 출력해주는 것이 아니라 인증된 채널로 패스워드 전송해야 함

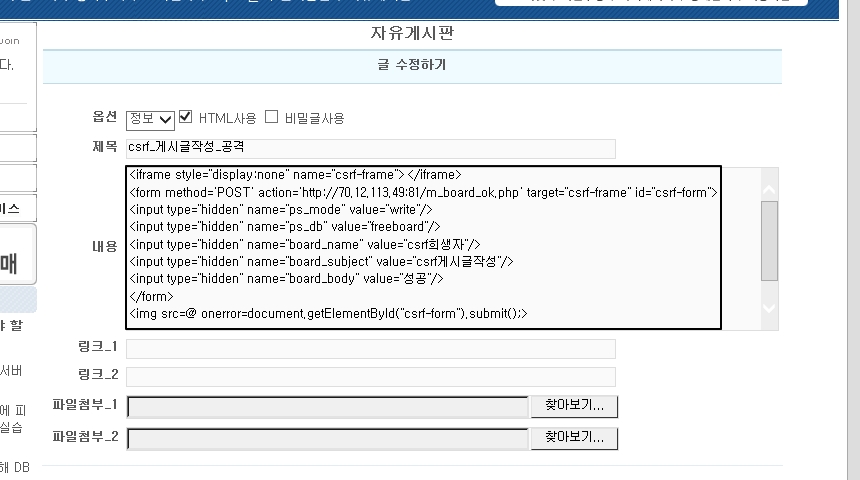
* 1. **크로스사이트 리퀘스트 변조(CSRF)**
     1. **취약점 정의**

사용자의 신뢰(인증) 정보 내에서 사용자의 요청(Request)을 변조함으로써 해당 사용자의 권한으로 악의적인 공격을 수행할 수 있음 **CSRF(Cross-site request forgery)** 희생자는 자신의 의지와는 무관하게 공격자가 의도한행위(수정, 삭제, 등록 등)를 특정 웹사이트에 요청하게 하는 유형의 공격

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로  (파라미터 및 내용) | http://70.12.113.49:81/m\_write.php?ps\_db=freeboard  (글쓰기 및 댓글)  http://70.12.113.49:81/m\_write.php?ps\_db=data\_board  (글쓰기 및 댓글) |
| 설명 | 공격자의 HTML 코드를 이용하여 희생자의 요청을 변조 |

**[Step #1]**



[그림 3- 9] [CSRF] 코드유형

**[Step #2]**



[그림 3- 10] [CSRF] 공격 상황

* + 1. **보안대책**

**[Step #1] 사용자가 입력하는 값에 대한 검증 로직 구현**

**[Step #2] 리퀘스트 변조 구분자 추가**

정상적인 리퀘스트와 비정상적인 리퀘스트를 구분할 수 있도록 Form/URL 에서 임의의

토큰을 추가하고 이 **토큰을 검증**하도록 설계

**[Step #3] 서버측면의 필터링**

HTML이나 자바스크립트에 해당하는 태그 사용을 사전에 제한하고, **서버 단**에서 사용자

입력 값에 대한 필터링 구현

**[Step #4] HTML Editor 사용하는 경우**

HTML Editor 사용으로 인한 상기사항 조치 불가 시,

**서버 사이드 / 서블릿 / DAO(Data Access Object)**영역에서 조치하도록 설계

**[Step #5] XSS 조치 방안 참조**

* 1. **불충분한 인가**
     1. **취약점 정의**

중요 정보 페이지 접근을 위한 인증 로직이 구현되지 않을 경우, 비인가 사용자의 페이지에 접근 및 중요 정보의 열람 및 변조가 가능함. 중요 정보 페이지의 **파라미터 변경**이나, **비인가자 접근 허용**으로 인한 타인의 정보를 열람하고 수정이 가능한 취약점

* + 1. **상세분석**

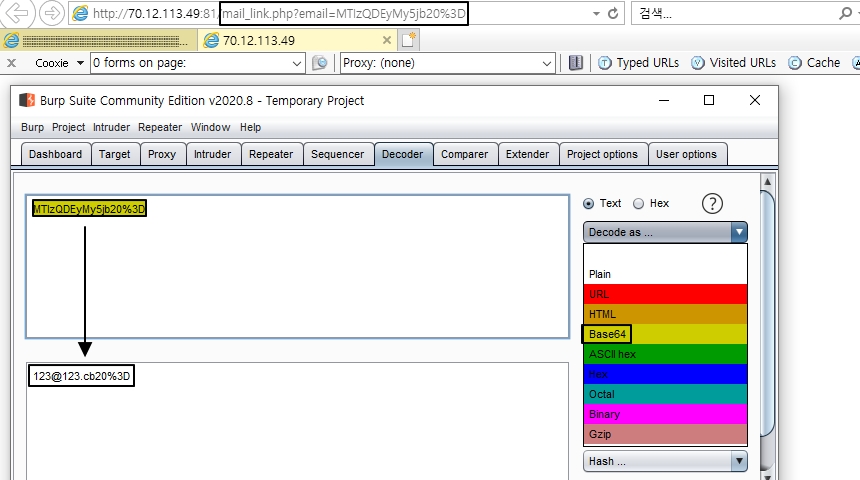
|  |  |
| --- | --- |
| 취약한 경로  (파라미터 및 내용) | http://70.12.113.49:81/m\_write.php?ps\_db=freeboard  (게시판 접근 및 닉네임, Mail링크)  http://70.12.113.49:81/m\_write.php?ps\_db=data\_board  (게시판 접근 및 닉네임, Mail링크) |
| 설명 | 비인가자의 게시판 접근으로 인한 회원정보 열람 가능 |

**[Step #1]**



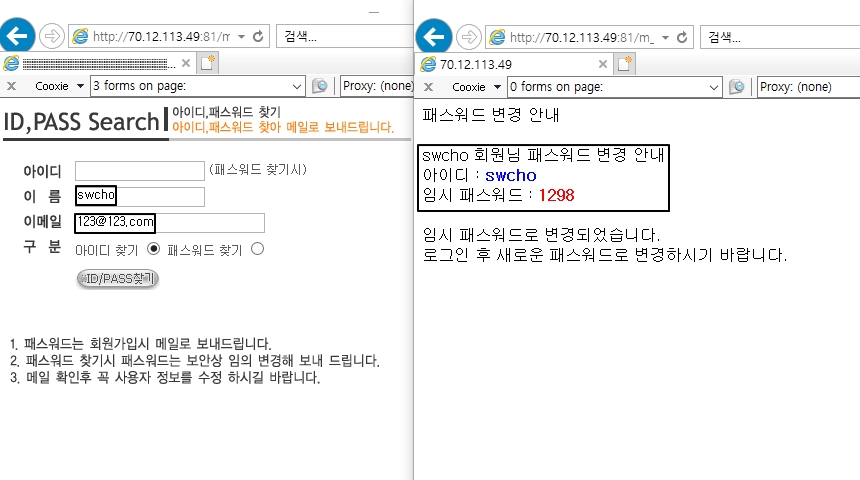
[그림 3- 11] [불충분한 인가] 비인가자 게시판 접근

**[Step #2]**



[그림 3- 12] [불충분한 인가] 메일주소(Base64) 디코딩

**[Step #3]**



[그림 3- 13] [불충분한 인가] 비밀번호 변경 취약점을 통한 계정탈취

* + 1. **보안대책**

**[Step #1] 권한 확인 후 접근통제**

민감한 중요 데이터에 대한 접근 페이지에서 인증을 위한 로직이 구현되지 않았다면,

세션을 통한 인증 및 사용자에게 확인을 위한 인증 값 입력을 통한 인증 절차를

구현하여 **정상적인 로그인 사용자인지** 또는 **권한이 허용된 사용자인지** 여부를 확인 후

해당 페이지에 접근할 수 있도록 함

**[Step #2] 권한 매트릭스 작성**

**페이지별 권한 매트릭스**를 작성하여, 페이지에 부여된 권한의 타당성을 체크 후에 권한

매트릭스를 기준으로 하여 전 페이지에서 권한 체크가 이뤄지도록 구현하여야 함

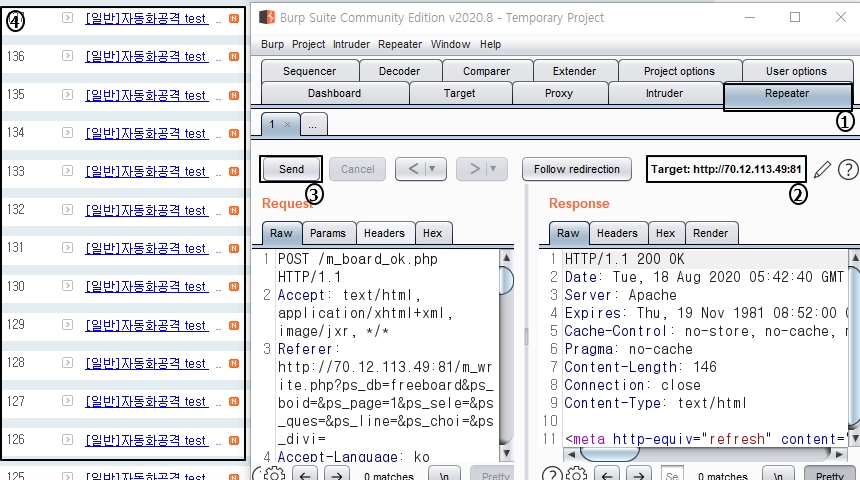
* 1. **자동화 공격**
     1. **취약점 정의**

웹 애플리케이션의 특정 프로세스에 대한 접근 **시도 횟수 제한**을 설정하지 않고 자동화 공격을 방치하면, 웹사이트를 다운시키거나 무차별 대입 공격으로 인해 사용자 계정을 탈취할 수 있고, 데이터 등록 또는 발송 기능 등을 이용하여 악의적인 활용이 가능

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로  (파라미터 및 내용) | http://70.12.113.49:81/m\_write.php?ps\_db=freeboard  (게시글 작성)  http://70.12.113.49:81/m\_write.php?ps\_db=data\_board  (게시글 작성) |
| 설명 | 접근시도 횟수제한 없이 무한정 공격가능 |

**[Step #1]**



[그림 3- 14] [자동화 공격] Repeater을 이용한 공격

* + 1. **보안대책**

**[Step #1] 일회성 확인 로직 구현**

데이터 등록 및 메일 발송 기능에서 사용자 등록이 일회성이 될 수 있도록, 캡차

(이미지를 이용하여 확인 값을 표시하고 사용자가 값을 등록하여 인증함)등 **일회성**

**확인 로직**을 구현하여야 함

**캡차(CAPTCHA)** : 자동화된 컴퓨터와 사람을 판별하기 위한 기술의 일종

**[Step #2] IDS / IPS**

자동화 공격을 시도하면 짧은 시간에 다량의 패킷(양)이 전송되므로 이를 공격으로

감지하고 방어할 수 있는 IDS / IPS의 시스템을 구축하여야 함

**IDS(Intrusion Detecting System)** – 침입탐지시스템

**IPS(Intrusion Preventing System)** – 침입방지시스템



**[Step #3] 모니터링 시스템**

서버에 요청되는 패킷(양)을 모니터링 할 수 있는 시스템 구축이 없이는 적시 적절한

방어가 어려움

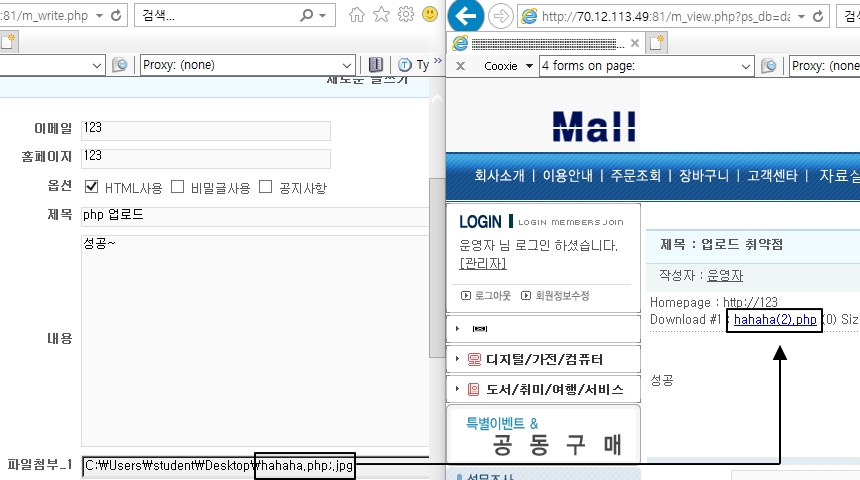
* 1. **파일 업로드**
     1. **취약점 정의**

공격자가 **조작된 Server Side Script 파일**을 업로드 한 후 실행하여, **쉘 권한 획득** 후 홈페이지를 통해 시스템 명령어를 실행하고, 웹 브라우저를 통해 그 결과 값을 보며, 시스템 관리자 권한 획득 또는 인접 서버에 대해 침입을 시도할 수 있음

* + 1. **상세분석**

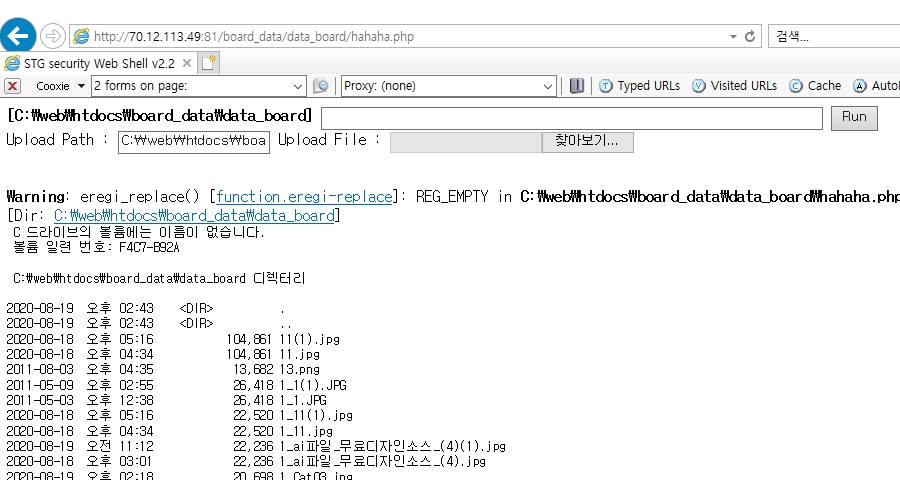
|  |  |
| --- | --- |
| 취약한 경로  (파라미터 및 내용) | http://70.12.113.49:81/m\_write.php?ps\_db=data\_board  (게시글 작성 및 파일 첨부) |
| 설명 | 파일명 변조를 통한 우회 가능 |

**[Step #1]**



[그림 3- 15] [파일 업로드] 파일 이름 변조

**[Step #2]**



[그림 3- 16] [파일 업로드] 정보누출 취약점 결합하여 웹쉘 실행

* + 1. **보안대책**

**[Step #1] 화이트 리스트**

화이트 리스트 방식으로 허용된 확장자만 업로드 허용

**[Step #2] 파일명과 확장자 변경**

업로드 되는 파일을 디렉토리에 저장할 때 파일명과 확장자를 외부 사용자가 추측할

수 없는 문자열로 변경하여 저장(파일 이름은 DB에 저장)

**[Step #3] 업로드 파일 전용 디렉토리 생성**

업로드 파일을 위한 전용 디렉토리를 별도로 생성하여 httpd.conf와 같은 웹 서버 데몬

설정파일에서 실행 설정을 제거함으로써, Server Side Script가 업로드 되더라도 웹

엔진이 이를 실행하지 않는 환경을 설정함

**[Step #4] 방화벽 룰셋 적용**

파일 업로드 필드를 대상으로 특수문자를 필터링 하도록 웹 방화벽 룰셋 적용

* 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**

|  |  |
| --- | --- |
| 취약한 경로  (파라미터 및 내용) | http://70.12.113.49:81/admin/admin\_login.php  (관리자 로그인 페이지) |
| 설명 | 관리자 로그인 페이지 노출 |

**[Step #1]**



[그림 3- 17] [관리자 페이지 노출] 페이지에 변경된 패스워드 노출

* + 1. **보안대책**

**[Step #1] 주소 및 포트 변경**

일반 사용자의 접근이 불필요한 관리자 로그인 페이지 주소를 유추하기 어려운

이름으로 변경하고 관리자 페이지 접근 포트도 변경함

**[Step #2] 세션 관리**

관리자 페이지 주소를 직접 입력하여 접근하지 못하도록 관리자 페이지 각각에 대하여

관리자 인증을 위한 세션 관리

**[Step #3] 웹 서버 설정**

중요한 정보를 가진 웹 서버의 특정 페이지들은 관리자 또는, 특정 사용자만 접근할

필요가 있으므로 이러한 주요 페이지들은 웹 서버에서 적절한 설정을 통하여 특정

사용자만 접근이 가능하도록 사용자 접근 제한이 필요함

**[Step #4] 웹 방화벽**

웹 방화벽을 이용하여 특정 IP만 접근 가능하도록 제한

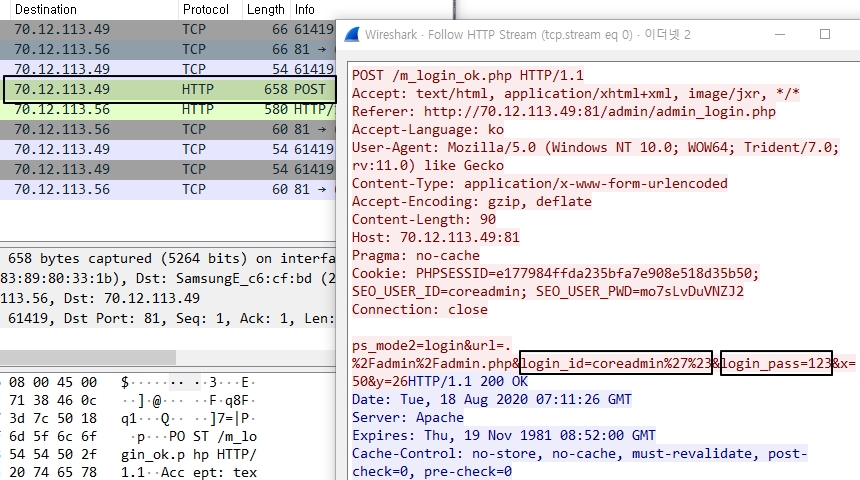
* 1. **데이터 평문 전송**
     1. **취약점 정의**

웹상의 데이터 통신은 대부분 텍스트 기반으로 이루어지기 때문에 서버와 클라이언트 간에 암호화 프로세스를 구현하지 않으면 간단한 **도청(Sniffing)**을 통해 정보를 탈취 및 도용할 수 있음

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로  (파라미터 및 내용) | http://70.12.113.49:81  (coremall 홈페이지 전체) |
| 설명 | http사용 데이터 평문전송 |

**[Step #1]**



[그림 3- 18] [데이터 평문 전송] Wire Shark를 이용한 패킷 분석

* + 1. **보안대책**

**[Step #1] 암호화 통신**

웹상에서의 전송 정보를 제한하여 불필요한 비밀번호, 주민등록번호, 계좌정보와 같은

중요 정보의 전송을 최소화하여야 하며, 중요 정보에 대해서는 반드시 **SSL** 등의 암호화

통신을 사용하여 도청으로부터의 위험을 제거함

**[Step #2] 쿠키 관리**

쿠키와 같이 클라이언트 측에서 노출되는 곳에 비밀번호, 인증인식 값, 개인정보 등의

정보를 기록하지 않음

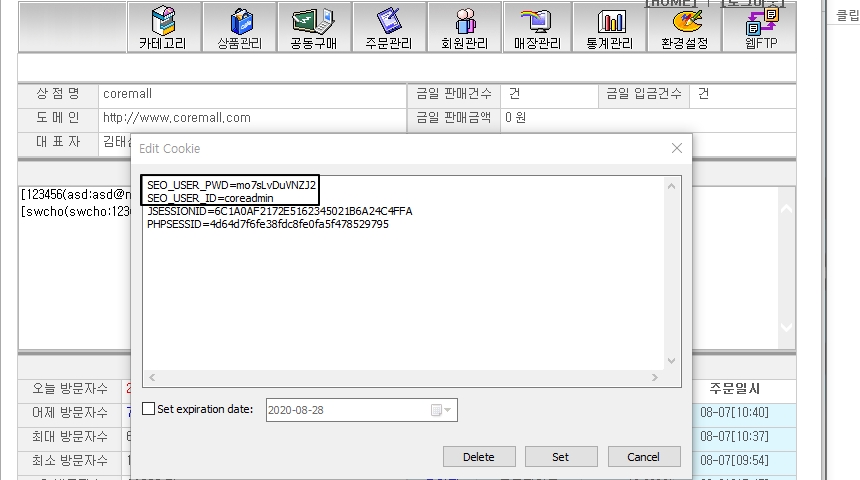
* 1. **쿠키 변조**
     1. **취약점 정의**

쿠키(Cookie)는 클라이언트에 전달되는 값으로 중요 정보로 구성되므로 이 정보의 조작을 통해 다른 사용자의 유효한 세션을 취득할 수 있으며, 기타 중요 정보의 유출 및 변조가 발생할 위험이 존재

* + 1. **상세분석**

|  |  |
| --- | --- |
| 취약한 경로  (파라미터 및 내용) | http://70.12.113.49:81/admin/admin.php  (관리자페이지) |
| 설명 | 쿠키내부 관리자 ID, PW 노출 |

**[Step #1]**



[그림 3- 18] [쿠키 변조] 쿠키내부 민감정보 노출

* + 1. **보안대책**

**[Step #1] Server Side Session**

쿠키 대신 보안성이 강한 Server Side Session을 사용하여 Client Side Session방식인

쿠키의 다양한 구조상 취약점 노출을 방지

**[Step #2] 암호화 알고리즘**

쿠키를 사용해서 중요 정보나 인증을 구현해야 할 경우 안전한 암호화 알고리즘 사용

1. **[별첨] 진단항목**

* 행전안전부에서 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 |