|  |  |  |  |
| --- | --- | --- | --- |
| **과제명** | 리버스 텔넷 및 웹 보안 | | |
| **팀명** | 8조 | **작성일자** | 2017.11.06 |
| **팀원** | 배동현 (20123214) | | |
| 김민지 (20176063) | | |
| 정유정 (20176064) | | |

1. 연구방법

리버스 텔넷과 웹 보안을 위해 XSS(Cross-Site Scripting)을 실습해보고, 리버스 텔넷에 대한 선행작업, 방화벽 정책을 알아본다. 그리고 웹 해킹의 종류와 보안방법들에 대해서도 알아본다.

그리고 국제 웹 보안 표준 기구인 OWASP에서 발표한 웹 주요 취약점 10개를 알아보는 등 웹에 관련된 자료들을 찾아본다.

1. 연구과정

웹 해킹은 웹 사이트의 취약점을 공격하는 기술적 위협으로, 웹 페이지를 통하여 권한이 없는 시스템에 접근하거나 데이터 유출 및 파괴와 같은 행위를 말한다.

유형으로는 웹 엔진 취약점을 이용한 해킹과 웹 애플리케이션의 취약점을 이용한 해킹, 각종 웹 서버 및 미들웨어 기본 제공 샘플 파일을 이용한 해킹 등이 있다.

웹 애플리케이션을 통해서 발생되는 해킹은 주로 일어나는데 이를 OWSAP에서 10대 웹 애플리케이션의 취약점을 발표하였다.

1. Injection (SQL삽입)

* 데이터베이스와 연동된 웹 입력창에 입력값을 SQL구문 등 입력값을 조작하고 DBMS에서 의도치 않은 값을 반환한다.
* 아이디, 패스워드, 개인정보 등의 유출 가능성이 높다.
* 대응방안

1. 사용자 입력창에 일부 특수문자 입력을 차단한다.
2. SQL서버의 에러메시지를 표시하지 않는다.
3. 일반사용자 권한으로 시스템저장 프로시저 접근을 차단한다.
4. Broken authentication and Session management (취약한 인증과 세션관리)

* 인증, 세션관리 관련 애플리케이션 비정상 동작으로 패스워드, 키, 세션토큰, 사용자도용 취약이 발생한다.

1. XSS (Cross-site Scripting)

* 공격자에 의해 작성된 스크립트가 다른 사용자에 전달되어 세션탈취, 웹사이트 변조 등 공격이 발생한다.
* 첫 번째 공격유형 ‘Stored XSS’ 는 게시판, 자료실 등에 평문이 아닌 스크립트 코드를 입력하여 일반 사용자가 게시물을 열람하는 순간 악성스크립트가 실행되어 사용자의 쿠키유출 등 스크립트에 짜여진 공격에 당한다.
* 두 번째 공격유형 ‘Reflected XSS’ 는 공격자는 스크립트가 포함된 공격 URL을 다른 웹사이트, 이메일 등을 통해 전달하여 유입 유도한다. 그래서 스크립트 부분을 인코딩하여 사용자가 알아채지 못하도록 숨어서 전달한다.

1. Broken Access Control (취약한 접근통제)

* 파일, Directory, 데이터베이스 키와 같이 내부적으로 처리되는 오브젝트가 노출되는 경우, 다운로드 취약점등을 이용하여 시스템 파일에 접근할 수 있다.
* UI에서 보여지는 특정기능 수행 전, 기능접근제한권한을 검증해야 하나, 애플리케이션은 각 기능에 대한 접근 시 동일한 접근통제검사 수행이 요구된다. 미수행될 경우 공격자는 비 인가된 기능에 접근하기 위하여 정상적 요청을 변조할 수 있다.

1. Security Misconfiguration (보안 설정 오류)

* 애플리케이션, 프레임워크, WAS, 데이터베이스 서버, 플랫폼의 보안 설정이 정의되고 적용되어 있다. 기본으로 제공되는 값은 종종 안전하지 않기 때문에 보안 설정은 정의, 구현 및 유지되어야 한다. 또한 소프트웨어는 최신의 상태로 유지해야 한다.

1. Sensitive Data Exposure (민감 데이터 노출)

* 많은 웹 애플리케이션들이 신용카드, 개인 식별 정보 및 인증 정보와 같은 중요한 데이터를 제대로 보호하지 않는다.
* 공격자는 신용카드 사기, 신분 도용과 같은 범죄를 수행하는 등 약하게 보호된 데이터를 훔치거나 변경할 수 있다.
* 중요 데이터가 저장 또는 전송 중이거나 브라우저와 교환하는 경우 특별히 주의하며, 암호화 같은 보호조치를 취해야 한다.

1. Insufficient Attack Protection (공격 방어 취약점)

* 대부분의 애플리케이션과 API는 수동 및 자동화된 공격을 모두 탐지, 방지, 대응할 수 있는 기본 기능이 부족하다.
* 공격 방어는 기본적인 입력 유효성 검사를 뛰어 넘어 자동으로 탐지, 로깅, 응답 및 공격 시도 차단을 포함한다. 애플리케이션 소유자는 공격으로부터 보호하기 위해 패치를 신속하게 배포할 수 있어야 한다.

1. Cross-Site Request Forgery (크로스 사이트 요청 변조)

* CSRF 공격은 로그온 된 피해자의 취약한 웹 애플리케이션에 피해자의 세션 쿠키와 기타 다른 인증정보를 자동으로 포함하여 위조된 HTTP 요청을 강제로 보내도록 하는 것이다.
* 공격자가 취약한 애플리케이션이 피해자로부터의 정당한 요청이라고 오해할 수 있는 요청들을 강제로 만들 수 있다.

1. Using Components with Known Vulnerabilities (알려진 취약점이 있는 컴포넌트 사용)

* 슈퍼유저 권한으로 운영되는 취약한 라이브러리, 프레임워크, 기타 소프트웨어 모듈로 인한 데이터유실 및 서버 권한획득과 같은 취약점이 존재한다.

1. Underprotected APIs (취약한 API)

* 최신 애플리케이션 및 API에는 API에 연결하는 브라우저 및 모바일 애플리케이션의 자바스크립트와 같은 여러 클라이언트 애플리케이션이 포함되는 경우가 많다. 이러한 API는 대부분 보호되지 않으며 수많은 취약점을 포함한다.

이렇게 OWASP에서 4년마다 한번씩 보안위협 취약점 TOP 10을 발표한다. 다음으로 웹 언어에 대해서 알아보았다. 인터넷 웹 브라우저에서 사용되는 웹 언어를 HTML, SSS, CSS로 나누어 보았다.

1. HTML

: HTML은 가장 단순한 형태의 웹 언어이다. 웹 서버에 HTML 문서를 저장하고 있다가 클라이언트가 특정 HTML 페이지를 요청하면 웹 서버가 검색하여 해당 HTML 문서를 클라이언트로 전송해주고 브라우저가 HTML을 처리하는데 이런 웹 페이지를 정적인(Static) 웹 페이지라고 한다. 이때 HTML 페이지를 요청하는 것은 HTTP프로토콜을 따르게 된다. 단순한 형태의 HTML문서는 클라이언트의 웹 브라우저를 통해 웹 서버의 무엇인가를 바꿀 수 있는 가능성이 매우 낮기 때문에 웹을 이용한 공격이 매우 어렵다.

1. SSS (Server Side Script)

: ASP나 JSP와 같은 언어가 있으며 이러한 언어는 동적인 페이지를 제공하는 스크립트 언어이다. 스크립트에 HTML 확장자 대신 ASP 또는 JSP의 확장자를 가진 웹 문서를 요청하면 ASP는 DLL이나 OCX 같은 파일을 이용해, JSP는 서블릿을 이용해 요청을 처리하고 그 결과를 HTML 파일로 만들어 클라이언트에 전송한다. 웹 서버가 스크립트 엔진에게 요청하여 스크립트와 일반 HTML페이지를 생성하기 때문에 동적인 페이지를 제공하는 스크립트이다.

1. CSS (Client Side Script)

: 웹 서비스에 이용되는 스크립트에는 자바 스크립트나 비주얼 베이직 스크립트 등이 있는데 이것들은 서버가 아닌 클라이언트 측의 웹 브라우저에 의해 해석되고 적용된다. CSS언어에서는 클라이언트가 자바 스크립트를 작성한다. 클라이언트가 웹 페이지에 포함된 스크립트 파일을 요청하면 웹 서버가 해당 스크립트 파일을 검색하여 해당 스크립트 파일을 브라우저에 반환하게 된다. 브라우저가 클라이언트 측 스크립트를 처리하므로 서버가 아닌 클라이언트 측의 웹 브라우저에 의해 해석, 적용되는 모습을 볼 수 있다.

웹 사이트에서 정보를 수집하는 방법으로는 웹 프록시를 이용하는 방법과 구글 해킹, 파일 접근법 3가지를 알아보았다.

1. 웹 프록시

* 프록시는 클라이언트가 자신을 통해서 다른 네트워크 서비스에 간접적으로 접속할 수 있게 해주는 컴퓨터나 응용 프로그램이다. 웹 해킹에 많이 사용되는 프록시 툴로는 Paros가 있다. Paros를 통해 프록시 서버를 설정 후에 Paros에서 웹 사이트 취약점 스캐닝을 통해 특정 웹 사이트에 취약점을 수집할 수 있다.

1. 구글 해킹

* 웹 해킹을 하면서 많은 정보를 수집하는 유용한 방법으로 검색 엔진을 이용하는 것이다. 이럴 때 흔히 이런 정보 수집 방법으로 구글을 사용한다. 구글에서는 여러 가지 고급 검색 기능을 제공한다. 검색 인자 중 site를 사용하면 특정 도메인으로 지정한 사이트에서 검색하려는 문자열이 포함된 사이트를 찾는다. 검색 인자로 filetype을 사용하면 특정한 파일 유형에 한해서 검색하는 문자가 들어있는 사이트를 찾게 되고 link를 사용하면 링크로써 검색하는 문자가 들어 있는 사이트를 찾게 된다. Cache를 특정 검색어에 해당하는 캐시된 페이지를 보여준다. Intitle은 페이지의 제목에 검색하는 문자가 들어 있는 사이트를 찾으며, Inurl 은 페이지의 URL에 검색하는 문자가 들어 있는 사이트를 찾게 된다. 구글 검색으로 인해 웹 사이트의 취약점이 파악되는 것을 회피하기 위한 방법으로는 서버의 홈 Directory 에 Robots.txt파일을 만들어 검색할 수 없게 만들 수 있다. Robots.txt파일이 있으면 구글 검색 엔진에는 Robots.txt에 있는 Directory 는 검색하지 않는다. 이렇게 검색 엔진의 검색을 차단할 수도 있다.

1. Directory Listing

* Directory Listing은 웹 브라우저에서 웹 서버의 특정 Directory 를 열면 그 Directory 에 있는 파일과 Directory 목록이 모두 나열되는 것을 말한다. 웹 서버의 Directory 구조를 알면 웹 브라우저 주소창에 해당 결로를 입력하여 직접 접근할 수 있고 Directory Listing으로 비정상적인 파일 접근이 가능하게 된다. 상용 프로그램을 이용한 편집은 확장자가 bak나 old인 백업 파일을 자동으로 생성되는 경우가 있다. 이러한 백업 파일들은 서버 측에서 실행되고 결과만 웹 페이지로 보여주는 ASP나 JSP파일과는 달리 클라이언트가 그 파일을 직접 다운로드 받을 수 있다. 이렇게 획득한 소스 파일에는 서버의 로직 뿐만 아니라 네트워크 정보 등 다양한 정보가 포함되어 있을 수 있으며, 대부분의 웹 스캐너는 이러한 파일을 자동으로 찾을 수 있다. 사용 프로그램을 통해 소스 코드를 편집할 경우 임시/백업 파일의 생성과 관리가 중요해진다.

앞서 살펴봤던 XSS에 대해 좀 더 자세히 알아보았다.

XSS 취약점은 공격자에 의해 작성된 스크립트가 다른 사용자에게 전달되는 것으로, 다른 사용자의 웹 브라우저 내에서 적절한 검증 없이 실행되어 발생하는 보안 취약점이다. 공격자가 작성한 스크립트를 통해 정상 사용자의 세션을 탈취하거나, 웹 사이트를 변조, 악의적인 사이트로 사용자를 이동시킬 수 있다. 따라서 XSS는 게시판, 웹 메일 등에 삽입된 악의적인 스크립트에 의해 페이지가 깨지거나 다른 사용자의 사용을 방해하거나, 쿠키 및 기타 개인 정보를 특정 사이트로 전송시키는 공격인 것이다. Reflected XSS는 매개변수를 통해 입력 받은 값을 다시 출력해주는 로직이있을 때 발생하는 취약점이다. 텍스트 입력창에 단순한 텍스트가 아닌 경고문을 출력하는 스크립트를 입력한 경우 스크립트가 실행되면 취약점이 존재하는 것이다. Stored XSS는 게시판의 게시물 등 사용자와 상호작용하여 서버 측에 저장되는 애플리케이션에서 발생하는 취약점이다. 예를 들어 게시판의 게시물에 XSS 라는 알람을 띄우는 스크립트 작성을 저장해두면 다른 사용자가 그 게시물을 클릭할 시 해당 스크립트가 실행되는 것이다. Stored XSS는 게시물에 악성 스크립트를 삽입함으로써 다른 사용자가 게시물을 열람했을 때, 해당 브라우저에서 스크립트가 실행되고, 그때 발생한 쿠키 및 세션 정보를 공격자가 얻는 공격이다.

그리고 리버스 텔넷 기술은 방화벽이 존재하는 시스템을 공격할 때 자주 사용되는 기법이다. 방화벽 정책에서 인바운드 정책(외부에서 방화벽 내부로 들어오는 패킷에 대한 정책)은 일반적으로 80번 포트 외에 필요한 포트 말고는 다 블록을 한다. 그러나 아웃바운드 정책(내부에서 외부로 나갈 때)은 보통 별다른 필터링을 수행하지 않는 경우가 많다. 이런 상황에서 리버스 텔넷은 유용한 기법이다. 서버에서 본인의 컴퓨터로 텔넷으로 접속하게 하는 기술로 일반적인 텔넷은 본인의 컴퓨터가 클라이언트가 되고 접속하고자 하는 상대방의 컴퓨터가 서버가 되지만 서로 바뀐 것으로 리버스 텔넷이라 한다. 이때 Netcat(nc)을 이용하는데 리버스 텔넷 기능뿐만 아니라 스캐닝 기능 등 매우 유용하고 다양한 기능을 수행할 수 있는 프로그램으로 해킹툴은 아니지만 사용을 하다보니 해킹의 용도로 자주 사용되고 있다. 일반적으로 공격자는 웹 서버의 80번 포트로의 접근이 가능하다. 그러나 방화벽 정책에 의해 내부에서 외부로 나가는 정책은 모두 허용이기 때문에 웹 서버에서 공격자 컴퓨터 쪽으로 리버스 텔넷 시도를 하는 것이 가능하게 되는 것이다.

리버스 텔넷은 공격자가 특정 포트를 열어놓고 피해자의 프로그램의 취약점을 공격자가 악용하여 피해자의 컴퓨터가 열어주는 명령을 실행하게끔 프로그램을 공격하는 것으로 단순하게 정확한 일시 일자에 순간 사이트를 공격하고 싶은 경우에는 리버스 텔넷과 같은 공격이 필요없다. 하지만 항상 공격하는 것이 목적일 때 다음과 같은 장점을 가진다.

* 포트 바인딩 쉘 코드 등과 달리 공격자의 서버로 피해자가 접속하는 구조이기 때문에 방화벽의 영향을 받지 않는 경우가 많다.
* 피해자의 프로그램이 웹 서버와 같은 종류라면 괜찮지만 워드와 같은 공격자가 공격을 시도하고 나서 피해자가 실제로 공격을 당하기까지 얼마나 걸리는지 모를 경우에 공격자는 피해자가 자기 서버로 접속하기만을 기다리면 된다.
* 피해자가 계속 서버로부터 오는 명령을 실행하기 때문에 다양한 공격이 가능해진다.

우리는 XSS취약점과 리버스 텔넷을 사용하여 실습해 보았다. (실습은 PPT자료를 참고해주세요.)

1. 연구결과

사용자 입장에서는 팝업차단 생활화가 중요하다. 배포자 입장에서는 HTML코드를 사용 안하는 것도 있으나 이는 서비스의 질이 떨어지게 된다. 그래서 Whitelist기법을 사용하여 HTML의 일부 서비스만 제공함으로 서 취약점 방어가 가능하다. 또는 사용자가 문자열에 스크립트 입력을 막기 위해 <, >, &, “ 등을 문자변환 함수나 메소드를 이용하여 &it, &gt, &amp, &quot로 치환하는 거도 하나의 방법이다.

1. 참고자료

* Tistory, ‘웹 해킹 연습 사이트’, <http://hack-cracker.tistory.com/41\>
* 위키백과, ‘해킹의 정의’,

<https://ko.wikipedia.org/wiki/%EC%9B%B9_%ED%95%B4%ED%82%B9>

* Tistory, ‘웹 해킹 원리와 방어 방법’,

<http://getroot.tistory.com/entry/%EC%9B%B9-%ED%95%B4%ED%82%B9-%EC%9B%90%EB%A6%AC%EC%99%80-%EB%B0%A9%EC%96%B4-%EB%B0%A9%EB%B2%95-%ED%95%98>

* 네이버, ‘XSS 취약점’,

<http://blog.naver.com/PostView.nhn?blogId=axlovelee&logNo=130169233811&parentCategoryNo=&categoryNo=&viewDate=&isShowPopularPosts=false&from=postView>

* 해커스쿨, ‘리버스 텔넷 사용 관련 질문’,

<http://www.hackerschool.org/HS_Boards/zboard.php?id=Free_Board&page=7&sn1=on&divpage=6&sn=on&ss=off&sc=off&select_arrange=reg_date&desc=desc&no=34857>

* Tistory, ‘웹 서버 해킹’, <http://12bme.tistory.com/132>
* EST Security, ‘리버스 텔넷 공격이란’,

<https://www.estsecurity.com/securityCenter/commonSense/view/83>