**Open**

팀명 :

팀장 :

팀원 :

**개 정 이 력**

|  |  |  |  |
| --- | --- | --- | --- |
| 개정 번호 | 개정 내용 요약 | 추가/수정 항목 | 개정 일자 |
| 0.1 | 최종 제정 승인 | 원안 작성 | 2015.12.25 |
| 0.2 |  |  |  |
| 0.3 |  |  |  |
| 0.4 |  |  |  |
| 0.5 |  |  |  |
| 0.6 |  |  |  |
| 0.7 |  |  |  |
| 0.8 |  |  |  |
| 0.9 |  |  |  |
| 1.0 |  |  |  |

**문 서 규 칙**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Open Source를 이용한 IDS구축 및 네트워크 관리 프로그램 | | | ④ |
| Category | 첨부파일 버전 | 문서 최종 수정일 |
| ① | ② | ③ |

* 작성 및 확인은 Microsoft Word 2007으로 작성 되어 졌으며, Acrobat Reader로 읽는다.
* Category(①)에는 Manual, Utility, Tip, Analysis Report 로 구분하며, 기재된 정보가

Manual과 Utility가 혼합된 경우에는 "Manual + Utility" 라고 표기되며, 머리글의

Category에 해당 구분 정보를 표기된다.

* 본 문서의 주제가 되는 대상은 오른쪽 큰 여백에 기재된다. (④)
* 첨부 파일 버전(②)은 첨부 파일이 존재하는 경우에 기재되며, 첨부 파일의 버전이

표기된다. (유틸리티의 경우 최종 버전은 날짜 표기 대신 버전으로 대체한다)

* 문서 최종 수정일(③)에는 문서의 최종 수정날짜가 표기된다.

**목 차**

[1. 프로젝트 개요 7](#_Toc441080698)

[1.1 프로젝트 목표 7](#_Toc441080699)

[1.2 팀원 및 역할 7](#_Toc441080700)

[1.3 프로젝트 일정 8](#_Toc441080701)

[1.4 프로젝트 환경 9](#_Toc441080702)

[1.5 부분별 기대효과 10](#_Toc441080703)

[1.5.1 현황 (AS-IS) 10](#_Toc441080704)

[1.5.2 개선 및 기대효과 (TO-BE) 10](#_Toc441080705)

[1.6 Open Source를 이용한 IDS구축 11](#_Toc441080706)

[1.6.1 IDS 11](#_Toc441080707)

[1.6.2 Snort 12](#_Toc441080708)

[1.6.3 Splunk 13](#_Toc441080709)

[1.6.4 Yara / Cuckoo Sandbox 16](#_Toc441080710)

[2. 프로젝트 내용 19](#_Toc441080711)

[2.1 GNS3 구축환경 19](#_Toc441080714)

[2.2 Snort를 이용한 공격확인 20](#_Toc441080715)

[2.2.1 공격 List 20](#_Toc441080716)

[2.2.2 Ping of Death 21](#_Toc441080722)

[2.2.3 Land Attack 23](#_Toc441080723)

[2.2.4 Nmap 25](#_Toc441080724)

[2.2.5 Netcat 27](#_Toc441080725)

[2.2.6 Telnet 서버 접근 실패 시 패킷 검출 29](#_Toc441080726)

[2.3 Splunk로 Log기록 확인 31](#_Toc441080727)

[2.4 Yara / Cuckoo Sandbox 33](#_Toc441080728)

[2.4.1 Python, Yara 설치 33](#_Toc441080729)

[2.4.2 Cuckoo Sandbox 설정 33](#_Toc441080737)

[2.4.3 Cuckoo Sandbox 실행 34](#_Toc441080738)

[2.4.4 Cuckoo Sandbox 관리자 페이지 35](#_Toc441080739)

[2.5 IP, Port 관리 프로그램 36](#_Toc441080740)

[2.5.1 개요 36](#_Toc441080741)

[2.5.2 Python 코딩 36](#_Toc441080750)

[2.5.3 실행화면 및 사용방법 37](#_Toc441080751)

[3. 결론 38](#_Toc441080752)

[4. 추후과제 및 후기 39](#_Toc441080753)

[5. 참고자료 40](#_Toc441080754)

**표 목 차**

[[표 1‑1] 팀원 및 역할 7](#_Toc441060694)

[[표 1‑2] 프로젝트 일정 8](#_Toc441060695)

[[표 1‑3] 프로젝트 환경 9](#_Toc441060696)

[[표 2‑1] 공격을 알려 주는 기능 List 20](#_Toc441060697)

**그 림 목 차**

[[그림 1‑1] NIDS와 HIDS 11](#_Toc441072461)

[[그림 1‑2] Splunk의 주요기능 13](#_Toc441072462)

[[그림 1‑3] Splunk 그래프와 통계 15](#_Toc441072463)

[[그림 1‑4] Cuckoo Sandbox 실행화면 16](#_Toc441072464)

[[그림 1‑5] Cuckoo Sandbox 파일 분석결과 창 18](#_Toc441072465)

[[그림 2‑1] GNS3 구축 환경 19](#_Toc441072466)

[[그림 2‑2] Show ip route 명령어 실행 결과 20](#_Toc441072467)

[[그림 2‑3] hping3를 이용한 Ping of Death 21](#_Toc441072468)

[[그림 2‑4] 공격 후 패킷 분석 21](#_Toc441072469)

[[그림 2‑5] Snort에서 Local Rules 생성 22](#_Toc441072470)

[[그림 2‑6] Ping of Death에 대한 Snort 알림 화면 22](#_Toc441072471)

[[그림 2‑7] hping3를 이용한 Land Attack 23](#_Toc441072472)

[[그림 2‑8] Land Attack 후 패킷분석 결과 23](#_Toc441072473)

[[그림 2‑9] sameip 옵션 추가 24](#_Toc441072474)

[[그림 2‑10] Land Attack Log기록 24](#_Toc441072475)

[[그림 2‑11] Nmap SYN Scan 25](#_Toc441072476)

[[그림 2‑12] 열린 포트 확인 25](#_Toc441072477)

[[그림 2‑13] SYN, ACK, NULL Rule 정책 추가 26](#_Toc441072478)

[[그림 2‑14] Port Scan 시 알림 26](#_Toc441072479)

[[그림 2‑15] 공격자 Netcat 명령어 27](#_Toc441072480)

[[그림 2‑16] Reverse Shell 연결 27](#_Toc441072481)

[[그림 2‑17] 접속 성공 27](#_Toc441072482)

[[그림 2‑18] Netcat Rule 정책 추가 28](#_Toc441072483)

[[그림 2‑19] Netcat Log기록 확인 28](#_Toc441072484)

[[그림 2‑20] Telnet 접속 실패 29](#_Toc441072485)

[[그림 2‑21] Login 실패 시 정책 29](#_Toc441072486)

[[그림 2‑22] Alert의 Log기록 30](#_Toc441072487)

[[그림 2‑23] Splunk Log Table 31](#_Toc441072488)

[[그림 2‑24] Dashboard 32](#_Toc441072489)

[[그림 2‑25] Python 설치 및 버전 확인 33](#_Toc441072490)

[[그림 2‑26] Yara 설치 및 버전 확인 33](#_Toc441072491)

[[그림 2‑27] cuckoo.conf 33](#_Toc441072492)

[[그림 2‑28] cuckoo.py 실행 34](#_Toc441072493)

[[그림 2‑29] Cuckoo Sandbox 파일분석페이지 35](#_Toc441072494)

[[그림 2‑30] Python 코딩 36](#_Toc441072495)

[[그림 2‑31] Python Overload 실행화면 37](file:///C:\\Users\\Project\\Open Source를 이용한 IDS구축 및 네트워크 관리 프로그램_v1.0_0120.docx" \l "_Toc441072496)

[[그림 2‑32] VIEW 실행 화면 37](#_Toc441072497)

# 프로젝트 개요

* 1. 프로젝트 목표

기존

* 1. 팀원 및 역할

아래

|  |  |
| --- | --- |
| 이름 | 역할 |
|  | - 총 |
| ) | - 네트워크 구 |
|  | - 보고서 작 |

[표 1‑1] 팀원 및 역할

* 1. 프로젝트 일정

아래 [표 1-2]와

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1주차 | | 2주차 | | 3주차 | | 4주차 | | 5주차 | |
| 자료조사 |  |  |  |  |  |  |  |  |  |  |
| 환경 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

[표 1‑2] 프로젝트 일정

* 1. 프로젝트 환경

아래 [표 1-3]의 프로그램들을 이용하여 프로젝트를 진행하였다.

|  |  |  |
| --- | --- | --- |
| 운영 체제  (Snort, Splunk) |  | CentOS 6.7 |
| 운영 체제  (Yara, Cuckoo Sandbox) |  | Ubuntu 15.04 |
| 가상 환경 |  | VMware 11 |
| 개발 언어 |  | Python 2.7.9 |
| 네트워크 구축 |  | GNS3 1.3.4 |
| 공격 시도 |  | Kali Linux 2.3.0.5 |

[표 1‑3] 프로젝트 환경

* 1. 부분별 기대효과
     1. 현황 (AS-IS)
* 비정상적인 공격이 있을 경우 차단 방법 없다.
* .  
  + 1. 개선 및 기대효과 (TO-BE)
* Rule 정책으로 인한 패턴 적용으로 비정상적인 공격 인식 가능하다.

# 실습 환경

2.1 GCP 클라우드

~~2.1.1 정의~~

~~\* GCP에서 제공하는 클라우드 서비스~~

~~2.1.2 설명~~

**\*** GCP 클라우드로 웹서버 + IDS를 구축하여 3명의 팀구성원들이 웹서버를 공격하고, IDS 룰 세팅을 하는 구성

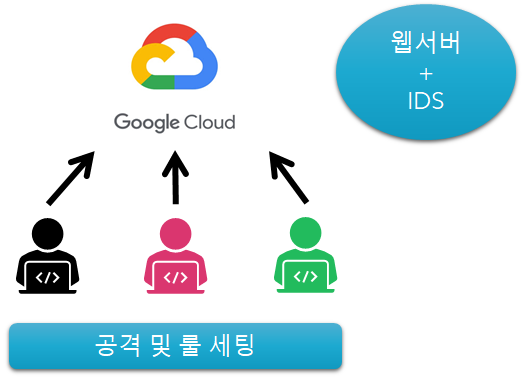


그림 1 GCP 환경 구성

# 프로젝트 내용

2.1 ~~File Inclusion~~

~~2.1.1 정의~~

~~\*~~ File Inclusion 공격 및 취약점이란 PHP로 이루어진 웹페이지에서 파라미터 값을 정확하게 검사하지 않는 경우에 발생하는 공격 및 위약점

- php의 inclusion 함수를 악용함

\* RFI(Remote File Inclusion)와 LFI(Local File Inclusion)로 나뉨

- RFI는 공격자 서버에 존재하는 파일을 가져와 이용하는 공격 기법

- LFI는 공격 대상 서버의 내부에 존재하는 파일을 이용하는 공격 기법

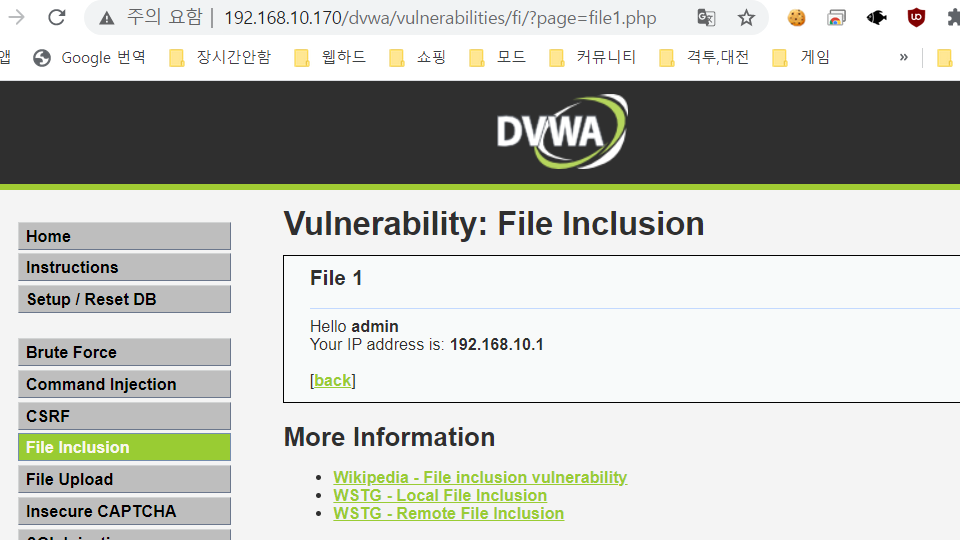
~~2.1.2 공격~~

**[Step #1] 변수값을 이용하는 페이지를 확인**



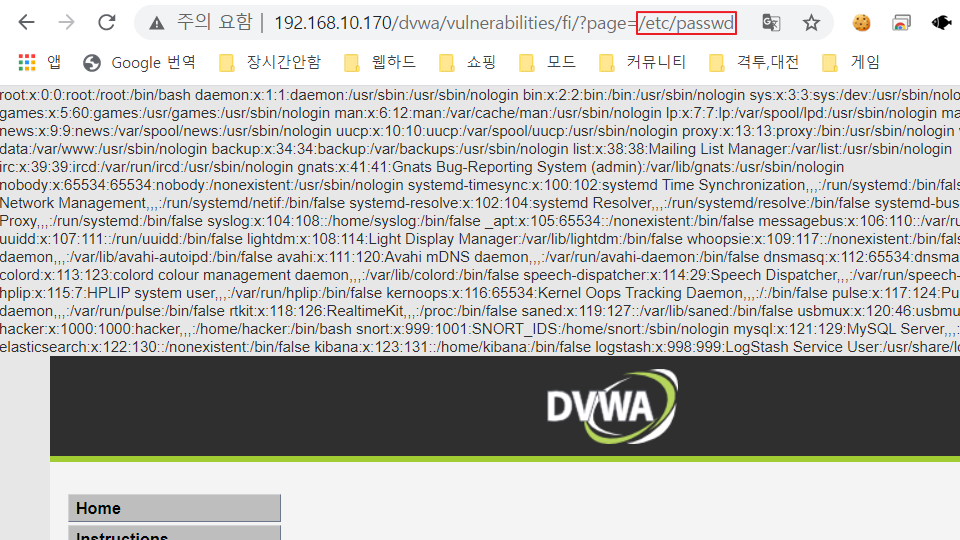
[그림 1 - 13] 변수값 이용하는 페이지 1

**[Step #2] 변수값을 사용해 사이트 내 파일을 보여줌**



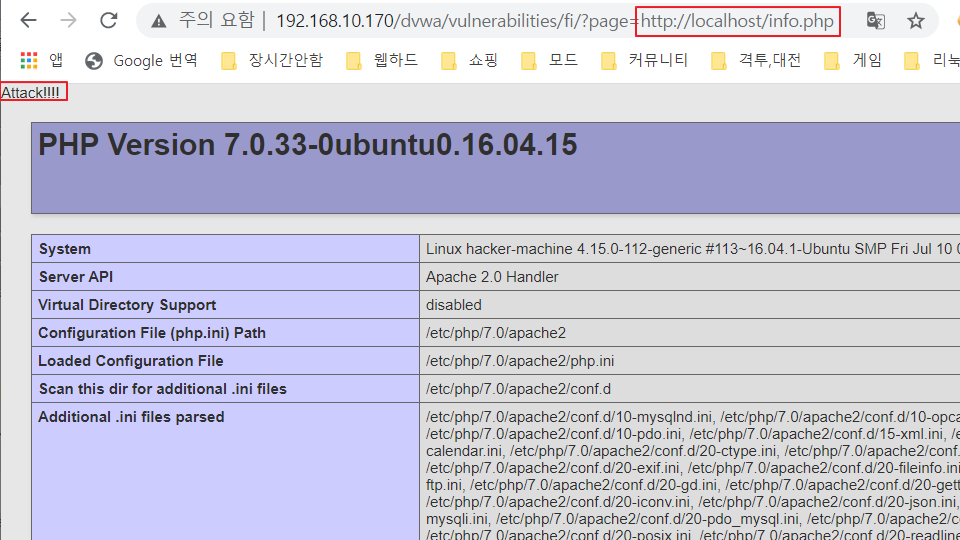
[그림 1 - 13] 변수값 이용하는 페이지 1

**[Step #3]** LFI 공격에 성공해, 서버내 '/etc/passwd' 파일 확인



[그림 1 - 13] 변수값 이용하는 페이지 1

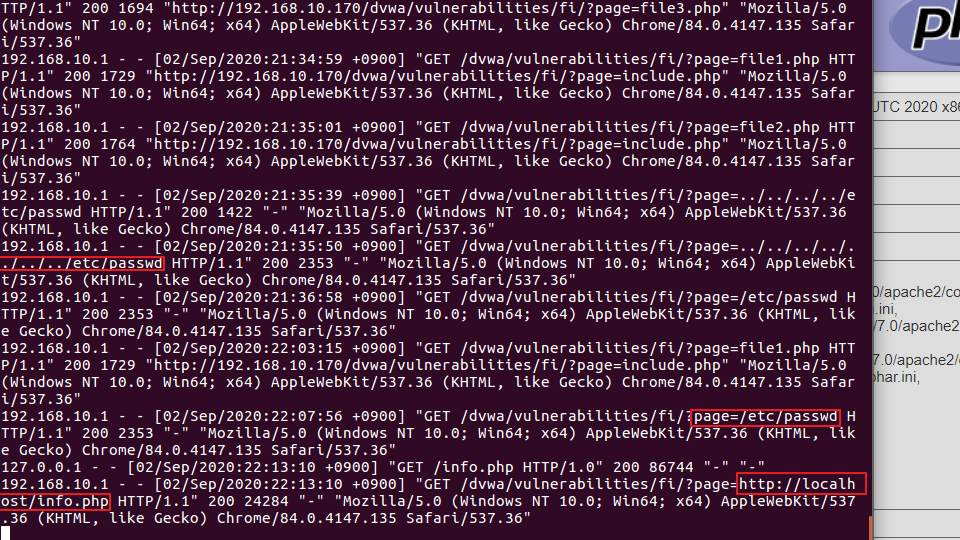
**[Step #4]** RFI 공격에 성공해, 공격자 서버에 존재하는 파일을 가져와 보여주고 있음 (http://localhost : 공격자의 서버라고 가정)



[그림 1 - 13] 변수값 이용하는 페이지 1

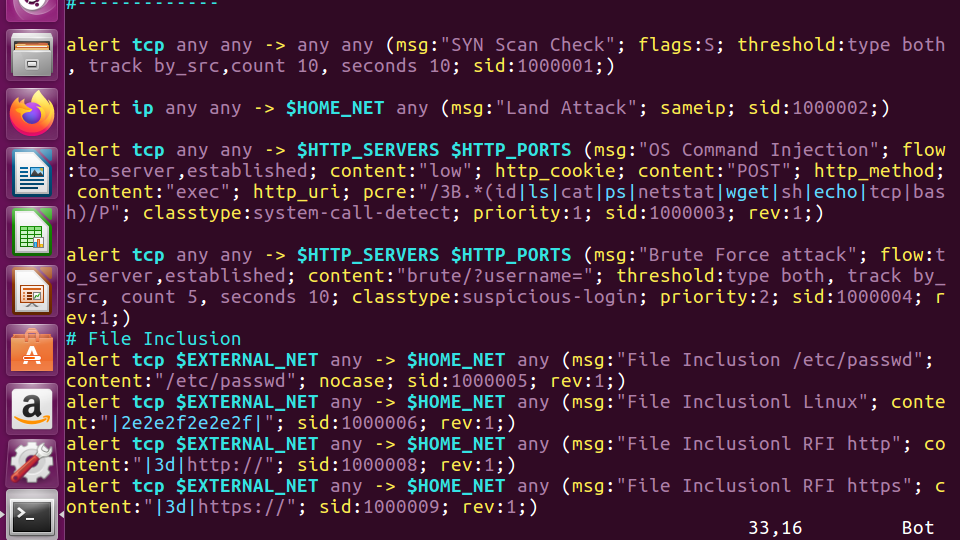
~~2.1.3 룰 세팅~~

**[Step #1] 웹서버 로그를 분석해 공격의 패턴을 확인함**



[그림 1 - 13] 공격 로그 확인

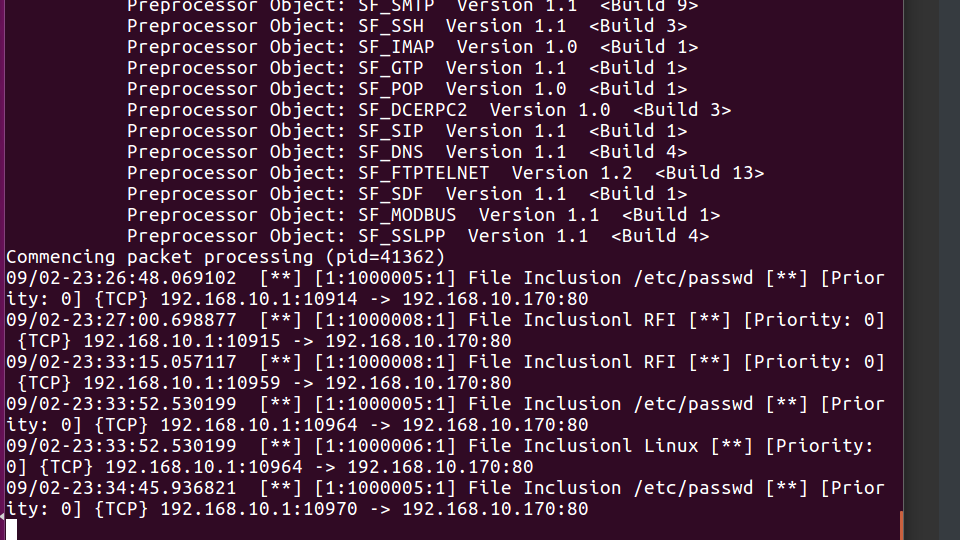
**[Step #2] 확인한 패턴과 공격에 대한 정보를 분석 후, 알맞은 룰 세팅**



[그림 1 - 13] 룰 세팅

~~2.1.4 탐지 확인~~

**[Step #1] 웹서버 로그를 분석해 공격의 패턴을 확인함**



[그림 1 - 13] 공격 로그 확인

2.2 ~~File Upload~~

~~2.2.1 정의~~

\* 파일 업로드 기능이 존재하는 웹 어플리케이션에서 확장자 필터링이 제대로 이루어지지 않았을 경우 공격자가 악성 스크립트 파일(웹쉘)을 업로드하여 해당 웹쉘을 통해 원격에서 시스템을 제어할 수 있는 취약점

\* 공격자는 취약점을 이용하여 시스템 명령 수행 등 시스템을 제어할 수 있으며 웹 페이지 변조 등의 공격도 수행할 수 있음

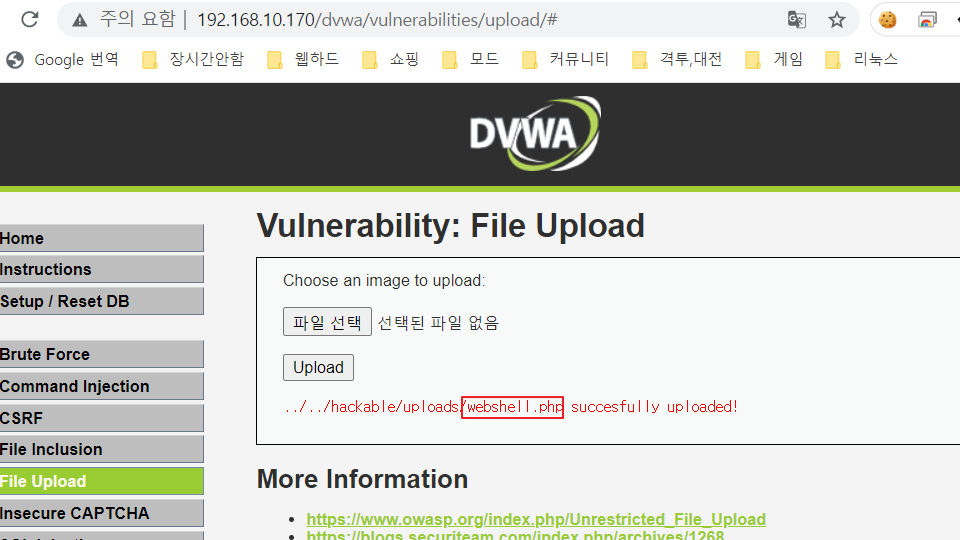
\* 웹쉘

- 악의적인 목적으로 웹서버 내 임의의 명령을 실행할 수 있는 서버 스크립트 파일

- 다양한 언어(ASP, PHP, JSP 등)로 만들어짐

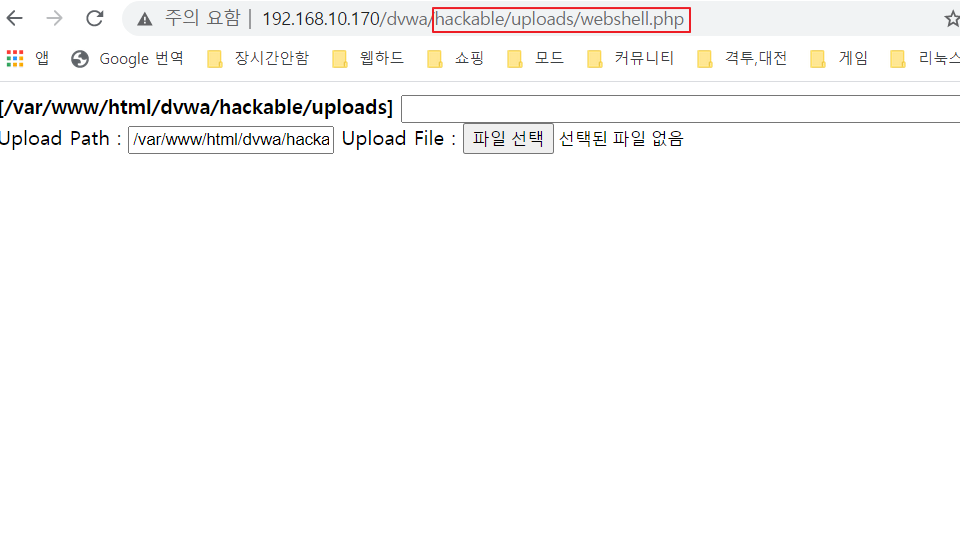
~~2.2.2 공격~~

**[Step #1] 웹셀이 업로드 가능한 것을 확인**



[그림 1 - 13] 웹쉘 업로드

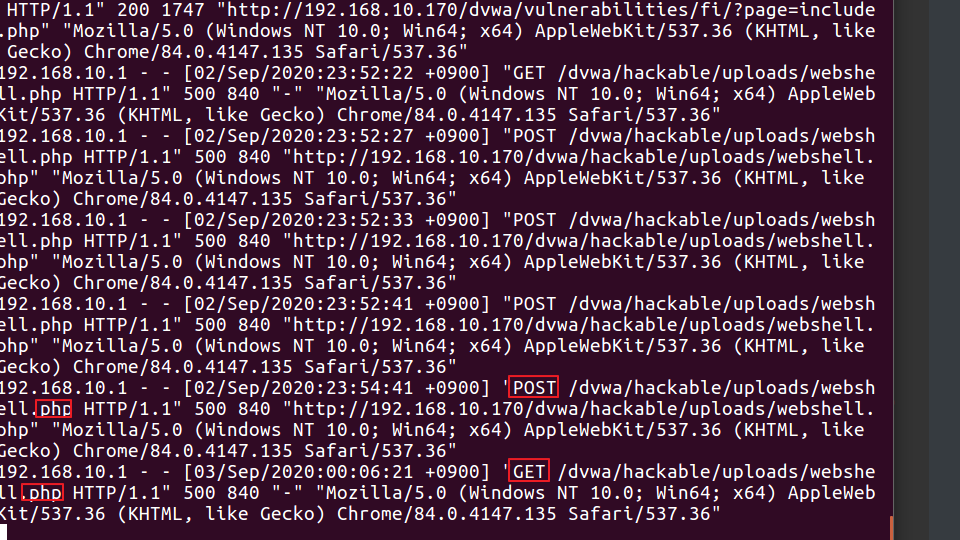
**[Step #2] 업로드시 보여주는 경로를 사용해, 업로드한 웹셀 실행**



[그림 1 - 13] 웹셀 실행

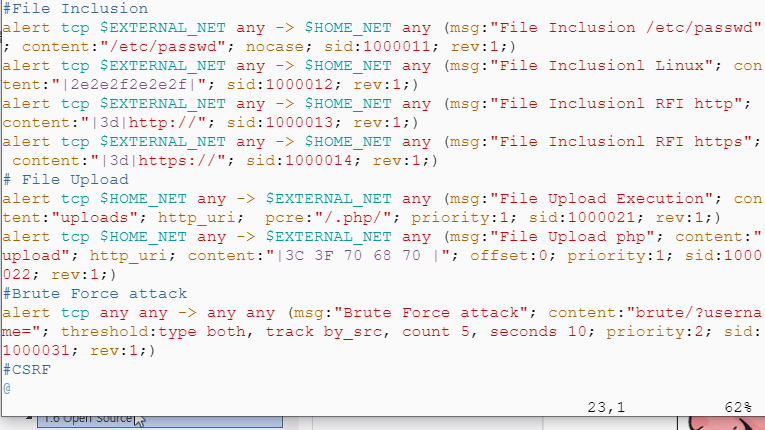
~~2.1.3 룰 세팅~~

**[Step #1] 웹서버 로그를 분석해 공격의 패턴을 확인함**



[그림 1 - 13] 공격 로그 확인

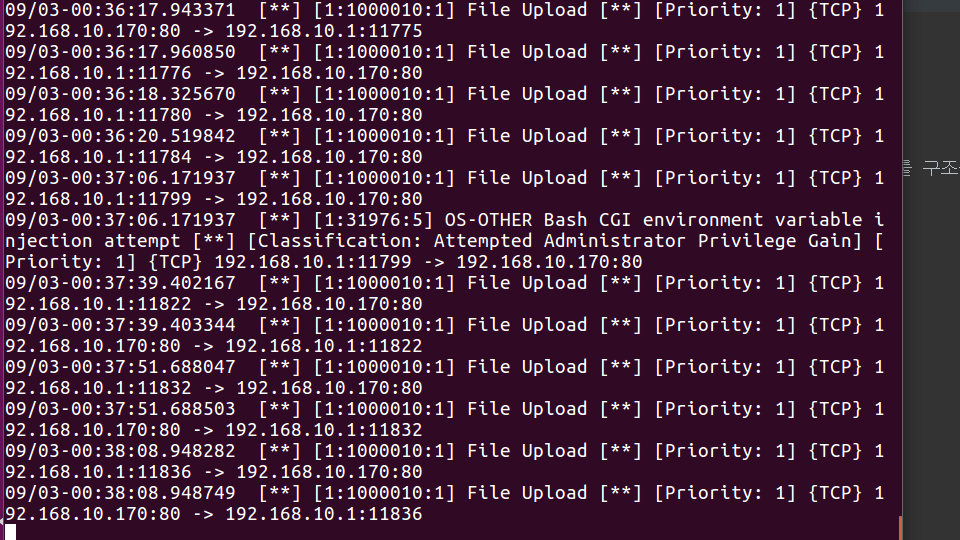
**[Step #2] 확인한 패턴과 공격에 대한 정보를 분석 후, 알맞은 룰 세팅**



[그림 1 - 13] 룰 세팅

~~2.1.4 탐지 확인~~

**[Step #1] 웹서버 로그를 분석해 공격의 패턴을 확인함**



[그림 1 - 13] 공격 로그 확인

2.3 ~~SQL Injection~~

~~2.3.1 정의~~

\* SQL Injection 이란 악의적인 사용자가 보안상의 취약점을 이용하여, 임의의 SQL 문을 주입하고 실행되게 하여 데이터베이스가 비정상적인 동작을 하도록 조작하는 행위

\* 공격 종류

- Error based SQL Injection : 논리적 에러를 이용한 SQL Injection

- Union based SQL Injection : Union 명령어를 이용한 SQL Injection

- Blind SQL Injection : 참, 거짓을 확인하여 이용한 Boolean based SQL

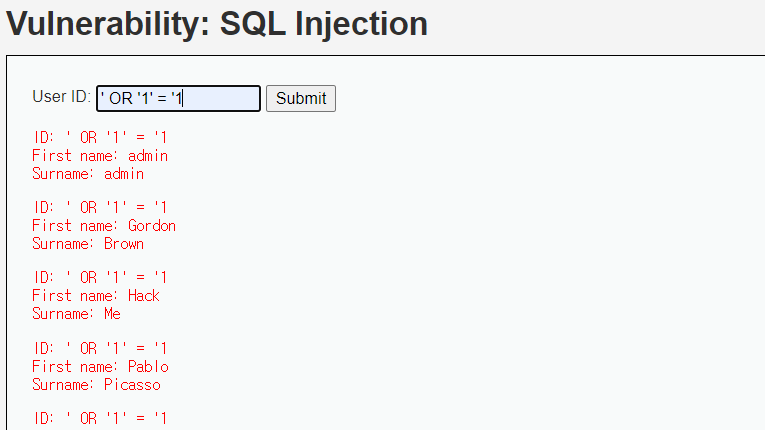
응답에 걸리는 시간을 이용한 Time based SQL

- Stored Procedure SQL Injection : 저장된 프로시저 에서의 SQL Injection

- Mass SQL Injection : 다량의 SQL Injection 공격

~~2.3.2 공격~~

**[Step #1] SQL Injection이 가능한 것을 확인**



[그림 1 - 13] 웹쉘 업로드

~~2.3.4 룰 세팅~~

**[Step #1] 먼저 스노트에서 제공하는 SQL 룰셋을 적용함**



[그림 1 - 13] SQL 룰셋 적용

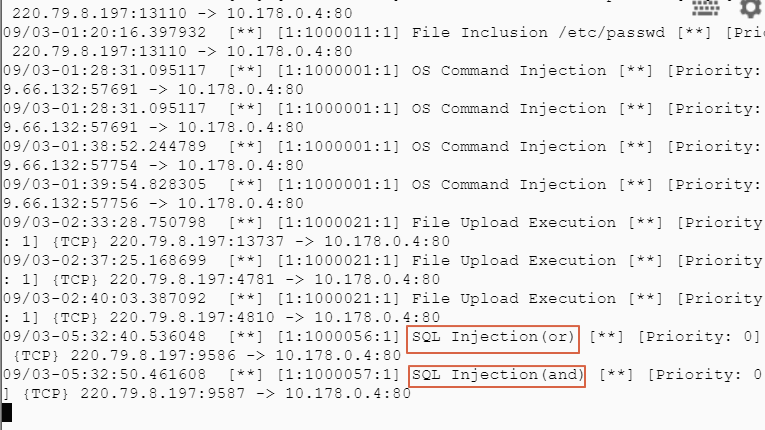
**[Step #2] 추가적인 알맞은 SQL 룰셋을 추가함**



[그림 1 - 13] SQL 룰 세팅

~~2.3.4 탐지 확인~~

**[Step #1] 웹서버 로그를 분석해 공격의 패턴을 확인함**



[그림 1 - 13] 공격 로그 확인

2.4 XSS

~~2.4.1 정의~~

\* XSS(Cross Site Scripting)은 스크립트를 인젝션하여 쿠키 정보 획득 또는 의도하지 않는 동작을 발생시킴

\* 사이트를 교차해서 스크립트를 발생시킴

\* 클라이언트측을 대상으로 한 공격

\* 공격 종류

- Stored 방식

+ 웹 애플리케이션 취약점이 있는 웹 서버에 악성 스크립트를 영구적으로 저장해 놓는 방법

+ 웹 사이트의 게시판, 사용자 프로필 및 코멘트 필드 등에 악성 스크립트를 삽입해 놓으면, 사용자가 사이트를 방문하여 저장되어 있는 페이지에 정보를 요청할 때, 서버는 악성 스크립트를 사용자에게 전달하여 사용자 브라우저에서 스크립트가 실행되면서 공격한다.

- Reflected 방식

+ 웹 애플리케이션의 지정된 변수를 이용할 때 발생하는 취약점을 이용하는 것으로, 검색 결과, 에러 메시지 등 서버가 외부에서 입력받은 값을 받아 브라우저에게 응답할 때 전송하는 과정에서 입력되는 변수의 위험한 문자를 사용자에게 그대로 돌려주면서 발생

- DOM 기반 방식

+ 피해자의 브라우저가 HTML 페이지를 구문 분석할 때마다 공격 스크립트가 DOM 생성의 일부로 실행되면서 공격

+ 페이지 자체는 변하지 않으나, 페이지에 포함되어 있는 브라우저측 코드가 DOM 환경에서 악성코드로 실행됨

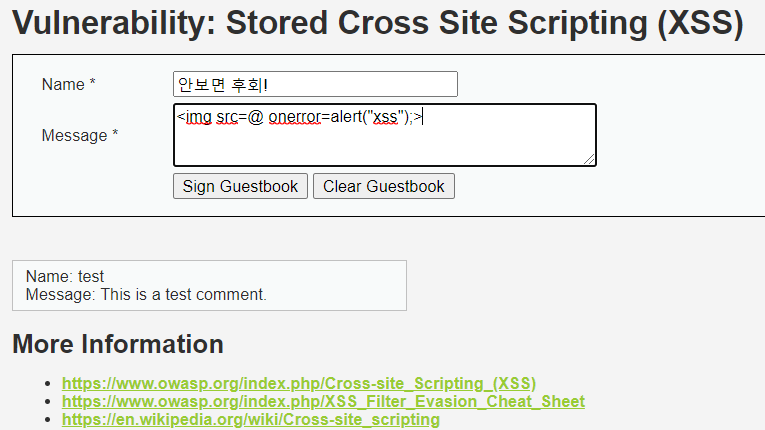
~~2.3.2 공격~~

**[Step #1] XSS Reflected 공격이 가능한 것을 확인**



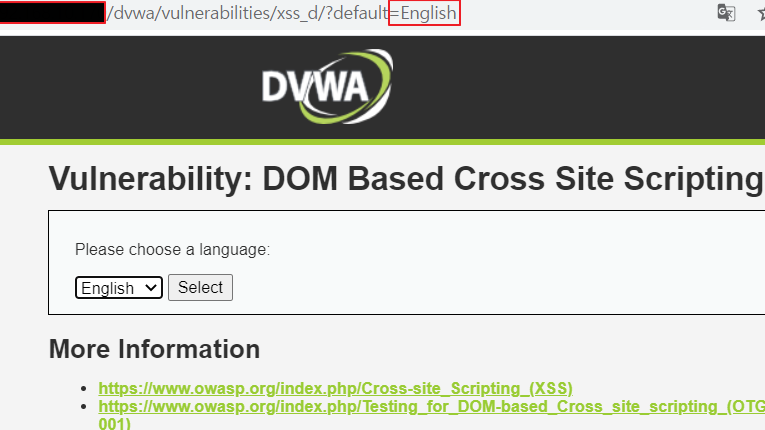
[그림 1 - 13] XSS Reflected 공격

**[Step #2] XSS Stored 공격이 가능한 것을 확인**



[그림 1 - 13] XSS Stored 공격

**[Step #3] DOM(Document Object Model) 기반을 사용하는 것을 확인**



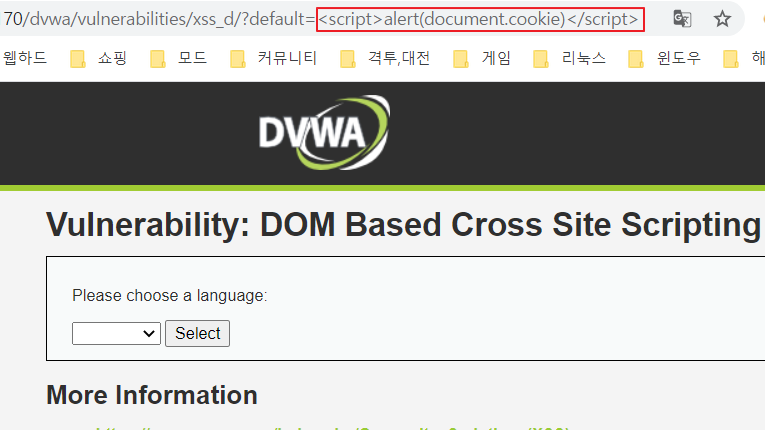
[그림 1 - 13] DOM 기반 확인1

**[Step #4] DOM(Document Object Model) 기반을 사용하는 것을 확인2**



[그림 1 - 13] DOM 기반 확인2

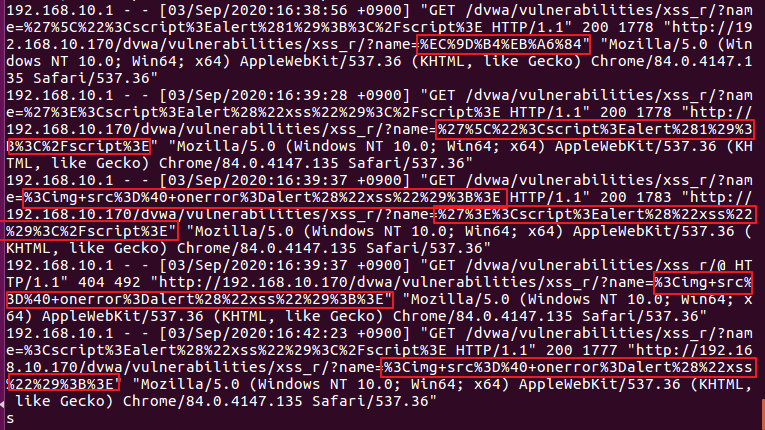
**[Step #4] DOM(Document Object Model) 기반의 XSS 공격**



[그림 1 - 13] DOM 기반 XSS 공격

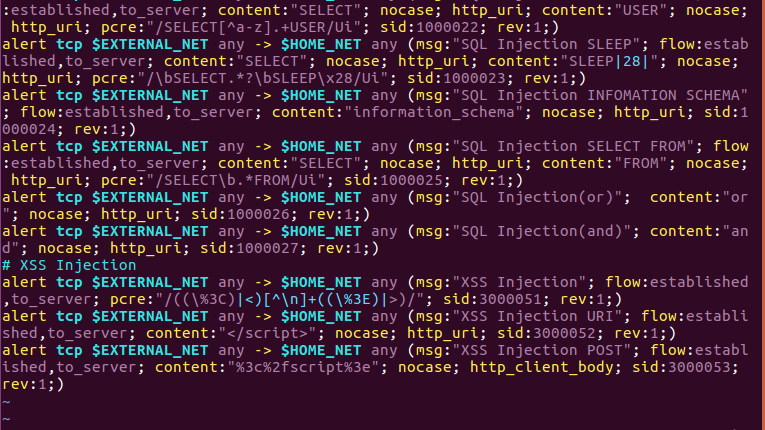
~~2.3.4 룰 세팅~~

**[Step #1] 웹서버 로그를 분석해 공격의 패턴을 확인함**



[그림 1 - 13] XSS Reflected 공격 로그 확인

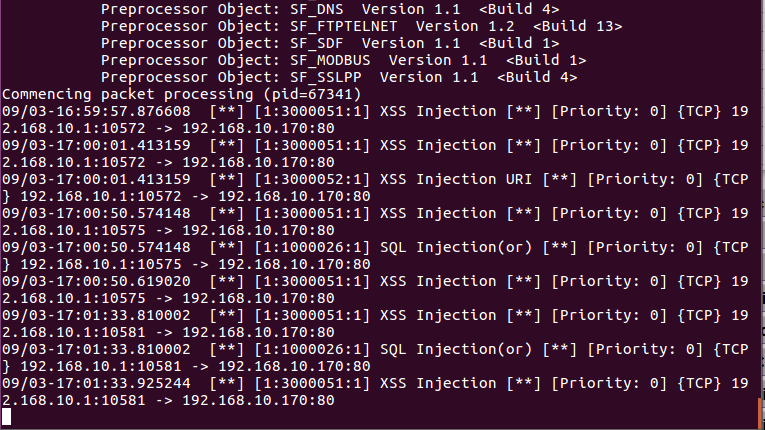
**[Step #2] 추가적인 알맞은 XSS 룰셋을 추가함**



[그림 1 - 13] SQL 룰 세팅

~~2.3.4 탐지 확인~~

**[Step #1] 웹서버 로그를 분석해 공격의 패턴을 확인함**



[그림 1 - 13] 공격 로그 확인

# 결론

IDS 시스템은 다양한

람이 있다.

# 추후과제 및 후기

이번 IDS서버 구축

# 참고자료

[1] Snort