

* **K-사이버 시큐리티 웹로그 분석 -**

|  |  |
| --- | --- |
| **소 속** | 서울과학기술대학교 |
| **팀 명** | 토익커 |
| **팀 원** | 김지환, 고병산 |
| **제출 일자** |  |

**목 차**

1. **서 론**

1.1. 분석의 목적

1.2. 공격 시나리오 종류

1. **분석 환경**

2.1. 분석 시스템

2.1.1. H/W 정보

2.1.2. 분석도구

1. **분석절차**3.1. 현상분석

3.2. 데이터 시각화

3.3. 시나리오 추측

3.4. 시나리오 검증

1. **분석결과**

4.1. Crawling

4.1.1. DirBuster

4.1.2. Web Scan

4.1.3. Directory traversal

4.1.4. Crawling

4.1.5. Scarping

4.2. Credential Stuffing

4.2.1. 권한 확인

4.2.2. User-Agent & Referer 변조

4.2.3. IP 변조

4.3. 기타 이상로그

4.3.1. XML-RPC Error Log

4.3.2. is-index

4.3.3. stderr

4.3.4. fastcgi

4.3.5. Referer

1. **발전방향**

**<붙임> 참고문헌**

1. **서론**

#### 1.1 분석의 목적

2020년, 애플리케이션 보안은 가장 중요한 보안요소중 하나로 손꼽히고 있다. 본 분석보고서는 웹로그 빅데이터 분석을 통한 웹스크래핑, 크리덴셜 스터핑 등 이상행위 탐지방법에 대한 연구에 목적이 있다.

#### 1.2 공격 시나리오 종류

데이터 셋에서 나타나는 공격시나리오를 크게 두가지로 나누어 정의했다.

Credential Stuffing

Credential Stuffing은 ID/PW쌍을 지속적으로 주입해 계정 권한 탈취를 시도하는 공격이다. 공격 시나리오는 아래 그림과 같다.

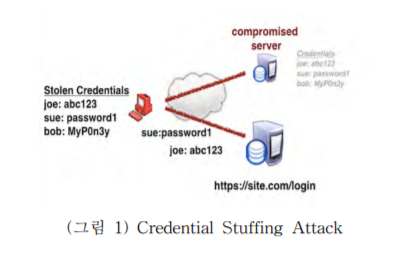


그림 Credential Stuffing 동작과정

1. 공격자가 취약한 서버로 침투해 저장되어있는 ID/PW 쌍을 탈취한다.
2. 얻어낸 ID/PW쌍으로 공격하려는 다른 서버에 로그인을 시도해 관리자 권한 탈취를 시도한다.

Crawling

웹 크롤링이란 하나 또는 더 많은 seed URL들을 가지고, 그들이 포함된 다른 하이퍼링크들을 찾아내는 것을 반복하며 하이퍼링크들을 다운로드 하는 행위이다.[[1]](#footnote-1)

본 보고서에는 이런 웹크롤링을 조금 더 확장하여, Bot에 의한 웹 탐험, 분석하는 행위들을 크롤링이라 정의하였다.

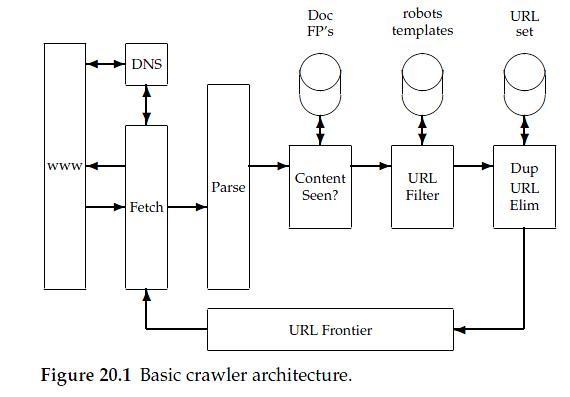


그림 Crawling 동작과정

#### 분석환경

#### 분석 시스템

#### H/W 장비

|  |  |  |
| --- | --- | --- |
| **분류** | **하드웨어 정보** | **OS 정보** |
| 데이터 분석/개발 | Intel Core i7-8565U CPU @ 1.80GHz, Memory 32GB | Debian Linux 64bit |
| 테스트 환경 | Intel Core i7-3770 CPU @ 3.4GHz, Memory 8GB | Windows 10 Ent 64bit |

**표 1. 분석에 사용된 H/W 장비**

#### S/W 정보

|  |  |  |
| --- | --- | --- |
| **Tool** | **Version** | **비고** |
| WordPress | Version 5.5.1 | 테스트 환경 |
| PHP | Version 7.2.10 | 테스트 환경 |
| DB | mysql 10.3.8-MariaDB | 테스트 환경 |
| Python | Python 3.8.5 | IDE Pycharm |
| grep | grep (GNU grep) 3.4 |  |
| GoAccess | GoAccess - 1.4. |  |
| ElasticSearch | Version 7.9.2 |  |
| Kibana | Version 7.9.2 |  |
| Microsoft Excel | Excel Enterprise 20009 |  |
| apache-scalp | Version 0.3 |  |

**표 2. 분석에 사용된 S/W 정보**

#### 3. 분석절차

데이터셋을 분석하고 가설을 세우기 위해 탑다운 방식으로 4단계의 연역방식을 거쳤다.  
1. 전체 로그를 대상으로 현상 탐구  
2. 그래프로 시각화된 통계 데이터에서 특이점을 찾고 공격 시나리오 가설 설정. [GoAccess, Kibana]  
3. 전체 로그에서 의심되는 일부 표본을 추출해 가설 비교 [Excel과 리눅스 Grep]  
4. 전체 로그를 대상으로 가설 검증 [Python]

#### 3.1 현상 탐구

배포받은 받은 데이터셋을 분석해 피처를 파악하고 특징을 분석하는 단계이다. 데이터셋의 크기가 너무 커 문서 편집기의 한계범위를 초과하는 이슈가 있었고, 문서 열람을 위해 문서를 분할하는 Python 코드를 작성하는 단계를 거쳤다.

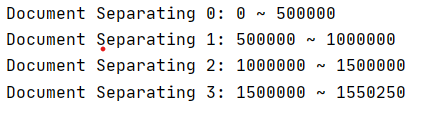
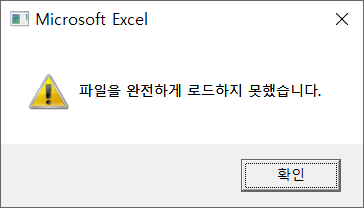


그림 Excel 에러 화면 (좌)

그림 데이터셋 분할에 쓰인 사용자 (우)

에러를 포함한 1,550,250개의 행 데이터가 있었으며, Excel은 1,048,576이 넘는 행을 표현하지 못하는 관계로 500,000 개씩 4개의 파일로 분할해주었다.

데이터셋은 클라이언트가 서버에 접속했을 때 미들웨어인 nginix에 의해 저장된 서버 로그였으며, 각 피처가 가지는 도메인과 의미를 분석하였다.

① Timestamp  
ngnix에 접속한 시간을 나타낸다. 호스트별로 접속하는 주기와 빈도를 살펴보았다.

|  |  |
| --- | --- |
| 이상행위 의심패턴 | 예상되는 공격 |
| 같은 시각에 동시다발적으로 접속하는 경우 | Bot에 의해자동화된 공격 |
| 같은 시간차를 두고 주기적으로 접근하는 경우 | Time-interval을 둔 우회기법 |

아래 그림처럼 짧은 시간 내에 접속량이 비정상적으로 상승하는 경우, (기계적인) 접근으로 의심하였다.



그림 비정상적인 TimeStamp

② Method

웹페이지 요청시 사용된 REST API 유형을 나타낸다.

|  |  |
| --- | --- |
| 도메인 | 내용 |
| GET | 리소스 요청시 파라미터를 URL뒤에 붙여 전달하는 방식. |
| POST | 리소스 요청시 파라미터를 Payload를 통해 전달하는 방식. |
| DELETE | 요청한 리소스를 삭제요청하는 Method. |

③ Protocol  
서버와 통신할 때 사용하는 프로토콜 유형이다.

④ Status

|  |  |
| --- | --- |
| 도메인 | 내용 |
| 200 | OK 리소스를 성공적으로 반환받음. |
| 206 | Partial Content  클라이언트에서 복수의 스트림을 분할 다운로드를 하고자 범위 헤더를 전송함. |
| 301 | Moved Permanently  요청한 리소스의 URL가 영구적으로 변경되었다는 의미이며 바뀐 URL을 반환함. |
| 302 | Found  요청한 리소스의 URL이 일시적으로 변경되었다는 의미이며 바뀌기 이전의 URL을 반환함. |
| 304 | Not Modified 요청한 리소스를 전송할 필요가 없음. |
| 400 | Bad Request 잘못된 문법의 요청으로 서버가 요청을 이해할 수 없음. |
| 403 | Forbidden 클라이언트가 권한이 없는 리소스에 접근시도했음. |
| 404 | Not Found 존재하지 않는 리소스에 접근함. |
| 405 | Method Not Allowed 서버에서 지원하지 않는 Method로 접근함 |
| 499 | Client Closed Request  ngnix에서만 제공하는 에러코드. 서버에서 요청 처리중에 클라이언트가 연결을 끊음. |
| 500 | Internal Server Error |

⑤ Referer  
방문자가 Path로 접속하기 전의 유입경로

|  |  |
| --- | --- |
| Referer가 있는 경우 | 브라우저 내의 웹페이지 링크를 클릭해 이동 |
| Referer가 없는 경우 | 자바스크립트를 통한 웹페이지 이동  북마크를 통한 웹페이지 이동 |

⑥ Path  
클라이언트가 요청한 리소스의 경로

⑦ Host  
 서버에 접속한 클라이언트의 IP주소.

⑧ UA  
 유저가 접속을 시도한 브라우저에 관한 정보를 가지는 피처. 변조가 쉽기 때문에 UA를 기반으로 한 통계는 신뢰하기 힘들다.

⑨ Payload  
 POST방식으로 접속 요청을 할 때 메시지바디에 담기는 파라미터 정보.

⑩ Bytes  
 Host에게 반환하는 데이터의 Bytes 수. WordPress는 정적 웹페이지가 아닌 동적 웹페이지를 제공하기 때문에 같은 메소드로 같은 경로에 접근해도 조금의 bytes차이가 있을 수 있다.

서버 로그 데이터의 피처 대부분은 클라이언트로부터 받아온 정보를 기반으로 작성되기 때문에 Zap, BurpSuite와 같은 프록시 툴을 사용하면 얼마든지 변조가 가능하다. 하지만 [UA, Referer]를 제외한 다른 피처들은 데이터를 주고받는데 영향을 끼치게 되기 때문에 신뢰할 수 있다는 가정을 바탕으로 공격 시나리오를 작성한다. [UA, Referer]는 변조되어도 통신에 영향이 없기 때문에 신뢰하기 어렵다.

#### 3.2 가설 설정 및 시나리오 추측

시나리오를 작성하기 전에 시각화된 데이터를 바탕으로 가설을 세웠다. 시각화를 위해 데이터 시각화 오픈소스인 GoAccess와 ElasticSearch와 Kibana를 이용했다.

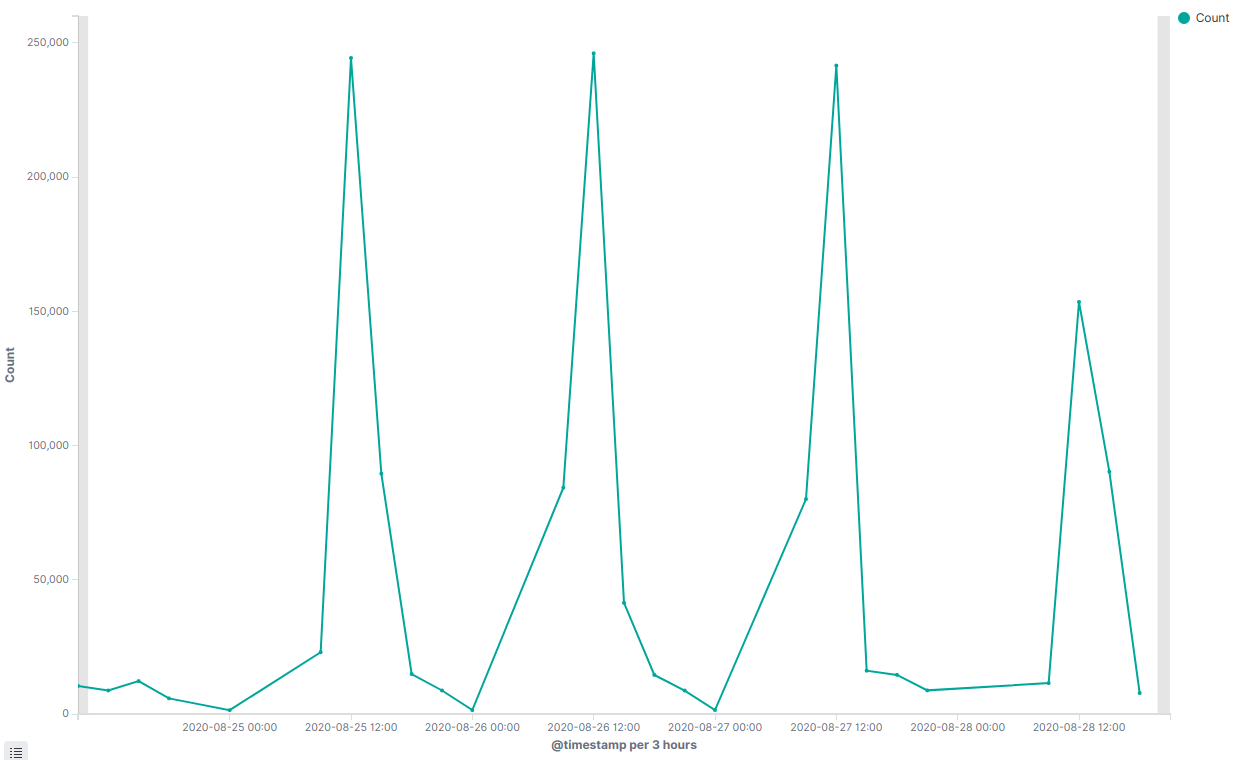


그림 전체 데이터 시간별 접속 그래프

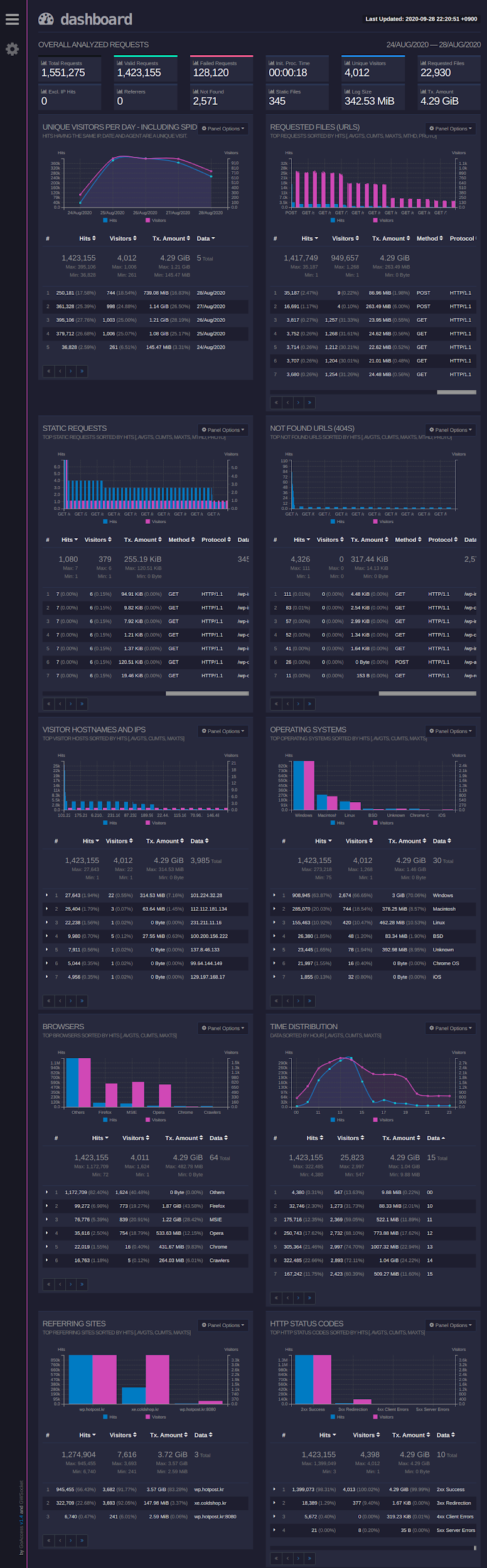


그림 GoAccess 분석 화면

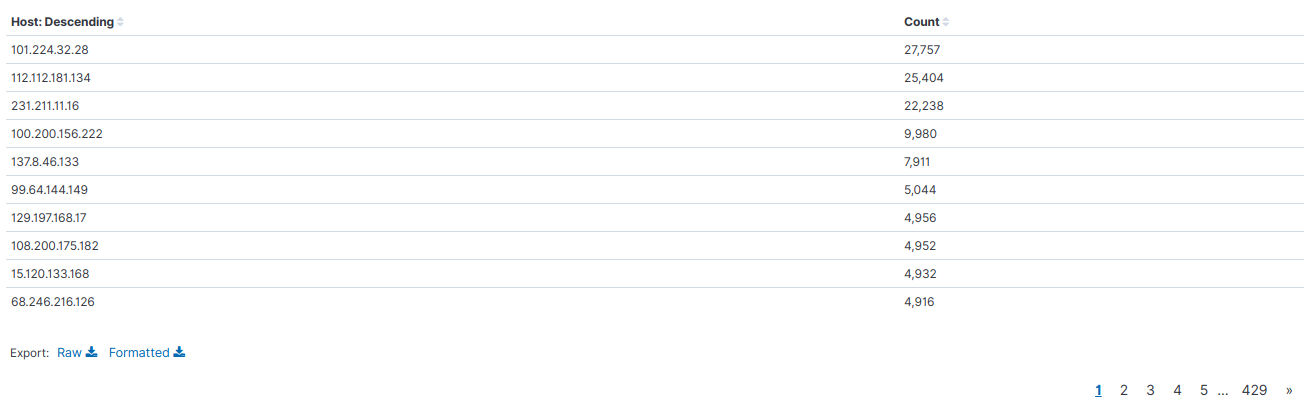
****

그림 Host별 접속 횟수

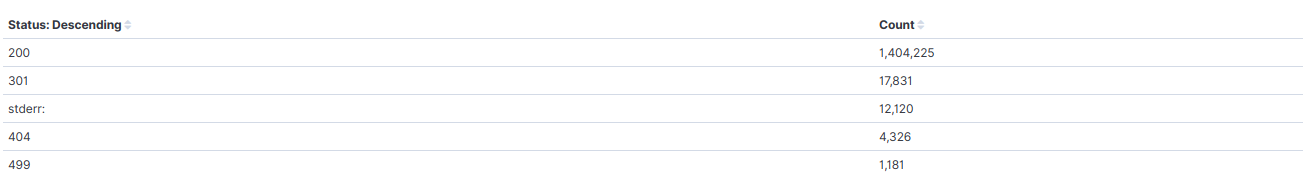


그림 상태코드별 발생 횟수

Kibana로 호스트별 로그들을 분석해, 평균보다 수치가 높은 데이터들을 추출했고, 이를 토대로 특징들을 파악해 시나리오 가설을 세웠다. 이후, 크롤링과 크리덴셜 스터핑의 특징들을 토대로 여러 가설들을 세운 후 호스트별 로그들을 살펴보며 가설들을 확립시켰다.

#### 3.4 시나리오 검증

시나리오가 적용되는지 확인해보기 위해 전체 데이터셋을 대상으로 분석하는 파이썬 코드를 작성하였다.

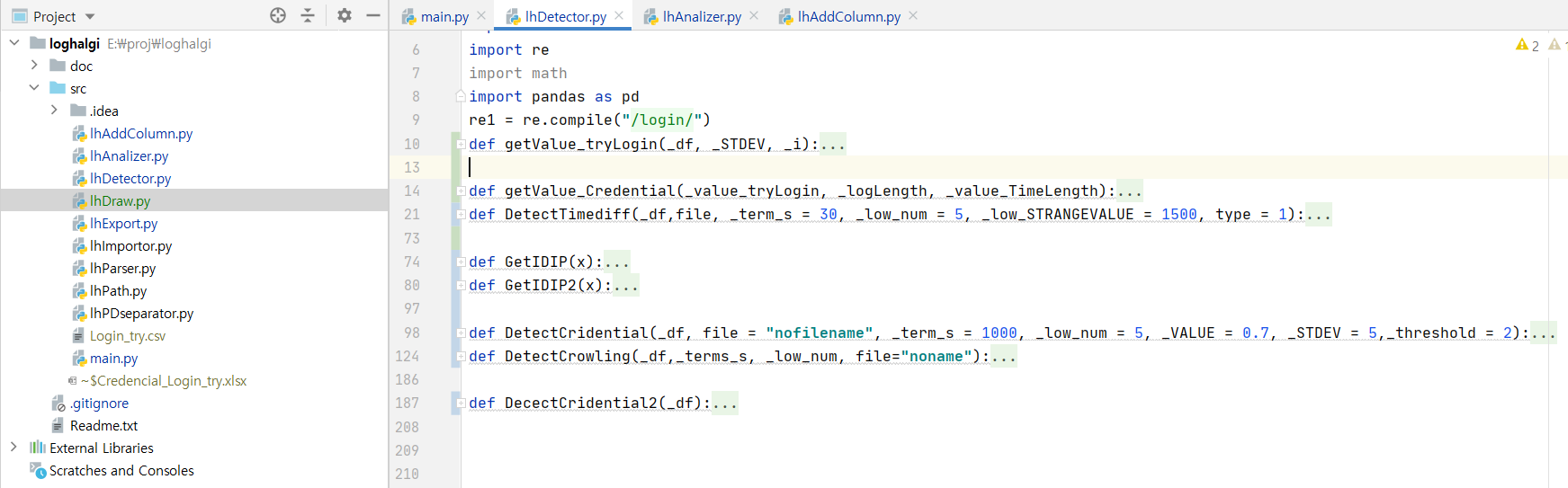


그림 시나리오 검증 코드(일부)

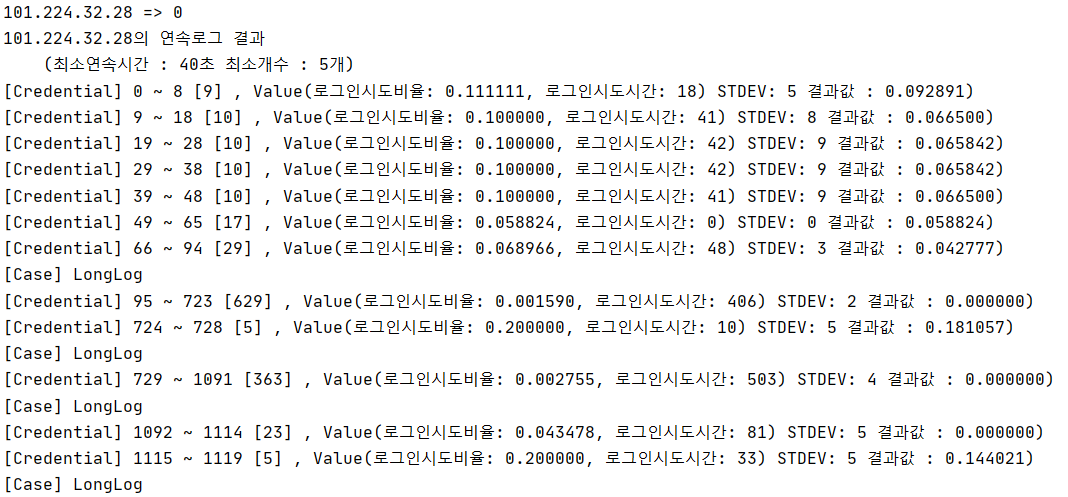
****

그림 시나리오 검증 결과 예시 (Credential Stuffing 예시)

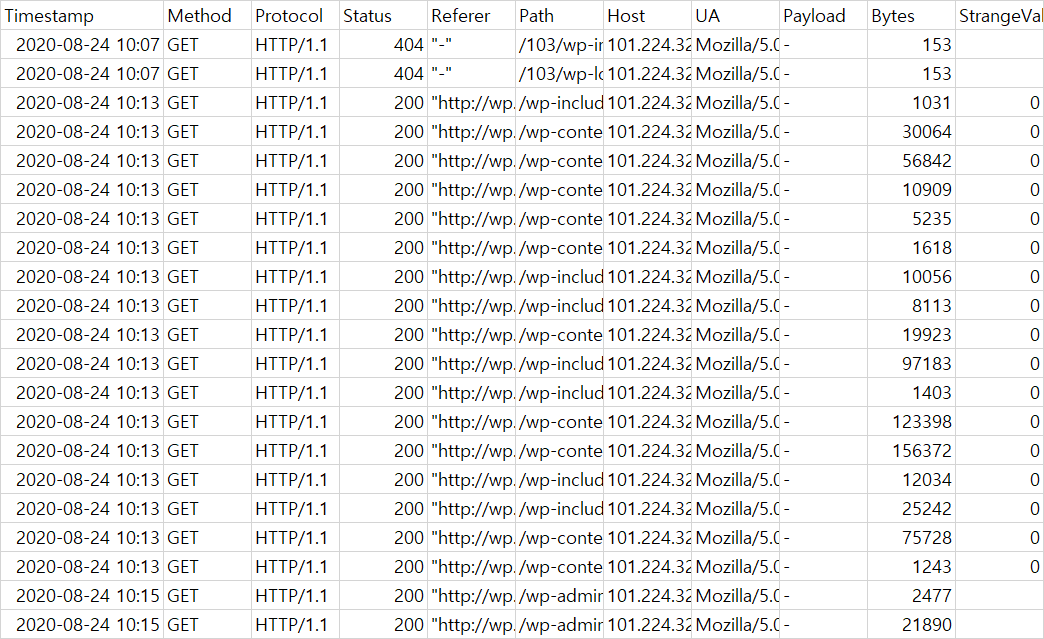


그림 시나리오 검증 결과 예시 (Crawling 예시)

#### 4. 분석결과

전체 Host : 4065

로그 : 1,537,008 Line (전체 로그는 1,550,250 이지만, 에러로그를 제외한 값)

|  |  |
| --- | --- |
| 상태코드 | 개수 |
| **200** | **1,511,769** |
| 206 | 24 |
| 301 | 17,831 |
| 302 | 1,583 |
| 304 | 2 |
| 400 | 14 |
| **403** | **148** |
| **404** | **4,326** |
| 405 | 3 |
| **499** | **1,287** |
| 500 | 21 |

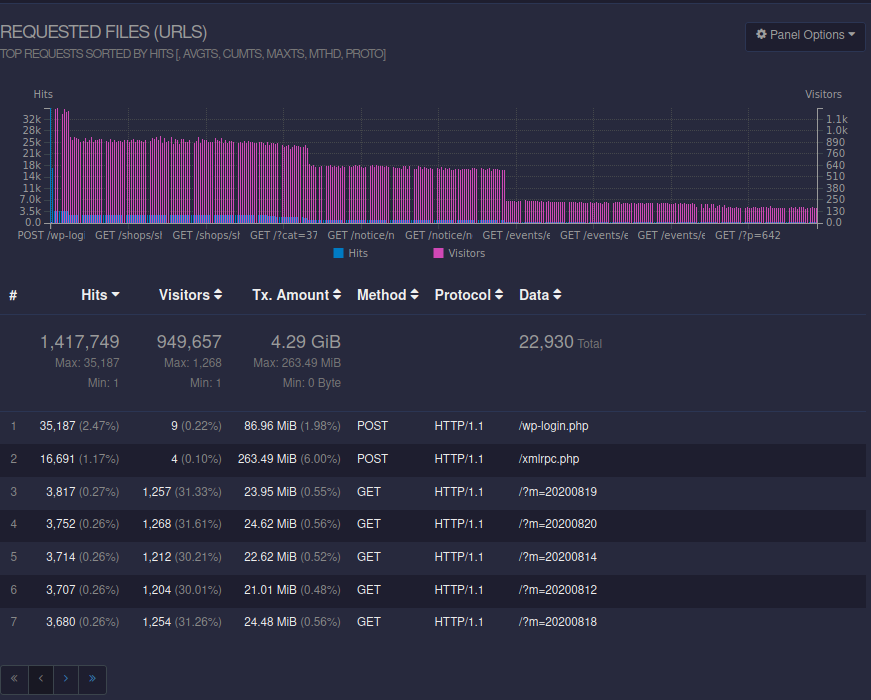


그림 GoAccess 中 자주 방문한 페이지

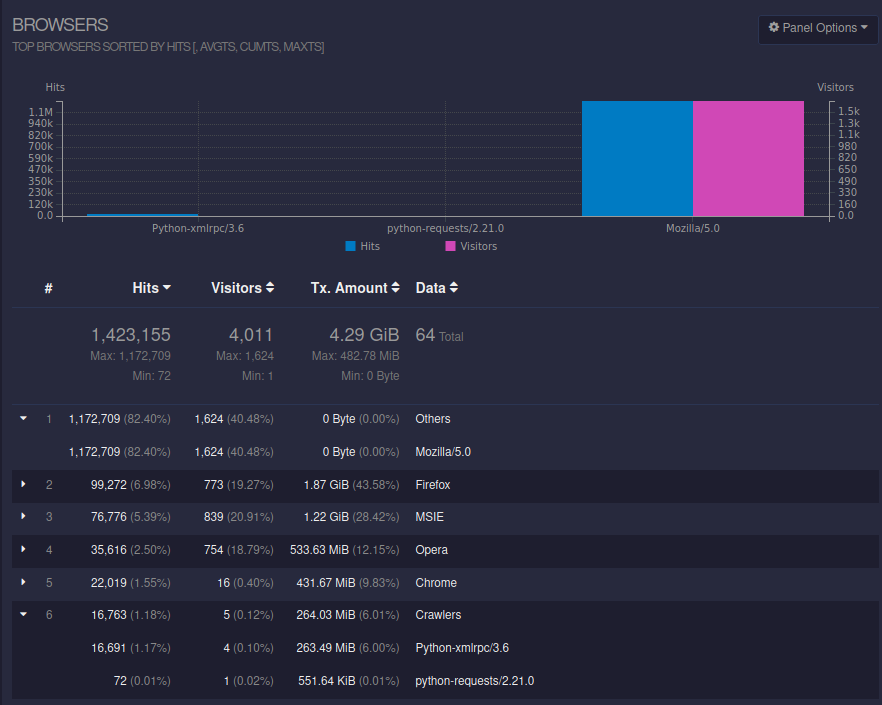


그림 GoAccess 화면 中 User-Agent 분석

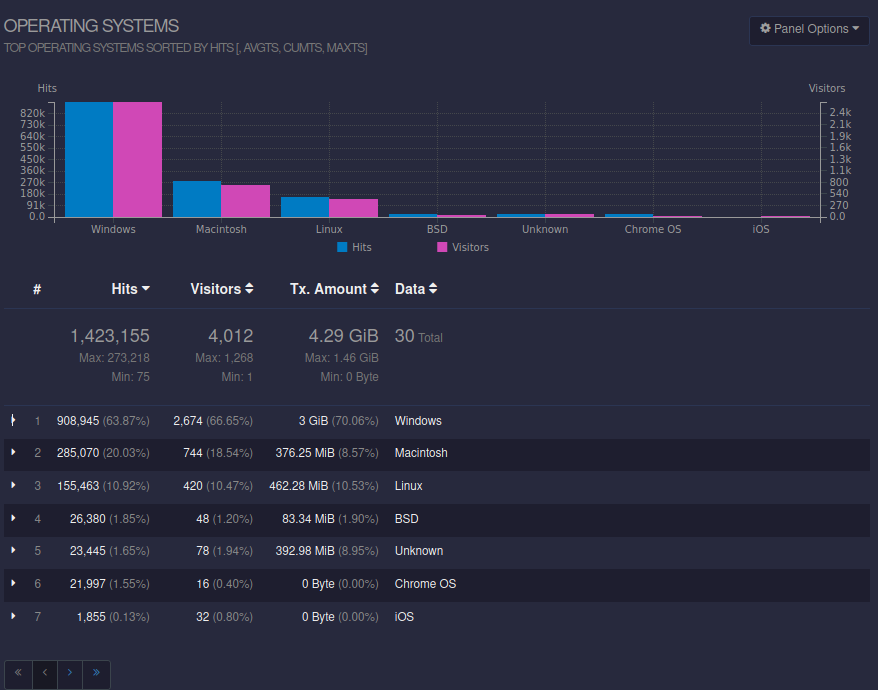


그림 GoAccess 화면 中 OS 분석

#### 4.1 Crawling

##### 4.1.1 DirBuster

개념 : 웹서버 디렉토리 구조를 파악하거나 숨겨진 중요파일들을 찾아내기 위해 웹/응용 프로그램 서버에서 디렉토리와 파일 이름을 무차별적으로 대입하는 공격이다.

IP : 101.224.32.28

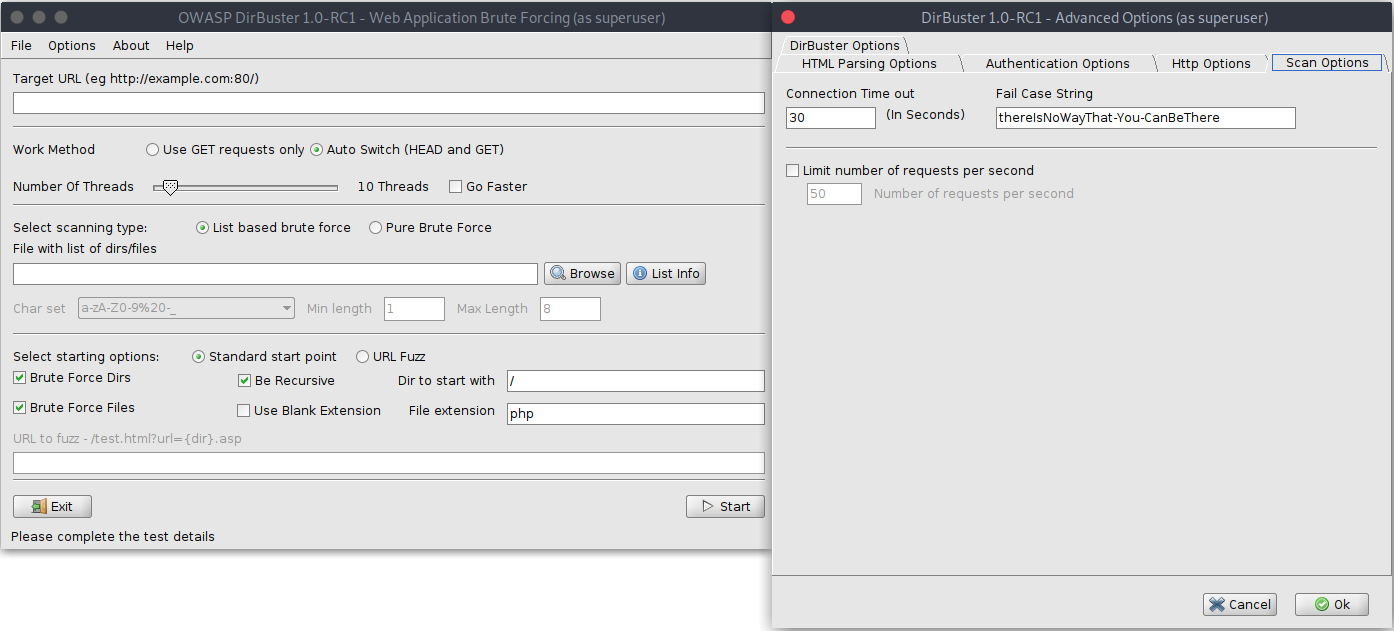


그림 OWASP DirBuster 실행화면



그림 DirBuster 실행 중 is-index 에러로그

그림 DirBuster Default String

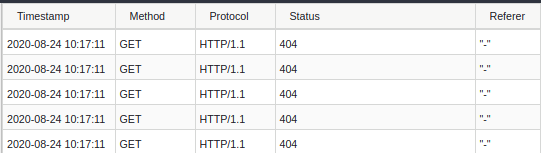


그림 Referer값이 없는 요청

시나리오 설명 :

1. Fail CaseD에 대한 반응을 보기 위해 Fail Case String(thereisNoWayThat-You-CanBeThere)을 입력하여 여러 디렉토리에 테스트해 본다.
2. 실패와 성공의 반환값에 따라 정해둔 Directory list를 가지고 무차별 대입을 한다.

※ DirBuster의 Default User Agent는 변경되었지만, Fail Case String이 변경되지 않아 그대로 “thereisNoWayThat-You-CanBeThere”이 나타난다.

※ DirBuster의 시도마다 User-Agent가 바뀌는 것을 볼 수 있는데, 프로그램 실행 시 랜덤 User-Agent로 설정한다는 것을 알 수 있다.

※ 비정상적인 수의 요청을 보내는 것을 볼 수 있다.

##### 4.1.2 Web Scan

시나리오 : 본래 있던 Web Scan은 웹 페이지의 취약점 혹은 구조 등을 파악하는 개념이지만, 여기서는 Bot에 의한 페이지스캔 개념으로 사용되었다.

IP : ~~~

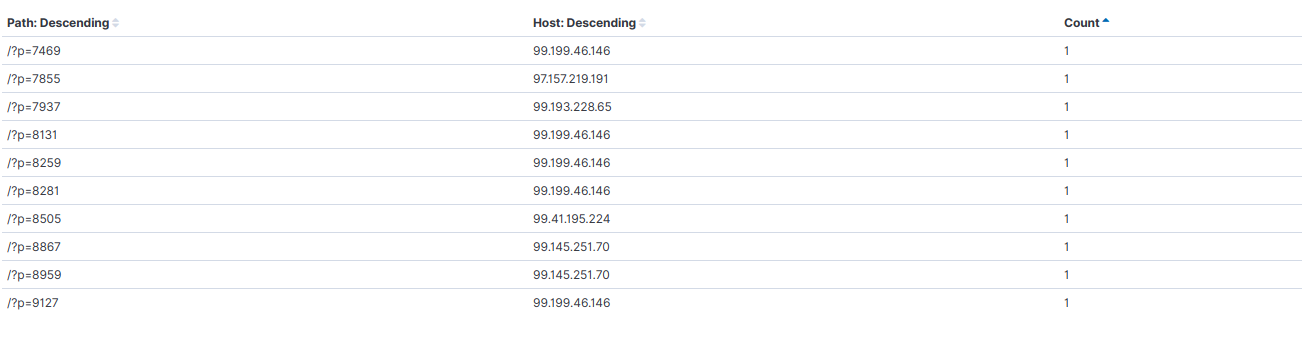


그림 방문횟수가 1인 페이지

Path별 통계를 내었을 때, 웹 스캔의 경우 다른 사람이 안들어가는 Path에 접속 시도하는것을 볼 수 있었다.

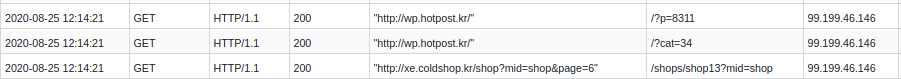


그림 비정상적 Time-Interval

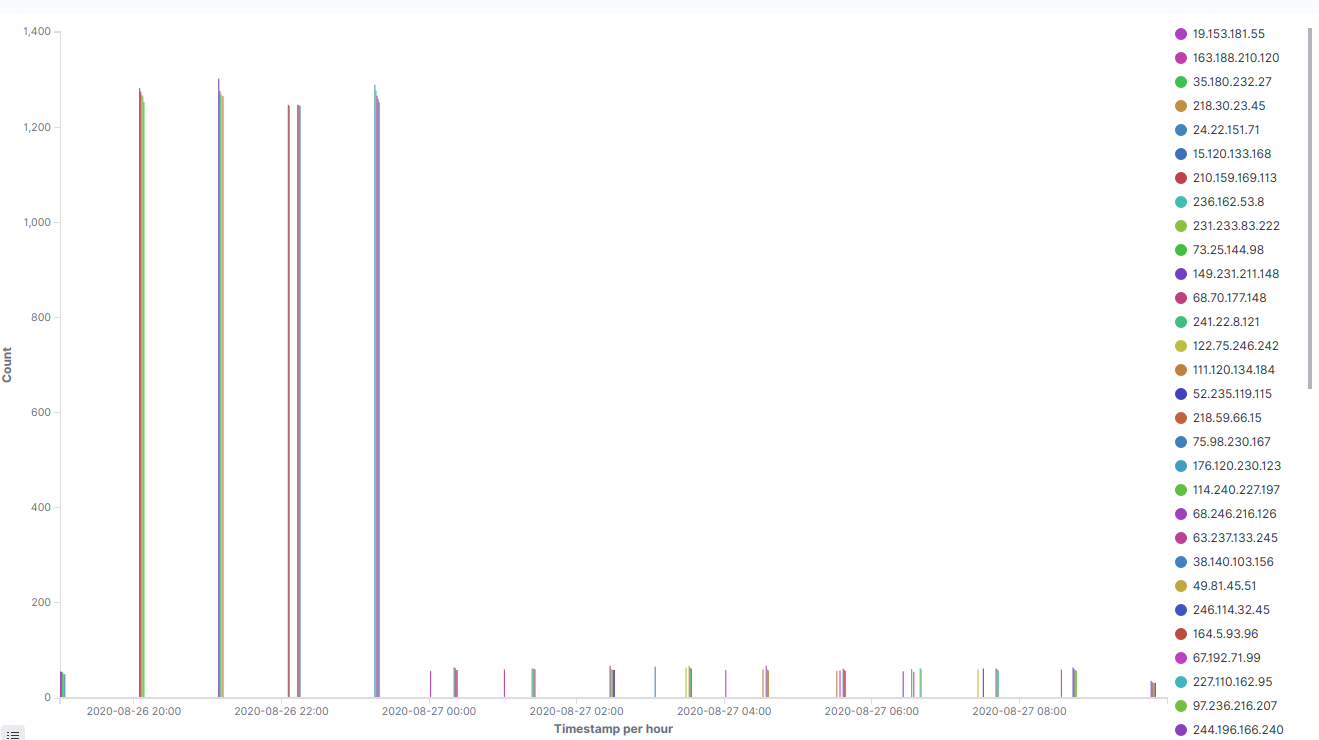


그림 정상 호스트와 이상호스트의 시간별 접속량 비교 (좌 이상, 우 정상)

짧은 시간에 여러번 요청하는 시도가 있다. 다른 DirBuster나 LFI에 비하면 분당 요청회수는 적지만, 해당 공격은 긴 시간동안 꾸준히 요청하는 특징을 가진다.

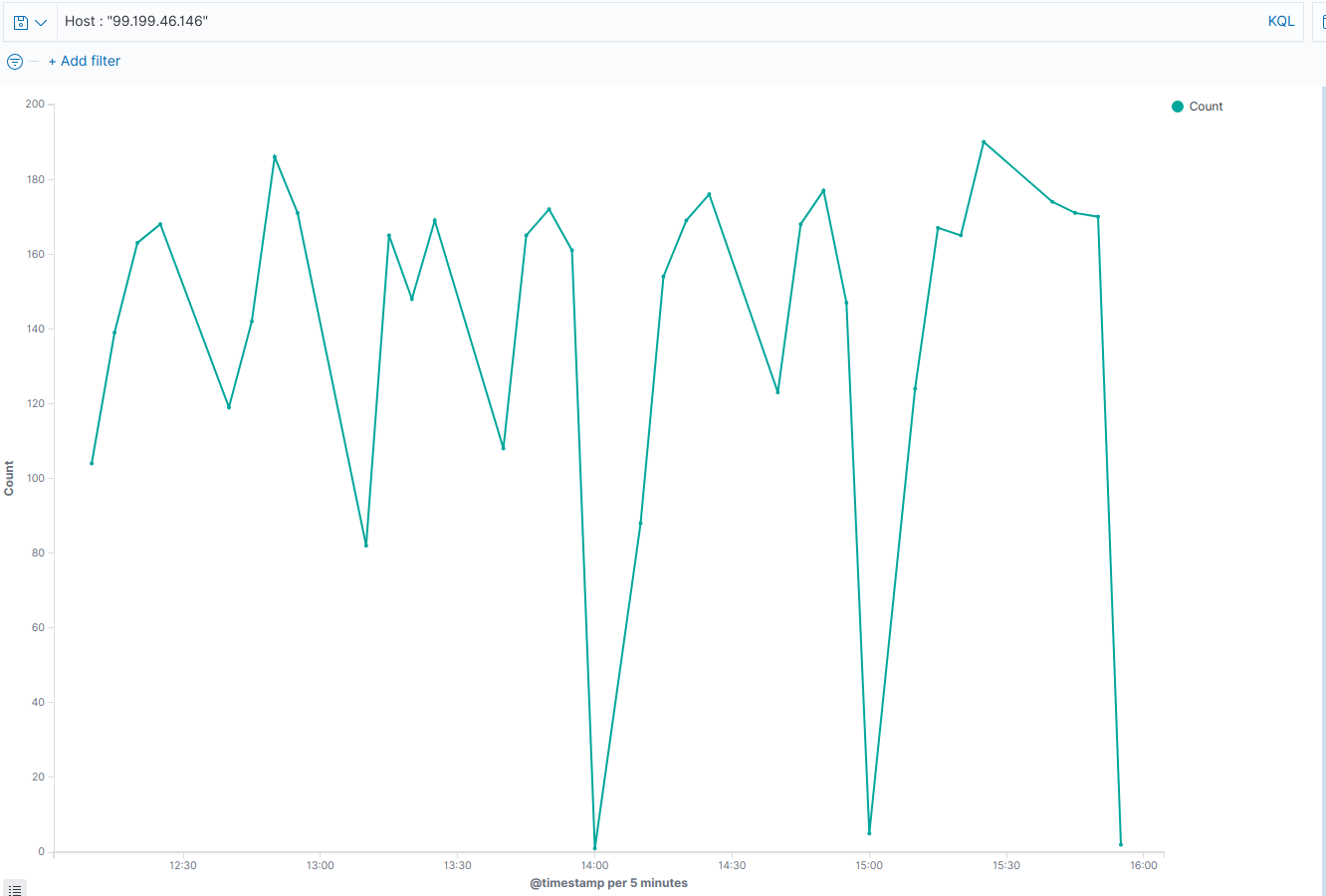


그림 Web Scan의 공격유형 (꾸준한 접속량)

위와 같이 12시 정도부터 16시까지 꾸준한 활동을 하며, 해당 시간에 비슷한 행동을 보이는 호스트들이 여러개 있는 것을 찾을 수 있다.

시나리오 설명 :

1. 특정 시간(12시와 3시)경 웹페이지에 접속한다
2. 같은 페이지를 접속하기보다는 다양한 페이지를 돌아다니며 글을 스크랩해간다.
3. 4시간 정도 활동 후 사라진다.

※ 1초 안에 3~4번의 접속을 시도할 때가 있고, 4시간 동안 3000번이상의 시도를 보이며, 그중 2500여개 이상의 경로를 탐색한다.

※ Bot에 의한 요청임을 숨기기 위해 User-Agent와 Referer이 지속적으로 변경하는 것을 볼 수 있다.

##### 4.1.3 Directory traversal

시나리오 : 시스템에서 허용한 디렉터리보다 상위 디렉터리의 파일을 요청하거나, 혹은 시스템 중요파일 호출하는 등의 공격이다.

IP : 101.224.32.28, 231.211.11.16

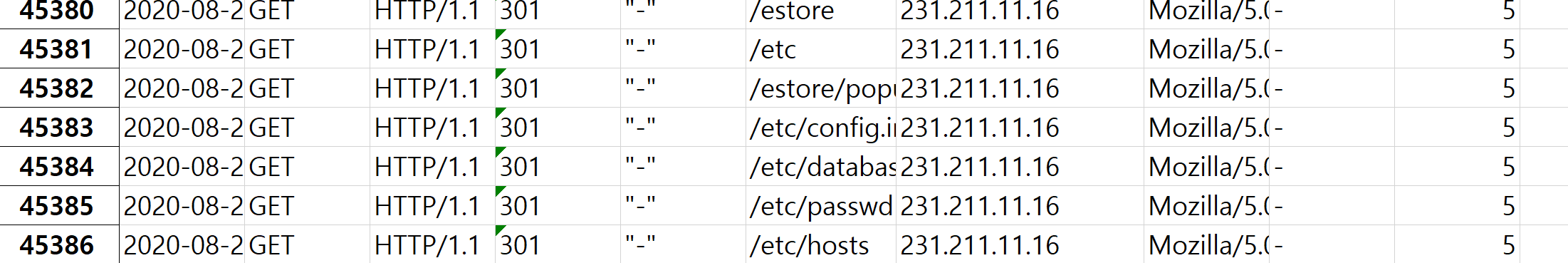


그림 시스템파일 접근 시도



그림 아스키코드를 이용한 디렉토리 탐색 문자열

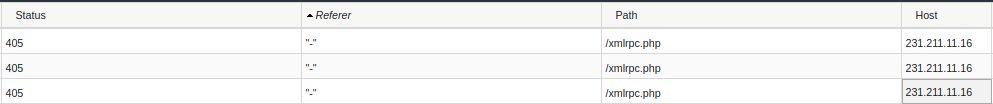


그림 상대코드 405

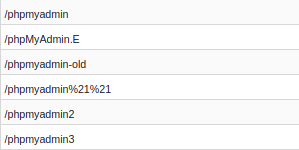


그림 사전파일 대입 예시화면

시나리오 설명 :

1. 중요 시스템 파일에 대한 접근을 시도한다.
2. %20../, %20.. 등 상위디렉토리에 대한 이동을 시도한다.
3. c99.php 와 같은 쉘 파일들에 대한 추측성 시도를 한다.

※ 405 Status Code가 반환되는 경우가 있다. 이는 서버에서 지원하지 않는 메소드를 시도하여 발생한 코드이다.

※ 403 Forbidden과 404 Not Found가 발생한다.

##### 4.1.4 Crawling

시나리오 : 웹 페이지 전체를 돌아다니며 홈페이지의 구조 및 내용을 다운받는 공격으로 일반적으로 Bot들이 사용된다.

IP : 137.8.46.133

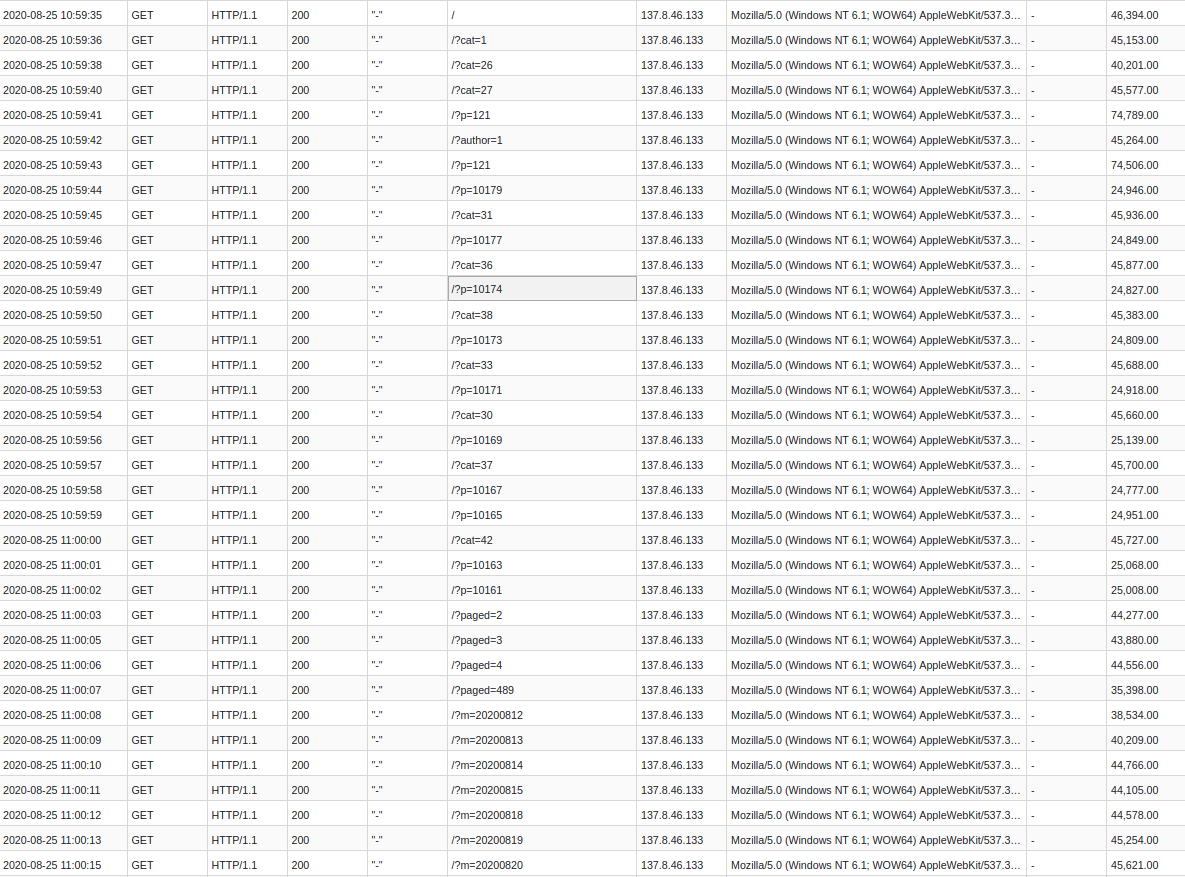


그림 동일한 TimeInterval과 Referer 이상

시나리오 설명:

1. 여러 경로의 Path를 동일한 시간간격으로 요청한다.
2. 계속해서 Referer가 없는 모습을 볼 수 있다.

※ UA에 정상적인 값을 입력하여 정상 호스트처럼 보이게 하였다.

※ 2.x초의 간격을 두고 실행하는 것을 알 수 있다.

※ /?p=1, /?p=2 등 모든 경로를 탐색하려는 것을 볼 수 있다.

##### 4.1.5 Scarping

시나리오 : 일반적인 사용자가 아닌 Bot이나 Program을 이용하여 데이터를 주기적으로 가져가는 행위

IP : 알 수 없음



그림 IP 없는 로그들의 UA별 접속경로



그림 Host 없는 로그의 UA별 그래프

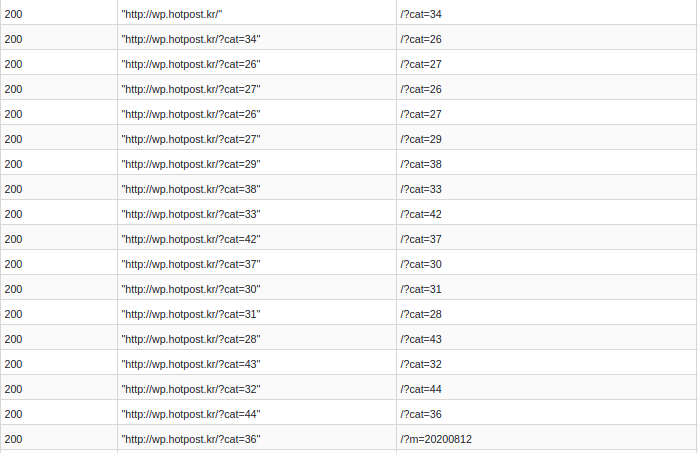


그림 Host 없는 로그의 Referer와 Path의 일치

시나리오 설명 :

1. Wget과 같은 프로그램을 사용하여 페이지를 다운받는다.
2. 크게 5가지의 UA로 구분되어, 스레드별 혹은 이용자별로 동일시간에 다른 Page를 다운받는다.
3. UA와 Referer이 동일한 것을 볼 수 있는데, Selenium과 같은 API를 사용하여 사용자처럼 보이게 하였다.

※ 그림21을 보면 알 수 있듯이, 3가지의 UA가 동일시간 다른경로를 요청하는 것을 볼 수 있었다.

#### 4.2 Credential Stuffing

Credential Stuffing의 공통적인 시나리오는

1. 반복적으로 POST방식으로 로그인 요청한다. 성공시 로그인 화면으로 리다이렉트되고 실패시 에러 창을 팝업한다. 리다이렉트 메세지는 5Bytes, 에러 창은 약 3000 이상의 Bytes를 차지한다.  
   

그림 로그인 시도했을때 실패(상) 성공(하)

1. 로그인에 성공하면 /wp-admin/profile.php 에 접속해 권한을 탈취한다.  
     
   

그림 로그인에 성공한 횟수(노란색과) 탈취한 계정으로 /wp-admin/profile에 접근한 횟수(빨간색)

서버 접속시도를 너무 많이해서 나타나는 Status 499 로그(8개), profile에 접근하는 Status 200 로그(246개)의 합이 노란색과 같다.

데이터셋에서 나타난 Credential Stuffing의 특징은  
 1. Referer가 정상적으로 작성되지 않는다.

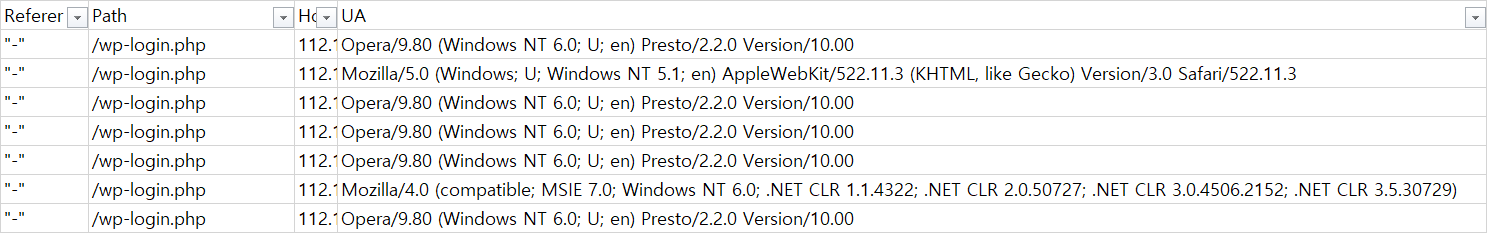


그림 Referer 이상 로그

2. 약 60번 로그인을 시도한 후 1분간 요청을 하지않는다.

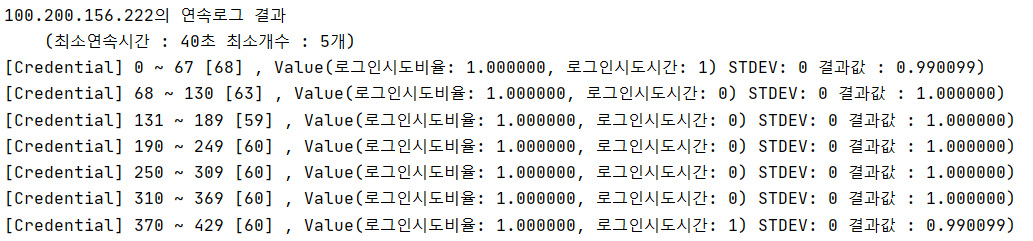


그림 60번씩 로그인 시도를 한 후 1분 쉬는 루틴을 반복하는 100.200.156.222 분석결과 화면

Credential Stuffing 침입 시도에서 두가지 우회기법이 발견되었다.  
1. UA 변조기법

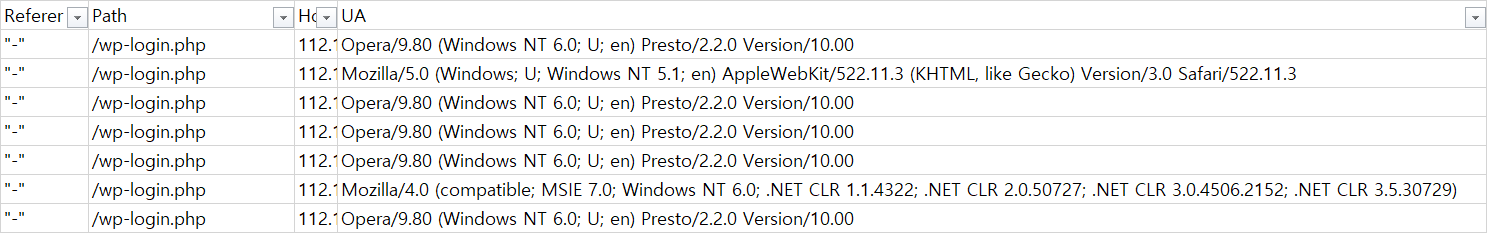


그림 랜덤 UA를 주입하는 우회기법

2. IP 변조기법

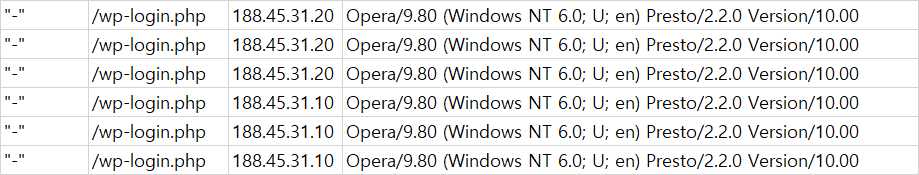


그림 IP 변조 로그

시나리오 설명:

1. 60회 로그인 시도를 반복한다
2. 로그인에 성공하면 /wp-admin/profile.php에 접속해 계정의 권한을 확인한다.

##### 4.2.1 User-Agent & Referer 변조

일반 Credential Stuffing의 파생형으로, 보안장비에 의한 차단을 피하기 위해 지속적으로 User Agent와 Referer를 변형하며 로그인을 시도한다.

##### 4.2.2 IP 변조

시나리오 : 여러개의 IP를 가지고 Credential Stuffing을 한다.

IP : (188.45.31.10, 188.45.31.20, 188.45.31.30, 188.45.31.40), (14.135.56.110, 14.135.56.120, 14.135.56.130, 14.135.56.140)

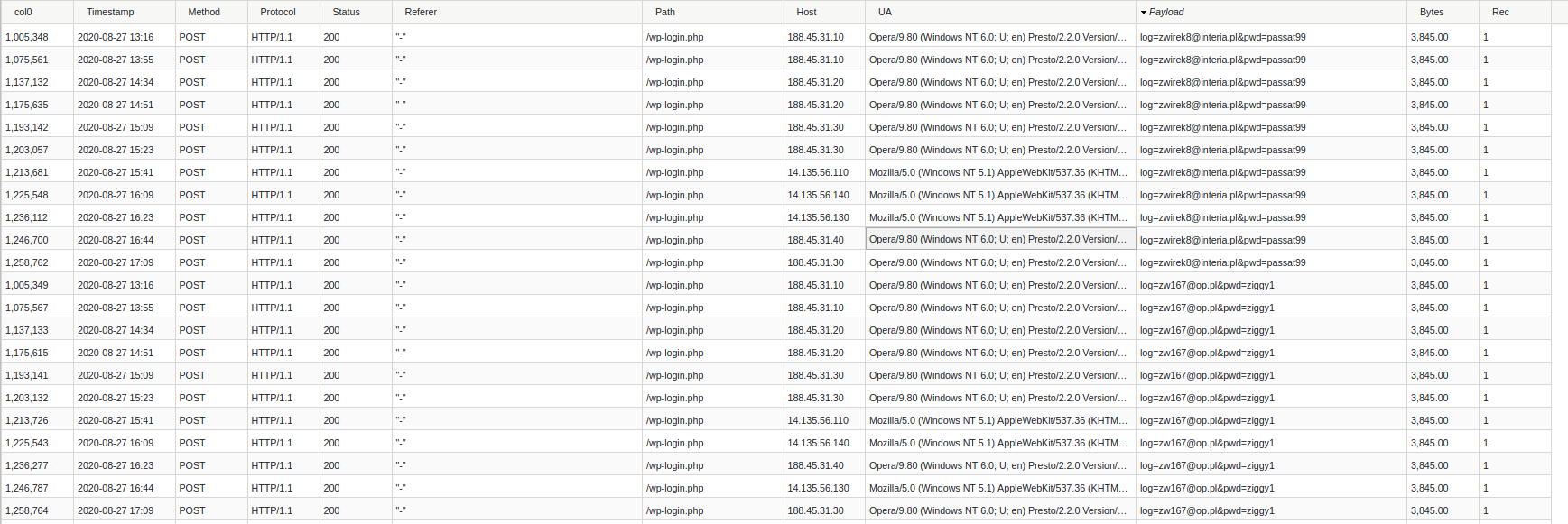


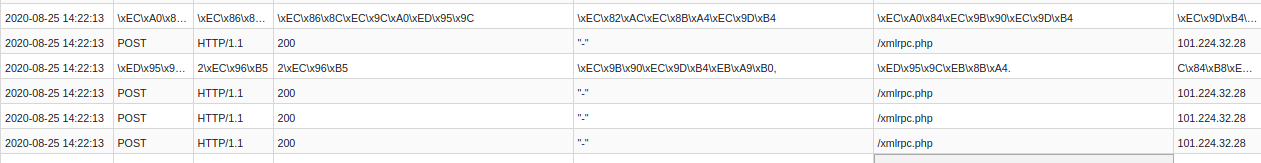
그림 IP변조 시나리오 로그

시나리오 설명 :

1. 한 IP로 로그인 시도를 한다.
2. 30여분 뒤 다시한번 같은 계정에 로그인 시도를 한다.
3. 1시간정도 지나면, 다른 IP로 변조하여 다시 시도한다.

#### 4.3 기타 이상로그

##### 4.3.1 XML-RPC Error Log



<그림 23> XML-RPC 에러로그

위와 같이 웹 로그에 이상데이터들이 있다. 유니코드로 이루어진 글들이며 바로 직전 xmlrpc로 보낸 데이터의 끝이

“</methodCall>\x0A”로 끝나지 않을 경우 해당 로그가 생기는 것을 확인하였다.

##### 4.3.2 is-index



<그림 24> DirBuster 실행 중 is-index 에러로그

로그 중 index, , directory, is index라는 구조의 로그들이 몇몇 찍혀있는 것을 볼 수 있었다. 이는 DirBuster와 같이 무차별대입 중 디렉토리의 인덱스페이지를 요청할 때 찍히는 것을 볼 수 있었다.

##### 4.3.3 stderr log



<그림 25> PHP sent 에러로그

sent, stderr:, stderr:, “PHP, sent라는 구조의 로그들이 있다. theme 디렉토리를 직접 접근하여 생긴 500 Error에 대한 로그로 확인할 수 있었다.

##### 4.3.4 fastcgi



<그림 26> fastcgi 에러로그

fastcgi관련 에러로그로 추정된다. 일반적으로 (1) 해당 시간에 접속한 Referer에 대한 에러로그가 입력되는 것과 (2) Referer가 ?cat=Number의 형식인 경우에는 입력되지 않을 때가 있었다. 특정 상황에서 발생하는 에러로그로 추정되면 이에 대한 자세한 분석은 실제 서버로 몇 가지 테스트 해보아야 가능할 것으로 추정된다.

##### 4.3.5 Referer

모든 데이터들의 Referer가 이전 로그의 Path가 아니었다. 정상적인 경우는 아니지만, 실제 데이터가 아닌, 대회용으로 나온 분석데이터이기 때문이라고 감안하여 생각하였다.

#### 5. 발전방향

크리덴셜 스터핑과 크롤링의 종류와 특징을 분석해 가능한 여러 시나리오를 설정한 후, 데이터 시각화를 기반으로 데이터셋으로부터 의심스러운 부분들을 추려내 데이터셋에서 나타난 공격들의 유형을 파악할 수 있었다. 이후 본선에서 데이터셋을 기반으로 머신러닝을 학습해 공격 행위들을 탐지할 수 있도록 공격 유형들에게서 나타나는 특징들을 추려내고 정량화 시킬 수 있는 식들을 세웠다.

예선에서는 주어진 데이터셋에서 나타나는 임의의 값으로 기준값을 설정해 이상행위를 판단하는 단계지만, 본선에서는 다양한 데이터셋을 기반으로 기계학습시켜 정교한 수치를 설정할 수 있는 학습모델을 만들 계획이다.

#### [붙임] 참고문헌

1. “Excel specifications and limits.” Microsoft. 2020년 10월 22일 접속, https://support.microsoft.com/en-us/office/excel-specifications-and-limits-1672b34d-7043-467e-8e27-269d656771c3#top
2. “HTTP 상태 코드.” MDN web docs. 2020년 10월 22일 접속, https://developer.mozilla.org/ko/docs/Web/HTTP/Status
3. “HTTP 499 Error Code.” BelugaCDN. 2020년 10월 22일 접속, https://www.belugacdn.com/499-error-code/
4. “참조링크(방문경로/레퍼러)의 검출.” LOGGER. 2020년 10월 22일 접속, https://logger.co.kr/manual\_FR/help.tsp?node=524&TSPSESSIONID=453ff1ecaa847822948f7681e96629fb

1. *MARC NAJORK Microsoft Research, 2009* [↑](#footnote-ref-1)