- 정보 보안의 3대 요소 관점에서 바라본 이상 행위 탐지 분석-

소 속	아이리포 교육센터
팀 명	챌린저
팀 원	김영옥, 김동훈, 이정현, 이진재
제출 일자	2020.10.30

목 차

1. 서론

1.1. 분석 배경과 목적

IT 기술과 통신기술의 발전, 스마트폰과 같은 모바일 기기의 증가는 데이터 생산과 자동저장 과정을 더욱 가속하였으며 이러한 빅데이터 시대로의 패러다임 전환은 웹 로그에 대한 새로운 관점을 제시한다.

구글은 웹 로그를 이용한 알고리즘으로 이전에는 생각하지 못한 혁신적인 맞춤형 광고 시스템으로 온라인 광고 시장에 큰 변화를 일으켰다. 이와 같이 K-사이버 시큐리티 챌린지를 참여하는 모든 팀이 웹 로그 이상 행위에 대한 패턴 분석과 다양한 탐지 알고리즘을 제시하며 선의의 경쟁을 통해 온라인 보안 체계를 더욱 혁신적으로 발전시킬 수 있으리라 기대한다.

1.2. 분석 범위와 방법

우리 챌린저 팀은 대회로부터 주어진 약 150만 건의 HTTP 웹 로그 데이터를 이용하여 분석을 진행했다. 2장에는 분석 환경에 대한 설명을, 이어 3장에서는 전반적인 분석 절차와 이상 행위의 기준에 대하여 서술할 것이며 4장에서 주어진 데이터에 대한 구체적인 분석에 대하여 설명할 것이다. 5장에서는 정의된 이상 행위를 탐지하는 과정에 대하여 서술한 후 6장을 통해 이상 행위로 특정된 로그 리스트와 이상 행위로 탐지된 이유에 대해 설명할 것이다. 마지막 7장 향후 계획을 통하여 탐지 알고리즘 자동화, CNN 알고리즘 접목에 대한 설명으로 마무리할 예정이다.

2. 분석환경

2.1. 분석 시스템

2.1.1. 분석에 사용된 시스템 정보

분석에 사용된 시스템 정보는 다음 <표2-1>과 같다.

카테고리	소프트웨어	버전
운영체제	Windows 10	1909
프레임워크	Anaconda 3	4.8.3
언어	Python	3.7.6
분석 툴	Jypyter notebook, Excel	-

<표2-1> 분석 환경

3. 분석절차

3.1. 분석 데이터 이해

이상 행위 패턴 분석에는 실제와 유사한 웹 사이트 환경 구축을 통해 정상·비정상 행위를 발생시킨 HTTP 웹 로그 약 150만건을 수집한 데이터를 사용했다. 해당 분석을 통해 웹 스크래핑, 크리덴셜 스터핑 등 악의적인 이상 행위를 식별하고 이상 행위 데이터의 패턴을 탐지할 수 있는 알고리즘을 만드는 것을 목적으로 진행했다. 본격적인 분석에 들어가기 앞서 HTTP 웹 로그를 이해하고 각 컬럼들이 의미하는 정보를 파악했다. HTTP 웹 로그 데이터의 내부 형태는 <표4-1>과 같다. 기술 통계 및 탐색적 데이터 분석을 통해 각각의 로그 데이터의 특징을 파악했다. 또한, Status 데이터에서 인코딩 에러로 보이는 이상 데이터가 발견되어 해당 데이터를 제거한 후 분석에 들어갔다.

3.2. 이상 행위 분석

정보 보안이란 정보 및 정보 시스템을 허가되지 않은 접근, 사용, 공개, 손상, 변경, 파괴 등으로부터 보호함으로써 기밀성, 무결성, 가용성을 제공하는 것을 뜻한다. 탐색적 데이터 분석 결과와 <표3-1>의 내용을 바탕으로 가설을 설정하여 이를 검정하는 방식으로 정보 보안의 3 요소인 기밀성, 무결성, 가용성을 침해하는 데이터의 패턴을 찾았다.

요소	정의	가설
기밀성	허가 받은 사용자만이 정보에 접근할 수 있다.	- 관리자 권한 없이 관리자 페이지와 같은 특정 권한이 필요한 사이트에 접근한 경우 이상 행위이다. - 로그인을 성공하지 못하고 여러 번 시도하는 경우 이상 행위이다.
무결성	적절한 권한을 가진 사용자만이 인가된 방법으로만 정보를 변경할 수 있다.	특정 사이트에서 비정상적으로 높은 바이트를 요청한 사용자가 있을 경우 이상 행위이다.
가용성	정보에 대한 접근과 사용이 적시에 확실하게 보장되어야 한다.	Host 차원에서 무차별로 접속 시도가 있을 경우 이상 행위이다.

<표3-1> 정보 보안의 3대 요소

4. 데이터 이해

4.1. HTTP 웹 로그 이해

○ 총 데이터 건: 1,550,250 행

○ 시간별 분석을 편리하게 하기 위해 'Timestamp' 컬럼의 타입을 <표4-1>의 내용과 같이 변경함

○ <그림4-1>의 이상 데이터를 제거한 후 null 데이터가 34,663개에서 5,396으로 줄어듦

컬럼	설명	데이터 수	타입	비고
Timestamp	시간	1,550,250	object	datetime64 로 변경
Method	클라이언트가 웹서버에게 사용자 요청의 목적/종류를 알리는 수단	1,550,230	object	GET/POST
Protocol	접속 프로토콜	1,549,819	object	
Status	HTTP 상태 코드	1,549,881	object	
Referer	요청 헤더, 현재 요청된 페이지 이전 주소를 포함	1,549,396	object	
Path	웹 서버 상의 리소스 경로	1,550,230	object	
Host	요청한 Host 의 IP	1,533,090	object	
UA	현재 사용자가 어떤 클라이언트로 접속하였는지	1,549,178	object	
Payload	HTTP 요청을 보낼 때, 포함되는 데이터	1,549,081	object	
Bytes	요청과 응답 메시지의 본문 크기를 바이트 단위로 표시. 메시지 크기에 따라 자동으로 만들어짐.	1,536,682	float64	

<표4-1> HTTP 웹 로그 데이터 컬럼 설명

4.2. HTTP 상태 코드 시간별 추이

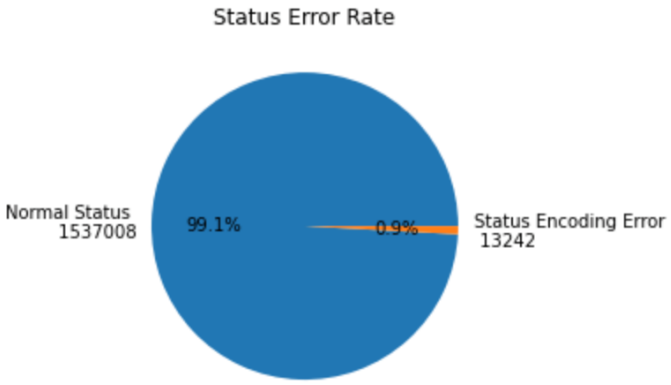
4.2.1. 이상 데이터 발견

○ Status에서 인코딩 에러로 보이는 데이터를 발견함

○ <그림4-1>에 나타난 비율을 통해 이상 데이터를 제거 후 분석을 진행하기로 판단함

Status	개수
stderr:	12,120
with	349
be	31
while	12
...	...
₩xEC₩x88₩x98	3

<표4-2> Encoding Error Data



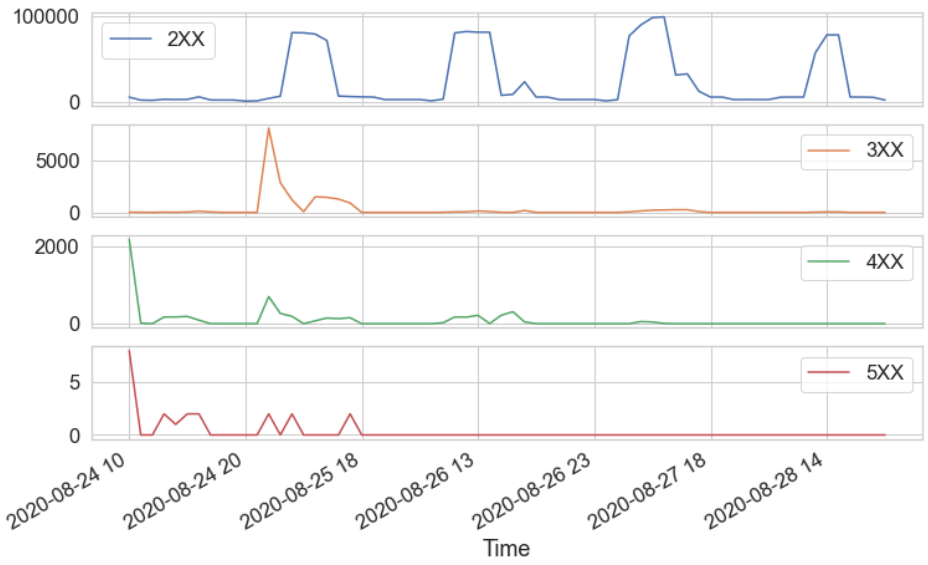
<그림4-1> 정상 데이터와 Encoding Error 데이터 비율

4.2.2. HTTP 상태코드 이상 패턴 발견

○ 2xx번대(성공) 상태 코드는 8월 25일 12시부터 15시까지, 8월 26일 11시부터 14시까지, 26일 17시, 27일 11시부터 17시까지, 28일 13시부터 15시까지 급격하게 증가했다 감소하는 패턴이 발생함

○ 3xx번대(리다이렉션 완료) 상태 코드는 8월 25일 10시~12시, 14시~17시에 대량으로 발생하고 2xx번대 상태 코드가 급격하게 증가했다 감소하는 패턴이 보인 시간대에 이외의 시간대보다 많이 발생하는 패턴을 보임

○ 4xx번대(요청 오류) 상태 코드는 8월 24일 10시에 전체 4xx 상태 코드 중 37.9%를 차지하였으며 24일 10시~16시, 25일 10시~17시, 26일 10시~17시, 27일 12시~14시에 증가했다 감소하는 패턴이 발생함



<그림4-2> HTTP 상태 코드 시간별 추이

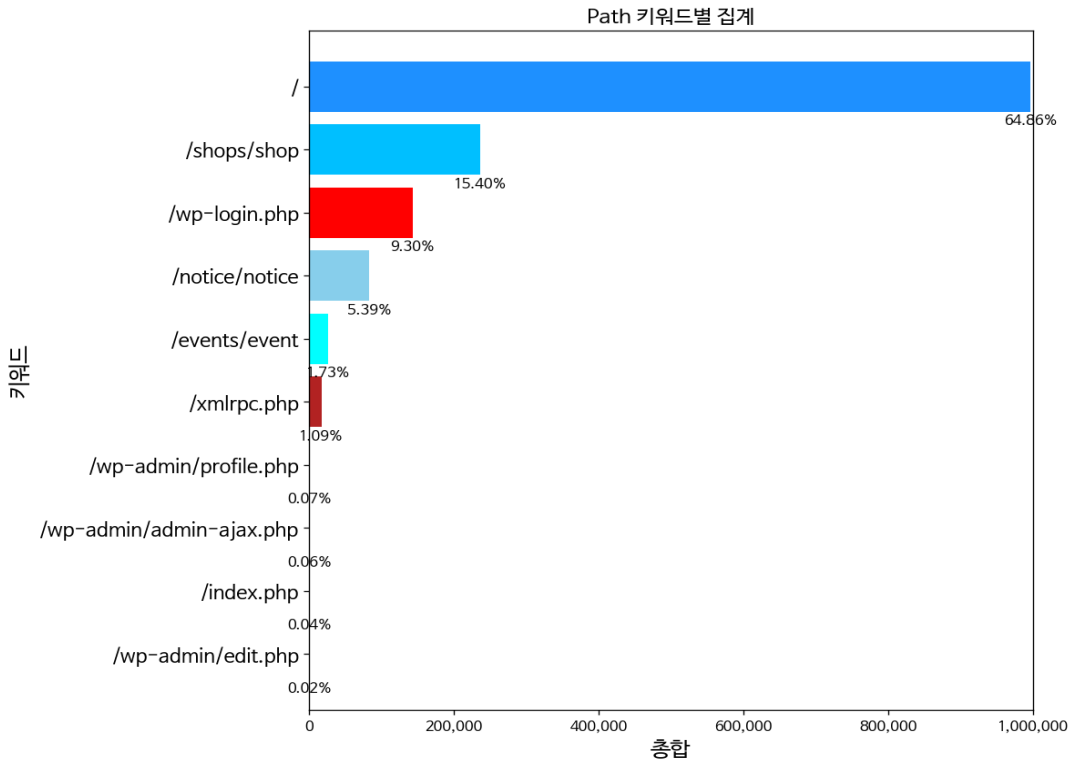
4.3. Path 키워드별 집계

○ Path에서 query 부분을 제외한 앞 부분 Path에 대한 집계

○ /?p=, /?cat=, /?paged= 와 같은 query를 가진 '/' 가 64.86%로 가장 많았으며 그 다음으로 /shops/shop, /wp-login.php 순으로 많음

○ 특이한 점으로 /wp-login.php 즉, WordPress 관리자 로그인 페이지에 9.30%이나 접속을 시도한 것을 발견함

○ WordPress의 취약점으로 잘 알려진 /xmlrpc.php에 1.09%의 수치로 접속을 시도한 것을 파악함



<그림4-3> Path 키워드별 집계

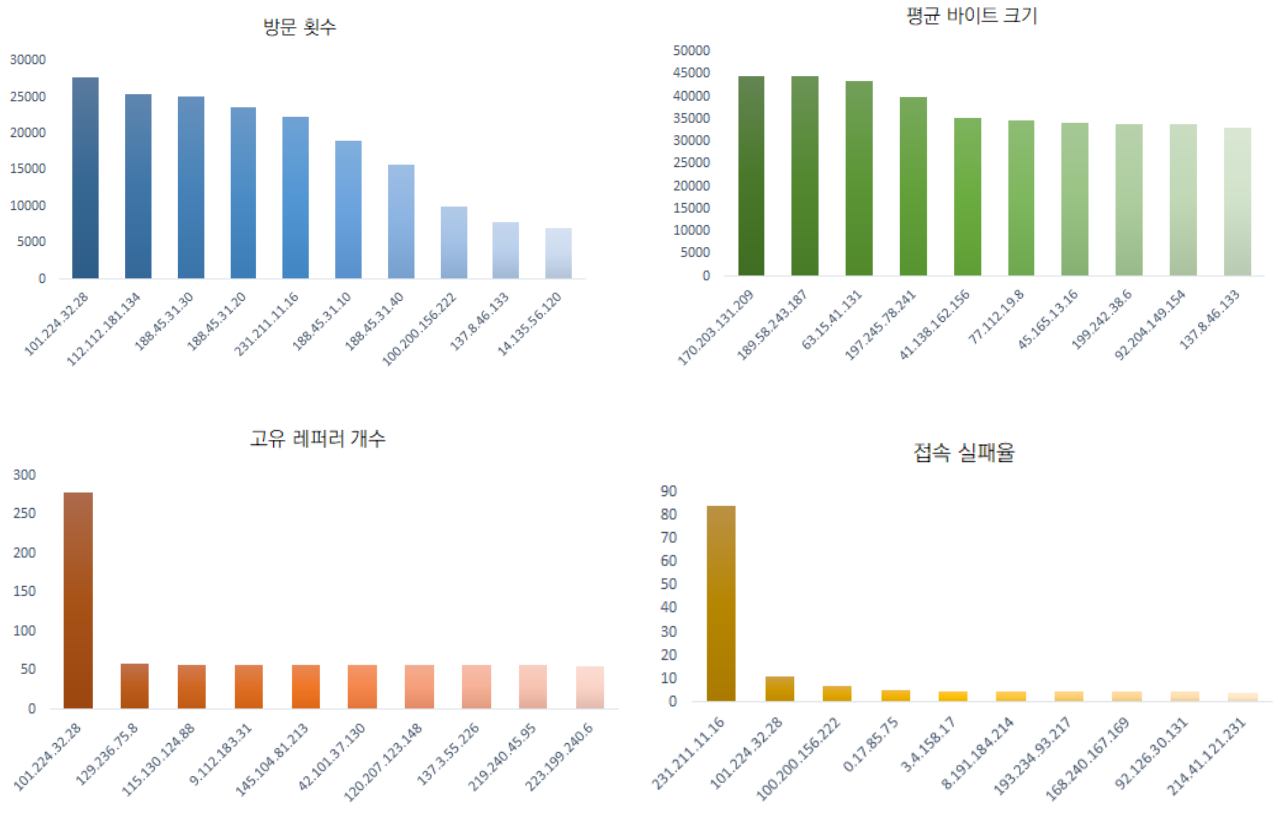
4.4. Host별 집계

○ <표4-2>는 Host별로 방문한 Referer의 고유 개수, 사이트 방문 횟수, 평균 바이트, 접속 성공 여부에 대한 정보를 나타낸다.

○ 각각의 Host의 방문 횟수, 평균 바이트 크기, 고유 Referer 수, 접속 실패율을 내림 정렬하여 <그림4-7>과 같이 각 Host 별로 어떤 특징을 보이는지 파악했다.

Host	고유 Referer 수	접속 횟수	평균 바이트	성공	리다이렉션 오류	요청 오류	서버 오류
0.113.88.147	17	437	22,713.4	437	0	0	0
0.116.162.78	6	9	14,324	9	0	0	0
0.128.52.6	7	14	17,991.79	14	0	0	0
0.145.130.53	18	41	11,297.15	41	0	0	0
…	…	…	…	…	…	…	…
99.98.31.109	18	124	20,798.81	124	0	0	0

<표 4-3> Host 별 집계표



<그림 4-4> 상위 10 개 Host 추출

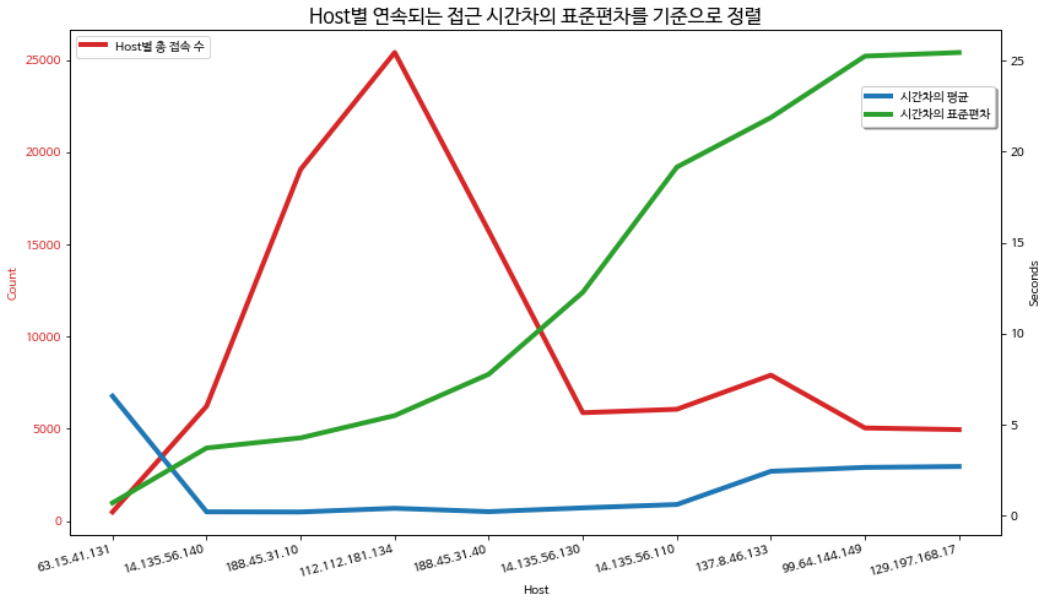
4.5. Host별 연속된 로그의 시간차 분석

4.5.1. 표준편차 기준으로 분석

○ Host별로 연속되는 로그의 시간차의 평균과 표준편차를 계산하여 시각화 하였다.

○ 연속되는 접속 시간차의 표준편차가 가장 작은 Host '63.15.41.131' 약 6.57초의 평균으로 표준편차가 0.7초로 접속하였던 것으로 분석되었다.

○ 표준편차로 정렬한 기준으로 2번째부터 6번째까지 Host '14.135.56.140', '188.45.31.10', '112.112.181.134', '188.45.31.40', '14.135.56.130', '14.135.56.110'는 접속 평균 시간차가 비정상적인 연속 접속이라고 할 수 있는 1초 이내인 것을 확인하였다.



<그림4-5> Host별 연속 접근 시간차의 표준편차 기준 정렬

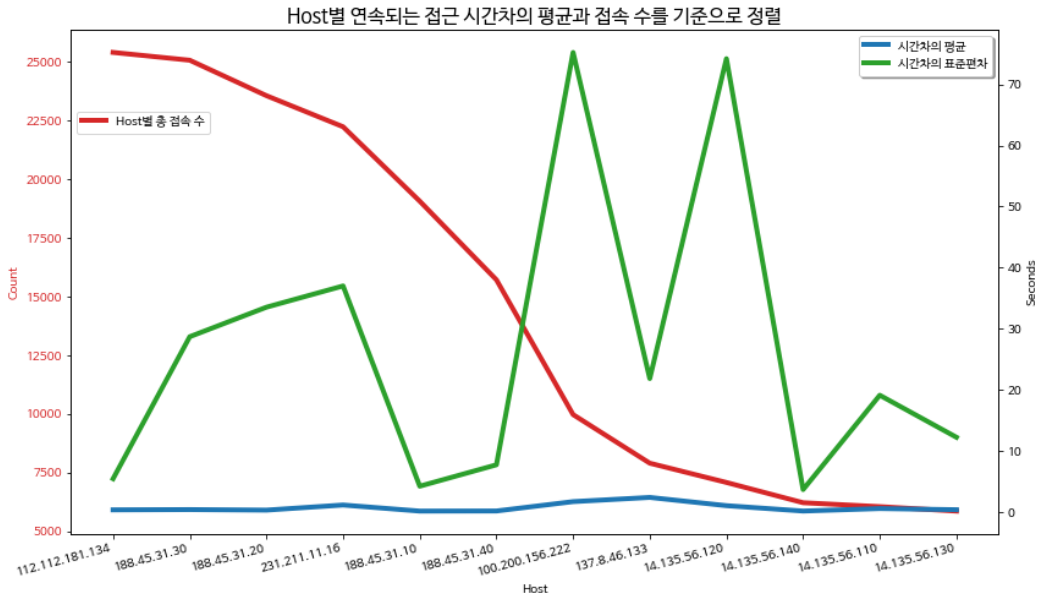
4.5.2. 평균과 접속 수를 기준으로 분석

○ Host별로 연속되는 로그의 시간차의 평균과 표준편차를 계산하고 평균시간이 짧은 순으로 정렬 후에 100개의 Host에서 다시 접속수가 많은 순서대로 정렬했다.

○ Credential stuffing[4] 의심 Host '188.45.31.10/20/30/40', '14.135.56.110/120/130/140', '100.200.156.222', '112.112.181.134'는 평균 1초이내의 비정상적인 접근 주기를 가지며 Path '/wp-login.php'에 접속하는 것으로 분석되었다.

○ Host '231.211.11.16'는 평균 1.19초의 접근 주기를 가지며 웹 서버의 디렉토리를 탐색하는 "Forced Browsing Attack"[5] 또는 "Directory Guessing Brute Force"[참조문헌추가]라고 불리는 Brute Force를 사용하여 제한된 URL에 접속하려는 공격을 하는 것으로 분석되었다.

○ Host '137.8.46.133'는 총 7911번 접속하면서 평균 2.44초대의 접속 주기를 가지며 표준 편차가 21.87초였다. '/?p=1620'과 같은 게시글이라 예상되는 Path에 접속하면서 대부분의 Path를 요청하기 전의 URL인 Referer가 '-'인 것을 보아 사람이 직접 접속하여 보는 것이 아닌 봇을 통한 '웹 스크래핑' 행위라고 예상하였다.



<그림4-6> Host별 연속 접근 시간차의 평균 및 접속 수 기준 정렬

5. 이상 행위 분석

5.1. 기밀성 침해 공격

5.1.1. 정의

정보보안의 3요소 중 하나인 기밀성은 정보를 오직 인가(authorization)된 사람들에게만 공개하는 것을 의미한다. 로그인 권한 없이 관리자 페이지와 같은 특정 권한이 필요한 사이트에 접근한 경우, 로그인을 성공하지 못하고 여러 번 시도하는 경우를 기밀성에 기초하여 가설을 설정하였다.

5.1.2. 분석

관리자 권한이 필요한 사이트의 경우 '/wp-admin/', '/wp-includes/', '/wp-content/'와 같은 WordPress 와 XpressEngine 웹 프레임워크의 디렉터리 구조[7],[8]를 찾아 키워드를 나열한 후 해당 키워드에 접근한 Host 만 추출했다. 웹 플랫폼 WordPress 에서 권한을 얻을 수 있게 로그인하는 페이지인 '/wp-

login.php'에 접근하여 권한을 정상적으로 얻었는지 분석하고 각 Host 들에 대해 어떤 패턴을 가지고 있는지 분석했다.

웹 프레임워크 디렉터리 키워드에 접속한 13 개의 Host 들에 대해서 분석을 하였다.

○ Host '101.224.32.28'

Referer 에서 'http://wp.hotspot.kr'을 통하여 접속하는 것으로 보아 WordPress 프레임워크로 접근하는 것을 알 수 있었다. 24 일부터 27 일까지 꾸준히 접속했다. WordPress 의 권한을 획득할 수 있는 'wp-login.php'에 146 회 접속하였고 Status '302'로 로그인 성공을 한 횟수는 4 번이었다. 로그인 아이디는 'admin', 'alpha', 'fnhba'을 사용하였다. 로그인 권한을 획득했다고 하기에는 로그인 이전에 'Directory Guessing Brute Force'와 같은 행동을 보여 비정상 적으로 권한을 얻었다고 간주했다. 로그인 이후에 WordPress 의 취약점 중 하나인 'xmlrpc.php'에 짧은 시간에 대량으로 접근을 시도하였으며 비정상적으로 높은 Byte 를 가진 로그가 있는 것으로 보아 시스템 내부의 데이터를 변조했다고 분석했다.

○ Host '231.221.11.16'

8 월 25 일 10 시 19 분부터 8 월 25 일 17 시 42 분까지 22238 번의 접속을 시도했다. 그 중 'wp-login.php'에 접속한 횟수는 6 회였으며 Payload 와 302 Status 를 확인했을 때 로그인을 성공한 흔적은 존재하지 않았다, 관리자 권한이 필요한 사이트에 8971 번 접속하였으며 그 중 'admin' 키워드가 들어간 로그의 수는 5183 회였다. 위 Host 의 행동은 권한없이 접근이 가능한 페이지나 디렉토리를 찾는 'Directory Guessing Brute Force' 공격 Host 라고 분석했다.

○ Host '137.8.46.133'

8 월 25 일 10 시 59 분부터 8 월 25 일 16 시 21 분까지 7911 번의 접속을 시도했다. 그 중 'wp-login.php'에 접속한 횟수는 2 회였으며 Payload 와 302 Status 를 확인했을 때 로그인 성공을 하지 못했다. 접속한 Path 를 분석한 결과 '/content/'에 접속을 시도하였으며 주로 사이트의 상세 페이지 같은 곳을 약 1 초의 간격으로 접속하는 것을 확인했다. 관리자 권한이 필요한 사이트에 접속한 것으로는 볼 수 없으며 '웹 스크래핑' 행위를 하고 있는 Host 라고 분석했다.

○ Host '100.200.156.222'

8 월 26 일 10 시 45 분부터 8 월 26 일 15 시 36 분까지 9980 번의 접속을 시도했다. 그중 'wp-login.php'에 접속한 횟수는 9969 회였으며 Status 가 302 로 로그인 성공을 했다고 가정한 횟수는 111 번이다. 총 접속 횟수가 9980 회인데 그중 로그인 시도가 9969 회(99%)이며 Payload 에 담고 있는 로그인 정보가 계속 새로운 아이디와 패스워드로 변경되며 접속 로그의 연속되는 시간차가 1 초에도 여러 번 발생하는 것을 통해 이 Host 가 로그인에 성공하여 권한을 얻었다고 하더라도 '크리덴셜 스터핑' 공격을 하고 있다고 볼 수 있다.

○ Host '112.112.181.134'

8 월 26 일 14 시 57 분부터 8 월 26 일 17 시 48 분까지 25404 번 접속을 시도했다. 그중 'wp-login.php'에 접속한 횟수는 25213 회였으며 Status 가 302 로 로그인 성공을 했다고 가정한 횟수는 194 번이다. 관리자 권한을 얻었지만 정상적이라고 할 수 없는 이유는 Host '100.200.156.222'와 유사하게 총 접속횟수 중에서 관리자 로그인 페이지에 접속한 비율이 99%였으며 Payload 의 아이디, 패스워드의 정보가 새롭게 바뀌며 연속 시간차가 비정상적으로 빠르게 발생하는 것을 통해 이 Host 도 '크리덴셜 스터핑' 공격을 시도하고 있다고 볼 수 있다.

○ Host '188.45.31.XX'

Host 의 뒤의 두자리만 다른 188.45.31.XX 로 총 8 월 27 일 12 시 38 분부터 같은 날 17 시 9 분까지 '188.45.31.10', '188.45.31.20', '188.45.31.30', '188.45.31.40' Host 가 위의 '크리덴셜 스터핑 '과 같은 패턴으로 공격을 시도했다.

○ Host '14.135.56.1XX'

Host 의 뒤의 두자리만 다른 14.135.56.1XX 로 총 8 월 27 일 14 시 57 분부터 같은 날 17 시 3 분까지 '14.135.56.110', '14.135.56.120', '14.135.56.130', '14.135.56.140' Host 가 위의 '크리덴셜 스터핑과 같은 패턴으로 공격을 시도했다.

Host	총 접속 횟수	wp-login.php 접근횟수	총 접속 횟수대비 wp-login.php.접근 횟수비율(%)	로그인 성공 횟수	분류
101.224.32.28	27757	146	0.52	4	디렉토리 추측 브루탈 포스, xmlrpc.php 취약점 공격
231.211.11.16	22238	6	0.02	0	디렉토리 추측 브루탈 포스
137.8.46.133	7911	2	0.02	0	웹 스크래핑
100.200.156.222	9980	9969	99.89	111	크리덴셜 스터핑
112.112.181.134	25404	25213	99.25	194	크리덴셜 스터핑

188.45.31.10	19068	18814	98.67	254	크리덴셜 스터핑
188.45.31.20	23562	23397	99.3	165	크리덴셜 스터핑
188.45.31.30	25071	24854	99.13	217	크리덴셜 스터핑
188.45.31.40	15720	15570	99.05	150	크리덴셜 스터핑
14.135.56.110	6060	5989	98.83	71	크리덴셜 스터핑
14.135.56.120	7090	7033	99.2	57	크리덴셜 스터핑
14.135.56.130	5871	5800	98.79	71	크리덴셜 스터핑
14.135.56.140	6227	6193	99.45	34	크리덴셜 스터핑

<표 5-1> 웹 프레임워크 디렉터리 키워드에 접속한 Host

5.2. 무결성 침해 공격

5.2.1. 정의

데이터 무결성은 데이터의 일관성, 정확성, 신뢰성을 보장하는 것을 의미한다. 무결성 손실은 권한이 없는 특정 사용자에 의해 데이터가 수정 또는 파괴되었거나 시스템 구성이 변경되어 사용자가 허위 정보를 겪을 수 있다는 것을 의미한다. 따라서 데이터는 전송하는 과정에서 변경되어서는 안 되며 권한이 없는 사람이 악의적으로 접근하여 데이터를 변경할 수 없도록 조치를 취해야 한다.

5.2.2. 분석

<표3-1>에 따라 '특정 사이트에서 비정상적으로 높은 바이트를 요청한 사용자가 있으면 이상 행위일 것이다'는 가설을 설정하여 분석했다. 이는 일반적으로 사용자가 해당 사이트에서 요청하는 데이터의 정상 범주를 벗어난 경우 기존의 데이터에 악의적으로 접근하여 데이터를 변조했을 가능성이 높다는 가정하에 이루어졌다.

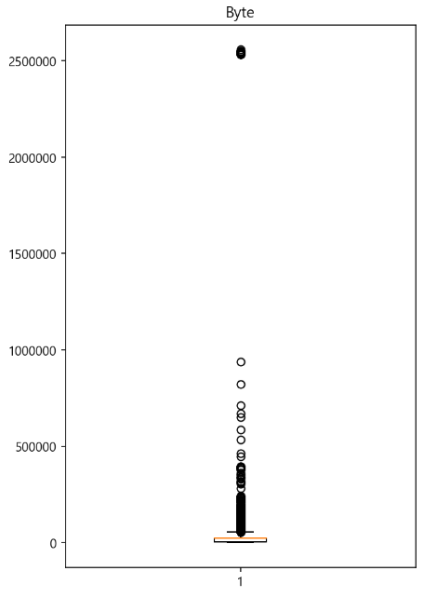
전체 데이터에서 평균값은 18,952.46인 반면, 사분위수 범위에서 크게 벗어난 이상치가 존재하는 것을 발견했다. 단순히 평균값 이상의 큰 값들을 추출하여 분석하기보다는 시간별 바이트 변동 량을 비교하여

이상치 패턴을 파악하는 것이 중요하다고 판단했다.

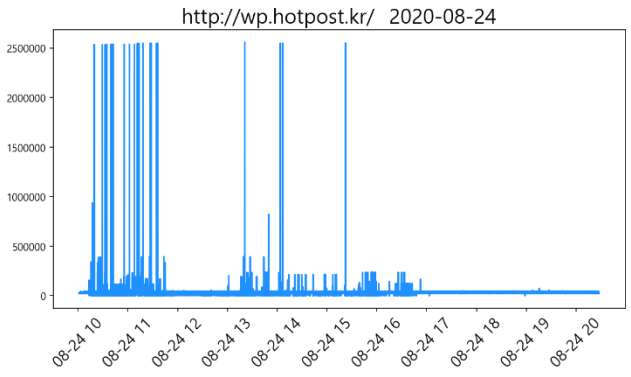
또한 날짜별로 전체 데이터의 Byte 변동 추이를 살펴보았을 때 8월 24일에만 유난히 많은 양의 데이터가 요청되었음을 발견했다. 해당 패턴에 유의하여 가장 많이 사용한 Referer를 중심으로 사용자들이 어떤 사이트를 통해 접속한 경우 이러한 이상치가 발견되는지를 확인해보았다.

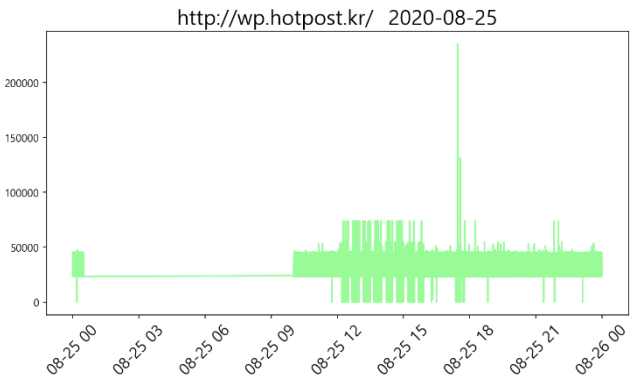
count	1,536,682
mean	18,952.46
std	16,497.07
min	0
25%	3,845
50%	24,066
75%	24,360
max	2,558,453

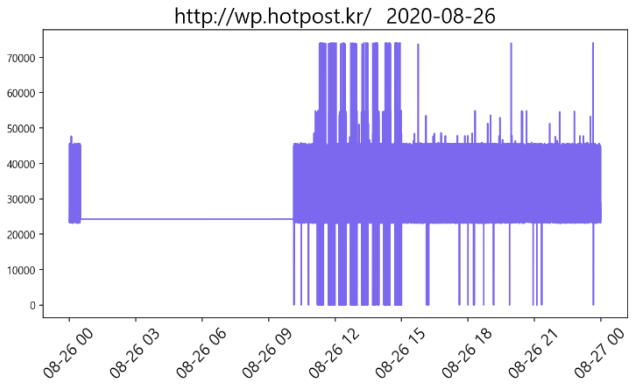
<표 5-2> Byte 통계 정보

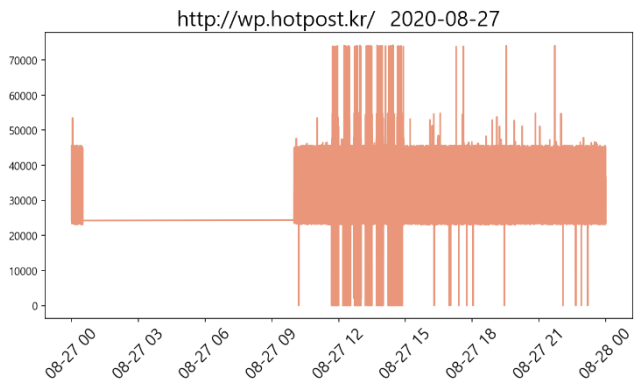


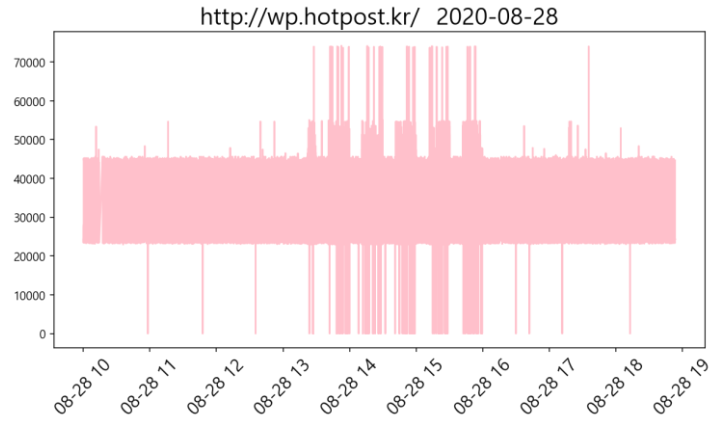
<그림 5-1> Byte Box Plot











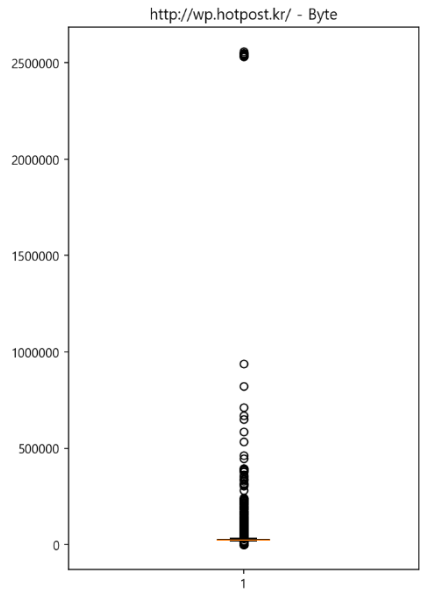
<그림 5-2> 날짜 별 Byte 크기

○ http://wp.hotpost.kr을 통해 접속한 경우

대체적으로 24,000~26,000 byte가 요청된 반면, 8월 24일에만 지나치게 많은 크기의 데이터를 요청한 이상치를 발견했다. <그림5->에 따라 500,000 byte 초과 요청한 경우와 50,000 byte 초과 500,000 byte 이하 요청한 경우로 분류하여 더 자세하게 데이터 패턴을 분석했다.

count	955,648
mean	26,700.73
std	14,786.76
min	0
25%	24,067
50%	24,253
75%	27,258
max	2,558,453

<표5-3> Byte 통계 정보

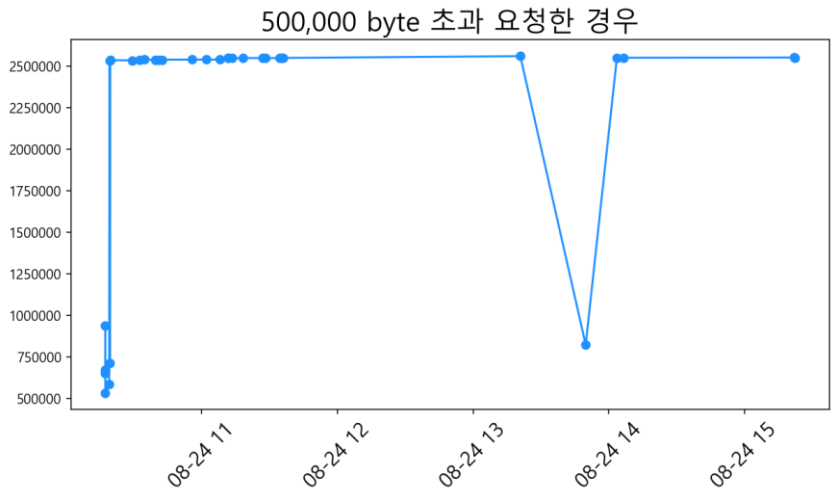


<그림5-3> Byte Box Plot

- 500,000 byte 초과 요청

http://wp.hotpost.kr 을 통해 접속한 사용자 중에서 500,000 byte를 초과한 경우는 총 32회다. 이에 해당하는 Host는 '101.224.32.28' 뿐이며 사용한 UA는 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)

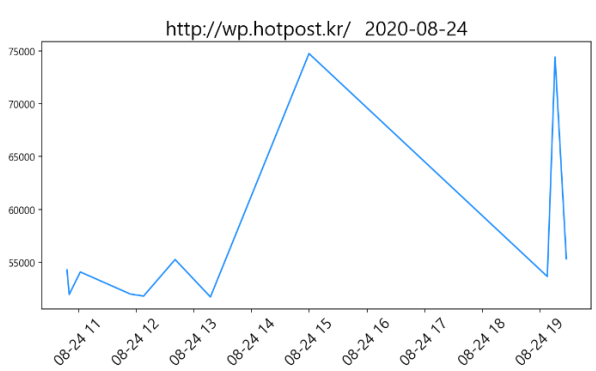
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36'로 하나의 UA만을 사용했다는 점을 발견했다. 해당 Host에서 사용한 경로는 /wp-admin으로 WordPress 관리자 페이지에만 집중적으로 접속했다.

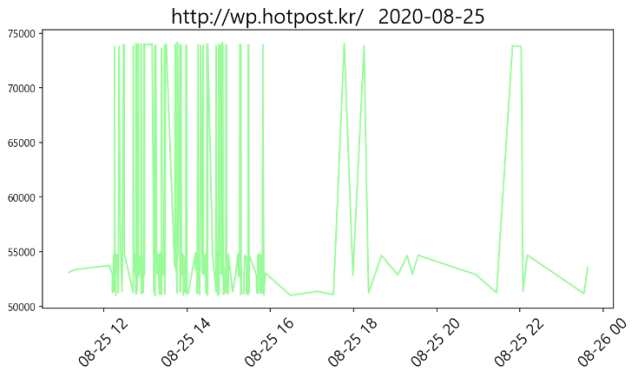


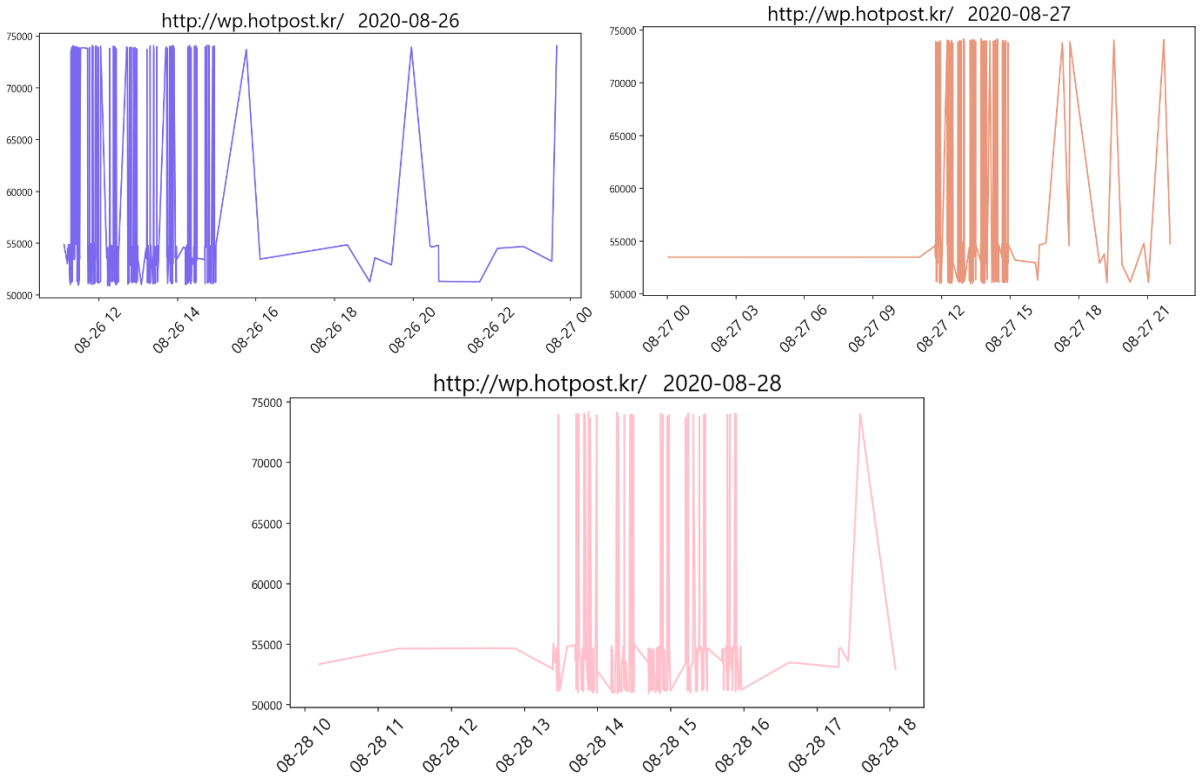
<그림5-4> 500,000 byte 초과 요청

- 50,000 ~ 500,000 byte 요청

http://wp.hotpost.kr/ 을 통해 접속한 사용자 중에서 50,000 ~ 500,000 byte를 요청한 경우는 총 2,065회다. Host는 총 504개, UA는 총 187개에서 해당 사이트를 접속했다. 가장 많이 접속한 Host는 앞서 발견한 '101.224.32.28'이다. 이를 제외한 나머지 503개의 Host는 대략 50,000 ~ 74,750 크기의 데이터를 사용했다는 점을 <그림5-5>를 통해 알 수 있다.







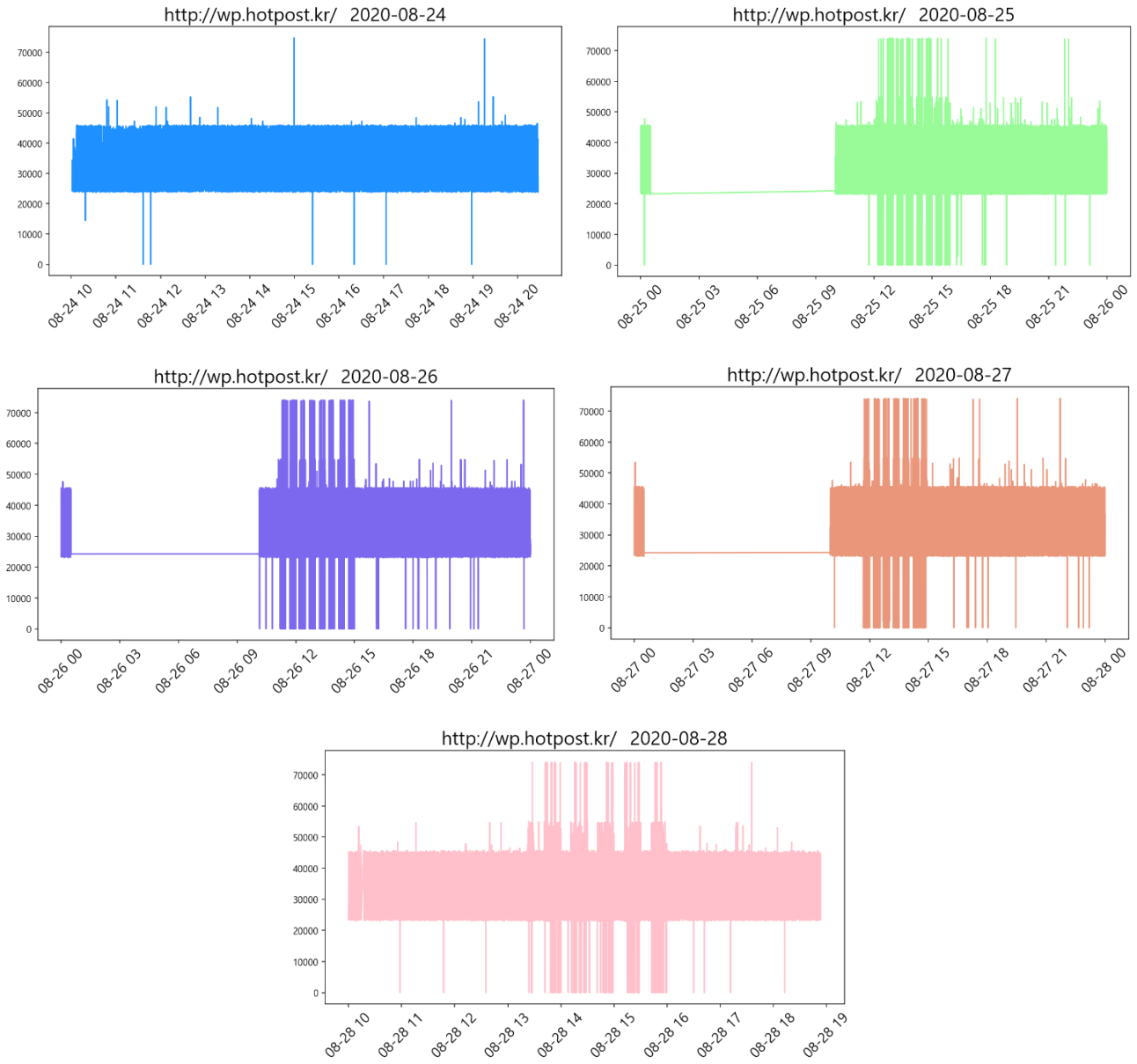
<그림5-5> 50,000 ~ 500,000 byte 요청

- Host가 '101.224.32.28'인 경우

해당 Host는 <표5-1>의 내용과 같이 주로 WordPress 관리자 페이지에 접근을 시도한 Host다. 접근 경로는 WordPress의 취약점 중 하나인 'xmlrpc.php'이며 사용한 UA는 Python-xmlrpc/3.6으로 각각 324개로 가장 많은 수치를 보였다. 비교적 짧은 시간에 다량 접속을 시도했으며 전체 바이트의 정상 범주를 넘어선 높은 바이트를 가진 로그가 있는 것으로 보아 해당 페이지의 내부의 데이터를 변조한 것으로 판단했다.

- Host가 '101.224.32.28'가 아닌 경우

Host를 기준으로 바이트 요청 량을 비교해 보았을 때 <그림5-6>과 같이 비교적 일정한 양의 데이터가 요청되었다는 사실을 발견했다. 일정 기간동안 동일한 크기의 데이터를 사용했다는 점에서 '웹 스크래핑'이 의심된다.



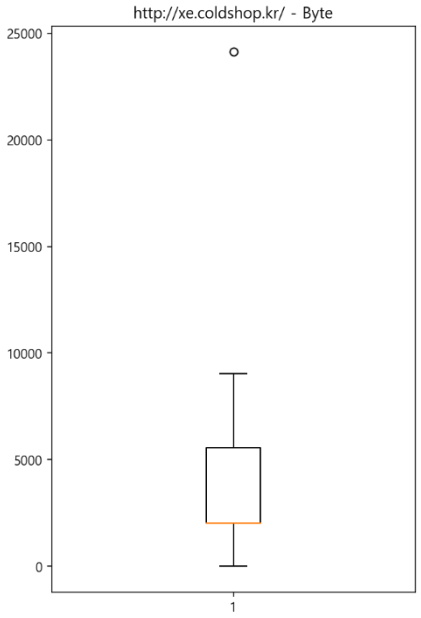
<그림5-5> Host '101.224.32.28'을 제외한 Byte 사용량

○ http://xe.coldshop.kr을 통해 접속한 경우

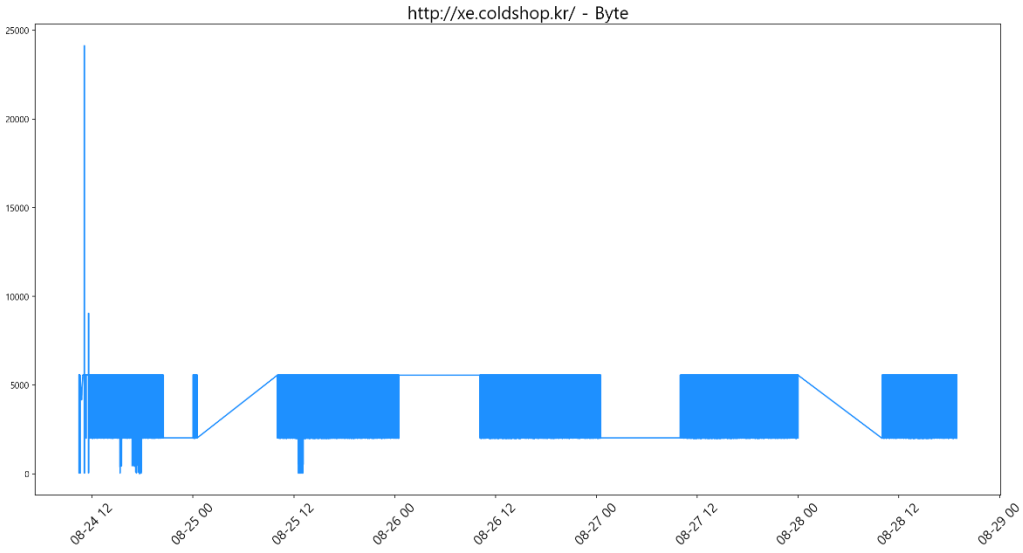
평균이 3,101 byte고 중앙값이 2,024 byte로 http://wp.hotpost.kr를 통해 접속한 경우보다 비교적 요청한데이터의 크기가 작다. 대체적으로 2,000 ~ 5,000 바이트가 해당 사이트에 접속할 때 요청되었다. 해당 사이트 역시 다른 날에는 데이터 크기가 고르게 분포되어 있는 반면, 8월 24일에 정상 범주를 벗어난 Byte 크기가 존재한다. <그림5-6>과 <그림5-7>을 통해 정상 범주를 벗어난 이상치가 있다는 것을 확인했고 이를 토대로 앞서 했던 방법과 동일하게 바이트 크기별로 분류하여 분석했다.

count	323,631
mean	3,101.10
std	1,629.63
min	0
25%	2,022
50%	2,024
75%	5,556
max	24,128

<표5-4> Byte 통계 정보



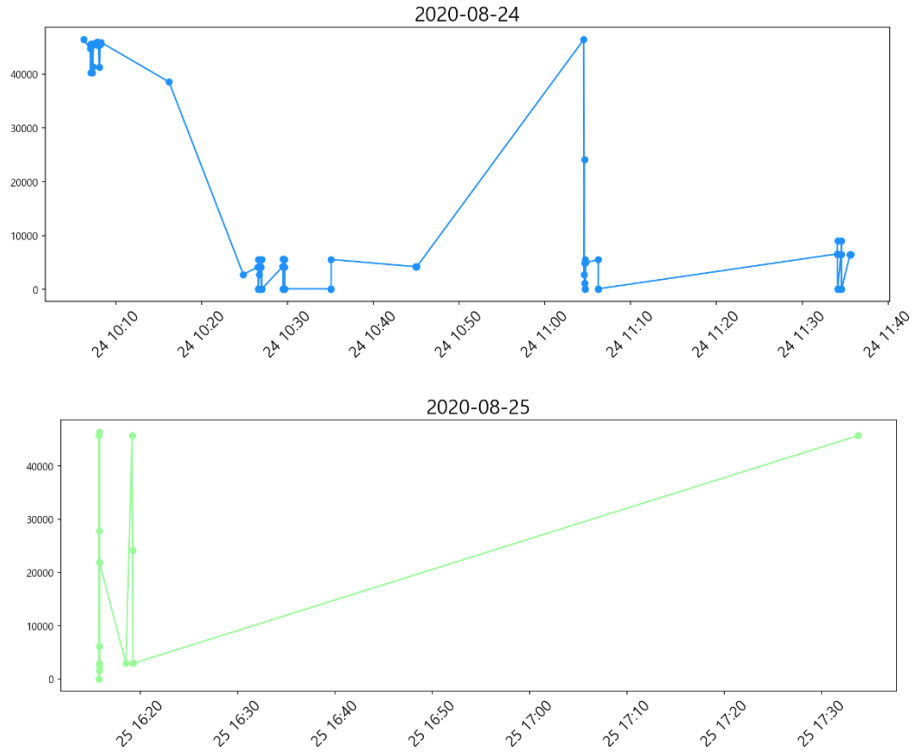
<그림5-6> Byte Box Plot



<그림 5-7> 날짜 별 Byte 크기

- 6,000 byte 초과 요청

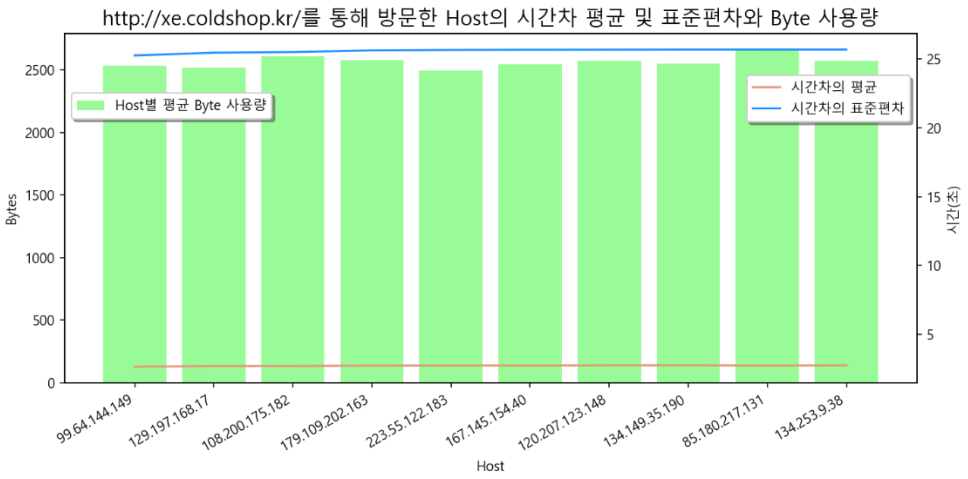
http://xe.coldshop.kr을 통해 접속한 사용자 중에서 6,000 byte를 초과한 경우는 총 10회다. 이에 해당하는 Host는 NaN 값이며 사용한 UA는 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.83 Safari/537.36 Edg/85.0.564.41 '로 하나의 UA만을 사용했다는 점을 발견했다. Host가 NaN이므로 Host별 데이터 특징을 파악하기 어려워 해당 UA를 기준으로 데이터를 추출했다.



<그림5-8> http://xe.coldshop.kr에서 6,000 Byte 초과 요청한 사용자의 전체 Byte 사용량

- 6,000 byte 이하 요청

http://xe.coldshop.kr을 통해 접속한 사용자 중에서 6,000 Byte 이하를 요청한 경우는 총 323,621회다. Host는 총 3,692개, UA는 총 253개에서 해당 사이트를 접속했다. 각각의 Host 별로 약 900회씩 사이트에 접속했다. 또한 Host별 접속 시간차를 구했을 때 상당 수의 Host의 시간차 평균이 2.8초, 시간차 표준편차가 25~27초 사이를 기록했다. <그림5-9>와 같이 일정한 시간 동안 비슷한 크기의 데이터에 접근한 것으로 보아 '웹 스크래핑'이 의심된다.



<그림5-9> 최단 시간 접속 상위 10개 Host의 시간차 평균 및 표준편차와 평균 Byte 사용량

5.3. 가용성 침해 공격

5.3.1. 정의

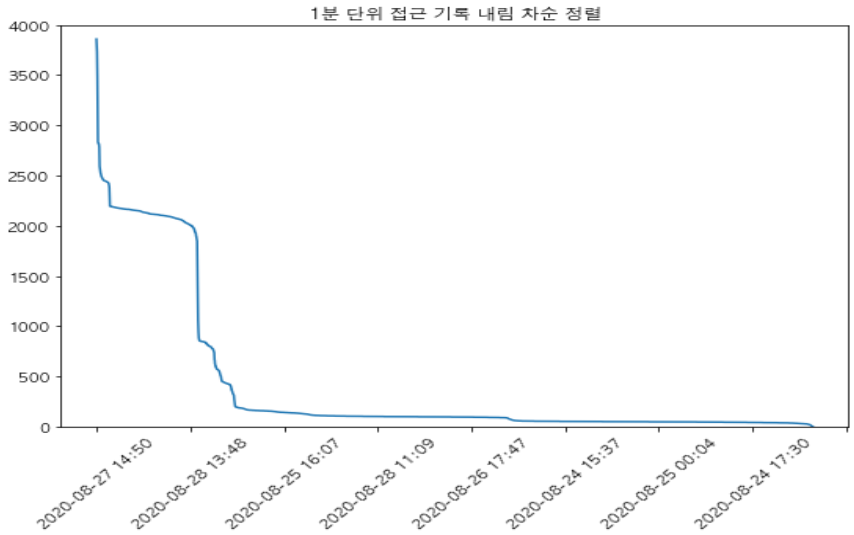
가용성 침해 공격에 대표적인 예로는 DDos 공격을 들 수 있다. 특정 정보를 갈취 또는 변조하여 사용자에게 피해를 주는 방식이 아닌 서버에서 허용할 수 있는 트래픽 양을 초과하는 요청을 보내 사용자가 해당 서버를 이용할 수 없도록 만드는 공격 방식이다. 웹 스크래핑 및 크롤러 또한 유형에 따라 악의적이지 않더라도 해당 서버에 가용성을 침해할 수 있다. 짧은 시간내에 특정 Host가 일반 사용자의 기본 통념을 벗어나는 수치의 요청을 보낼 시 이를 웹 스크래핑으로 판단하고 해당 Host를 차단하는 조치를 취하는 것이 바람직하다.

5.3.2. 분석

우선 '짧은 시간'이라는 추상적 개념을 1분으로 정의하였다. 1분동안 그릇에 쌓이는 모래의 양을 측정하듯 모래의 총 질량은 어떻게 되는지 어떠한 비율로 성분이 구성되어 있는지 판단하는 방식으로 접근했다.

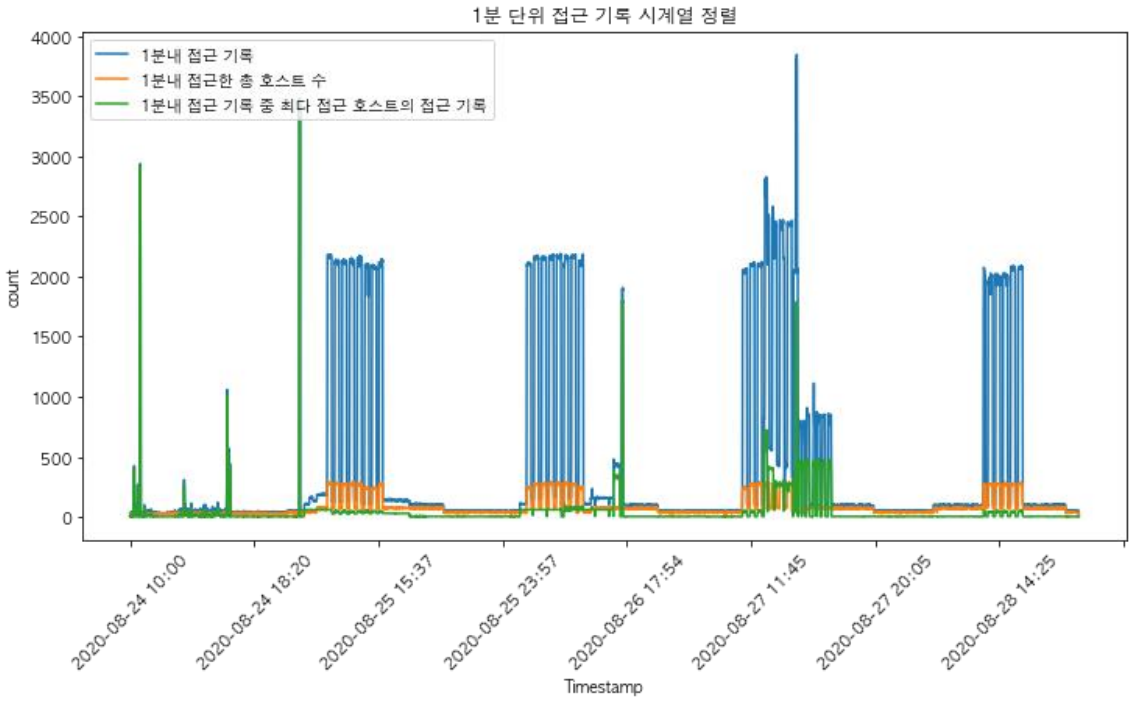
데이터 프레임을 '1분'단위로 그룹화하여 총 3882개의 '1분 '을 도출하였고 통계 분석해보았다.

count	3822
mean	402.147567
std	736.033061
min	1
25%	48
50%	95
75%	148
max	3859



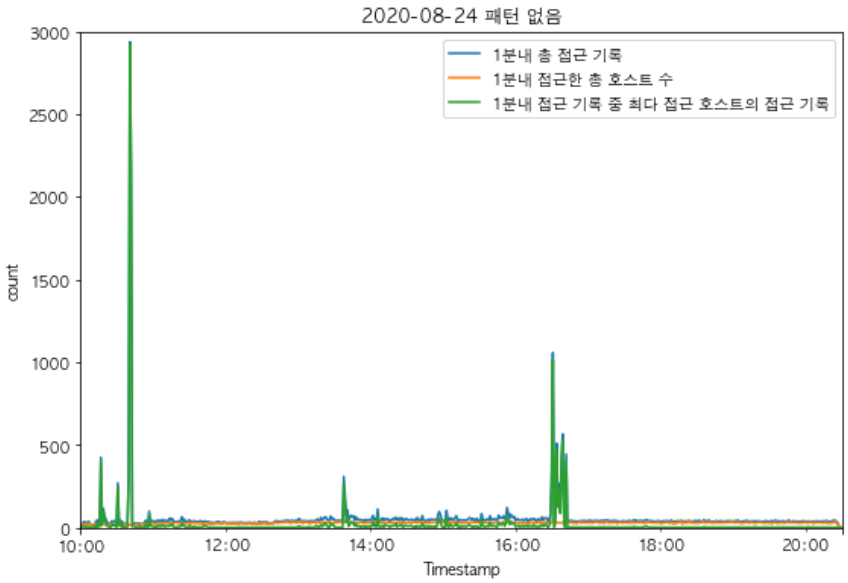
<표 5-5> 1 분 단위 그룹 통계 정보 **<그림 5-10> 같은 1 분으로 묶여 있는 로그의 양으로 정렬**

<표5-5>와 <그림5-10>를 통해 알 수 있듯이 75% 이상의 데이터가 분당 148회를 넘지 않는다. 보다 정확한 분석을 위해 분당 개별 Host와 분당 가장 많은 접근 기록을 남긴 Host의 접근 수를 계산해 보았다.



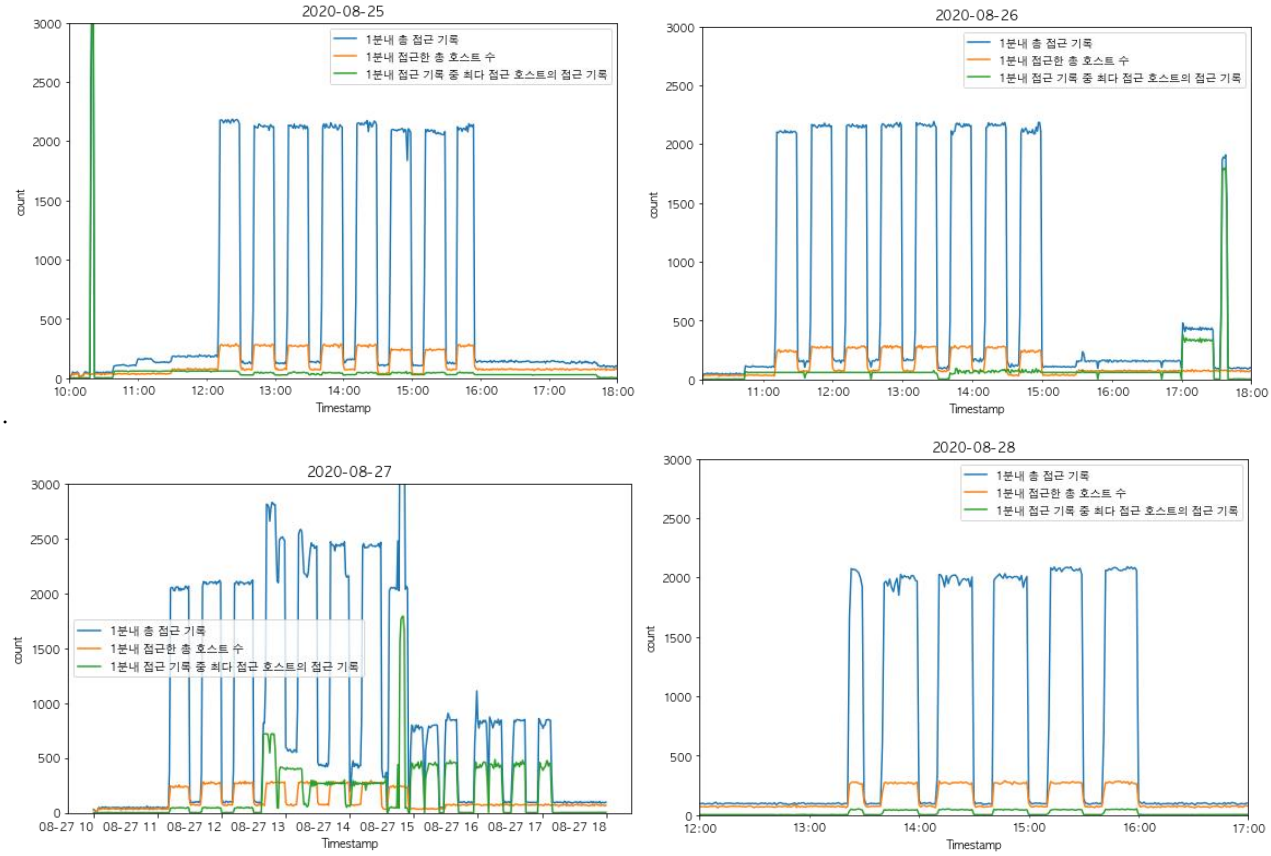
<그림 5-11> 1 분당 접근 횟수, 가장 많이 접근한 Host 의 접근 횟수, 개별 Host 의 수

개별 Host 가 증가하는 시간대마다 전체 카운트도 늘어나는 양상을 보인다. 반면 분당 가장 많이 접근한 Host 의 카운트가 특별히 높게 나타나는 지점에는 그 Host 가 접근한 카운트와 전체 카운트의 수가 비슷하다는 사실을 알 수 있다. 보다 자세한 분석을 위해 특정 포인트를 일별로 나누어 보았다.



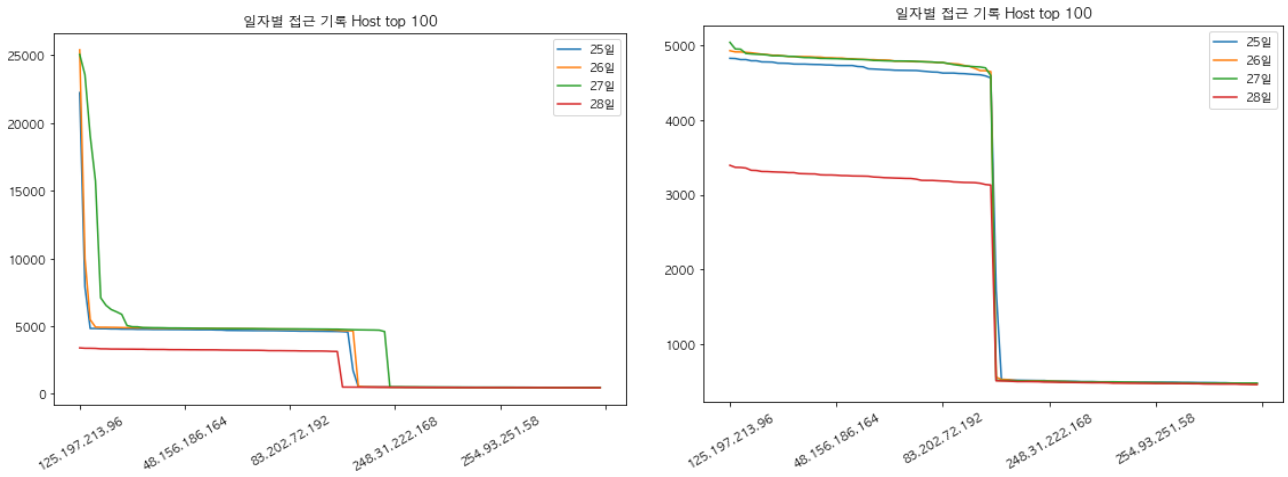
<그림 5-12> 분당 접속 횟수, 가장 많이 접근한 Host 수, 접근한 개별 유니크 Host 의 수

개별 Host 의 수가 평균 270 개 이상으로 증가하는 포인트는 일자별로 양상이 조금씩 다르다. 24 일은 개별 Host 의 수가 증가하는 패턴은 보이지 않았고 분당 카운트와 분당 가장 많이 접근한 Host 의 카운트가 거의 일치한다는 것을 알 수 있다



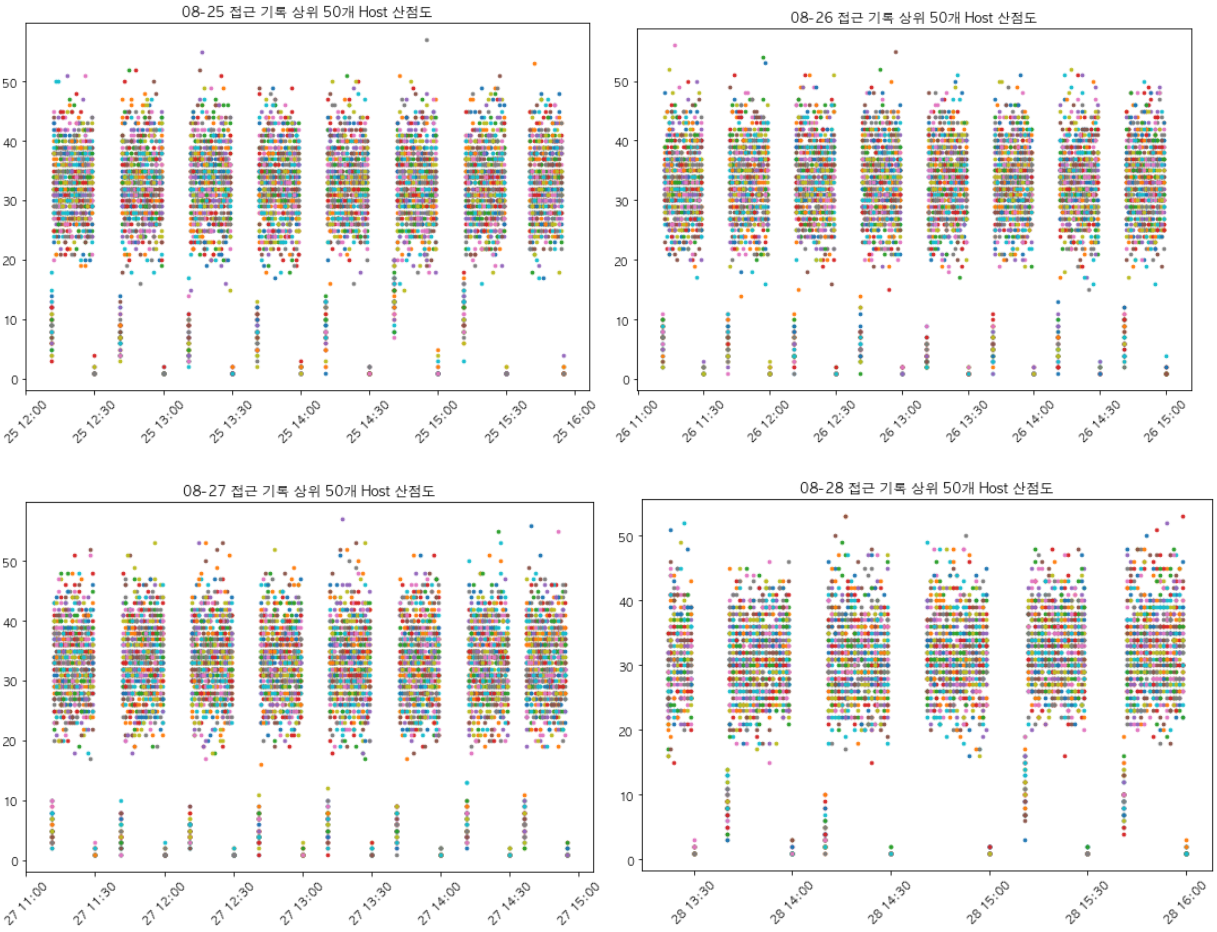
<그림 5-13> 일별로 나눠본 1 분 단위 접속 횟수, 1 분 단위 가장 많이 접근한 Host 수, 1 분 단위 접근한 개별 유니크 Host 의 수

25~28 일은 비슷한 양상의 패턴을 보인다. 1 분 단위 가장 많이 접근한 Host 의 카운트(브루트 포스 및 크리덴셜 스터핑)가 1 분 단위 카운트의 대부분을 차지하는 경우를 제외하면 개별 Host 의 수가 증가함에 따라 1 분 단위 카운트가 확연히 늘어난다는 사실을 알 수 있다. Host 의 수가 늘어나는 패턴과 일자별로 접근기록이 높은 Host 간 상관관계가 있는지 분석해보았다.



<그림 5-14> 크리덴셜 스터핑 및 브루트 포스 일으키는 상위 100 개 Host 일자 별 접근기록, 크리덴셜 스터핑 및 브루트 포스를 일으키는 Host 를 제외한 상위 100 개 Host 일자 별 접근기록

이미 비정상적인 Host(크리덴셜 스터핑 및 브루트 포스)로 가정한 Host 를 제외한 상위 50 개의 Host(25 일~28 일까지 일정하게 50 개)가 나머지 Host 보다 평균적으로 높은 수치를 기록하고 있다. 일자별로 도출한 상위 50 개의 Host 를 전체 데이터에 대입하여 산점도를 찍어보았다.

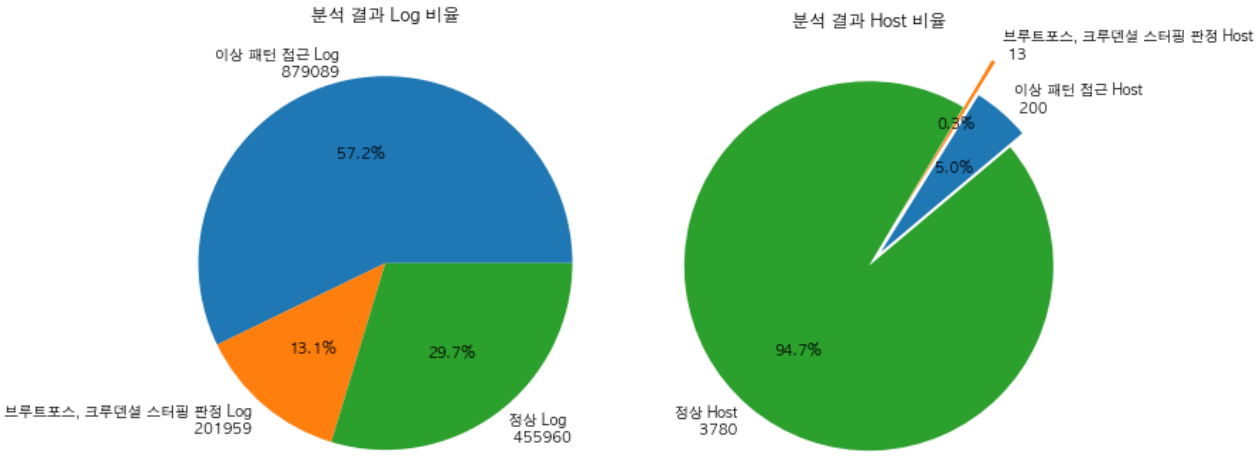


<5-15> 일자 별 접근 기록 상위 50 개 Host 산점도

단 하나의 Host 도 다른 시간대에서 나타나는 현상은 없었다. 정확하게 위에서 발견된 Host 가 증가하는 패턴과 일치하는 시간대에 분포한 모습을 보인다. 마치 자로 잰 듯 10 분 간격으로 활동이 없고 20 분동안 활동한 기록을 남긴다. 이런 패턴이 나타나는 총 200 개의 Host 들로 다른 컬럼을 통해 도출할 수 있는 패턴을 찾아보려 했으나 나타나는 패턴은 없었다.

그러나 시계열 상에서 나타나는 이러한 특수한 현상은 공격자가 마음만 먹으면 헤더에 정보(Host, Referer, UA)를 조작하여 접근을 시도하는 가능성이 있으므로 예의주시하는 것이 좋다.

○ 인위적 패턴을 보이는 접근 그룹



<그림 5-16> Log 와 Host 비율로 보는 분석결과

이상 패턴 접근이 전체 로그의 57.2%를 차지하고 있다. <그림 5-15>를 통해 인위적인 시간 패턴을 보이며 정상 Host 의 접근 기록과 10 배 이상차이나는 200 개의 Host 그룹을 묶어 가용성 침해를 위한 이상행위로 분석했다.

6. 이상 탐지 데이터

6.1. Web Scraping

<목차 4.4.1>과 <목차 5.1>에 근거하여 '63.15.41.131', '137.8.46.133' Host들을 웹 스크래핑 공격행위자로 분석했다. 도출해낸 패턴은 다음과 같다. Referer가 'nan', '-'과 같은 정상적인 루트가 아니거나 연속된 로그의 접속 시간차가 6초, 2초와 같이 일정 시간동안 주기적으로 유지된다.

6.2. Credential Stuffing

<목차 4.4.1>, <목차 4.4.2>와 <목차 5.1>에 근거하여 '100.200.156.222', '112.112.181.134', '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' Host 들을 크리덴셜 스터핑 공격행위자로 분석했다. 도출해낸 패턴은 다음과 같다. 로그인을 성공하지 못하고 여러 번 시도, Payload 의 'log='와 'pwd='가 계속 바뀌며 개별 Host 의 접근 시간차 평균이 1 초 내외, Host 의 전체 로그 중 로그인 페이지 '/wp-login.php'에 접근하는 비율이 95%이상이다.

6.3. Directory Guessing Brute Force

<목차 4.4.2>과 <목차 5.1>에 근거하여 '231.211.11.16' Host 를 디렉터리 추측 브루탈 포스 공격 행위자라고 분석했다. 도출해낸 패턴은 다음과 같다. 접근 시간차 평균이 1 초 내외로 로그인 권한 없이 관리자 페이지와 같은 특정 권한이 필요한 사이트에 비정상적 주기로 접근한다.

6.4. xmlrpc.php 취약점 공격, 무결성 공격

<목차 5.1.2>, <목차 5.2.2>에 근거하여 '101.224.32.28' Host 를 xmlrpc.php 취약점 공격 및 무결성 공격 행위자라고 분석했다. 도출해낸 패턴은 다음과 같다. 짧은 시간 내에 WordPress 관리자 페이지와 xmlrpc.php 에 대량으로 접근을 하였으며 다른 사용자들에 비해 비정상적으로 높은 Byte 의 데이터에 접근했다.

7. 향후 계획

지금까지는 <표3-1>의 3가지 가설에 따라 Host 별로 통계를 내어 <목차 6>의 공격 행위자를 추출했다. 앞으로의 기간 동안에는 앞서 분석한 결과를 바탕으로 하여 공격행위자별 데이터 간의 상관분석과 군집분석을 통해 각 데이터의 유사성과 의미 있는 데이터 패턴을 파악할 것이다. 또한 머신 러닝 기반의 분류 학습 모델을 만들어 정확도를 측정하고 모델을 평가할 것이다.

[붙임] 참고문헌

[1] Celia Paulsen, Patricia Toth, Small Business Information Security: The Fundamentals, November, 2016

[2] Shuchismita Biswas, Virgilio A. Centeno, Chair Vassilis Kekatos Jaime De La Reelopez Seemita Pal, Understanding the Impacts of Data Integrity Attacks in the Context of Transactive Control Systems, December, 2018

[3] Tomi Ruha, Cybersecurity of Computer Networks, April, 2018

[4] https://en.wikipedia.org/wiki/Credential_stuffing

[5] https://owasp.org/www-community/attacks/Forced_browsing

[6] https://docs.sucuri.net/definitions/attacks/brute-force/directory-guessing-brute-force-attacks/

[7] "WordPress Files and Directory Structure - Interserver Tips." 22, September, 2016, https://www.interserver.net/tips/kb/wordpress-files-directory-structure/.

[8] "XpressEngine Core: File List - GitHub Pages." http://xpressengine.github.io/xe-manual-api/html/files.html.