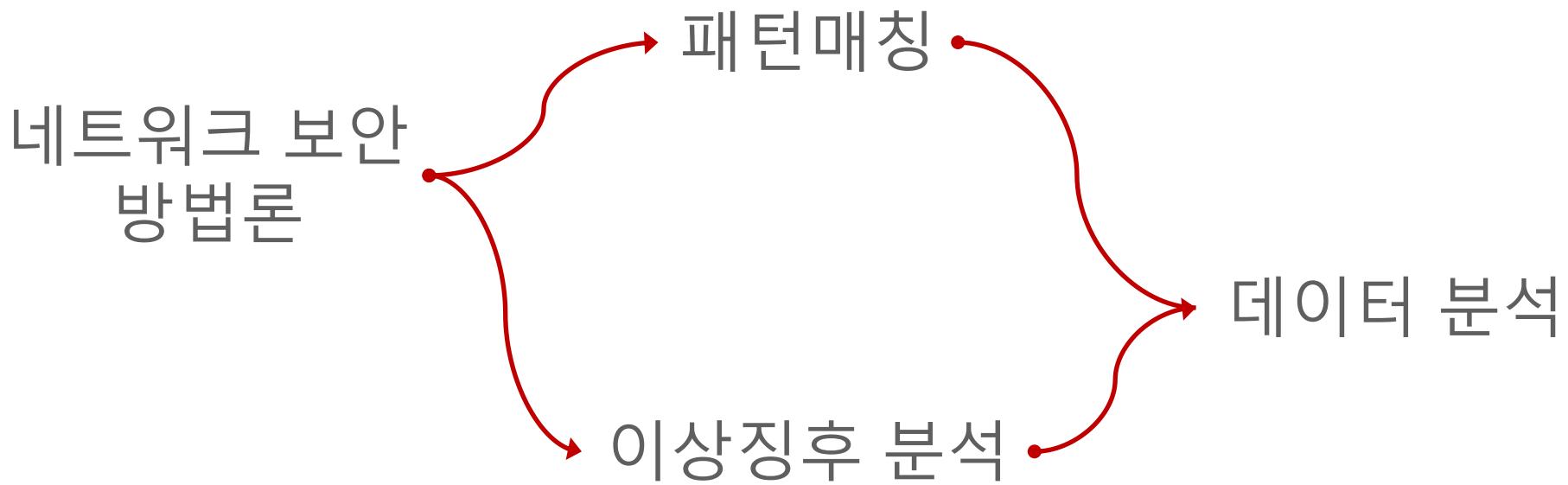


# 네트워크 보안



# 네트워크 보안

▶ 주제



# 네트워크 보안

## ▶ 강사

- 강명훈



4년 간의 시스템 엔지니어  
경험만으로 용케 아직까지  
정보보안 분야에 종사 중.  
  
사고 분석이 아닌 데이터  
분석 관점의 보안 솔루션  
활용 방안에 관심 많음.

- 이메일 : [mhkang589@gmail.com](mailto:mhkang589@gmail.com)
- 블로그 : [kangmyounghun.blogspot.com](http://kangmyounghun.blogspot.com)

### 빅데이터 분석으로 살펴본 IDS와 보안관제의 완성

파坦 해지 기법을 이용한 정보보안 감시체계의 완성

김명훈 저



### 데이터 분석이 쉬워지는 정규표현식

빅데이터·인공지능 시대의 데이터 길들이기



### ElasticSearch로 알아보는 이상징후 분석

Data Anomaly Detection



### SPLUNK로 알아보는 이상징후 분석

Data Anomaly Detection

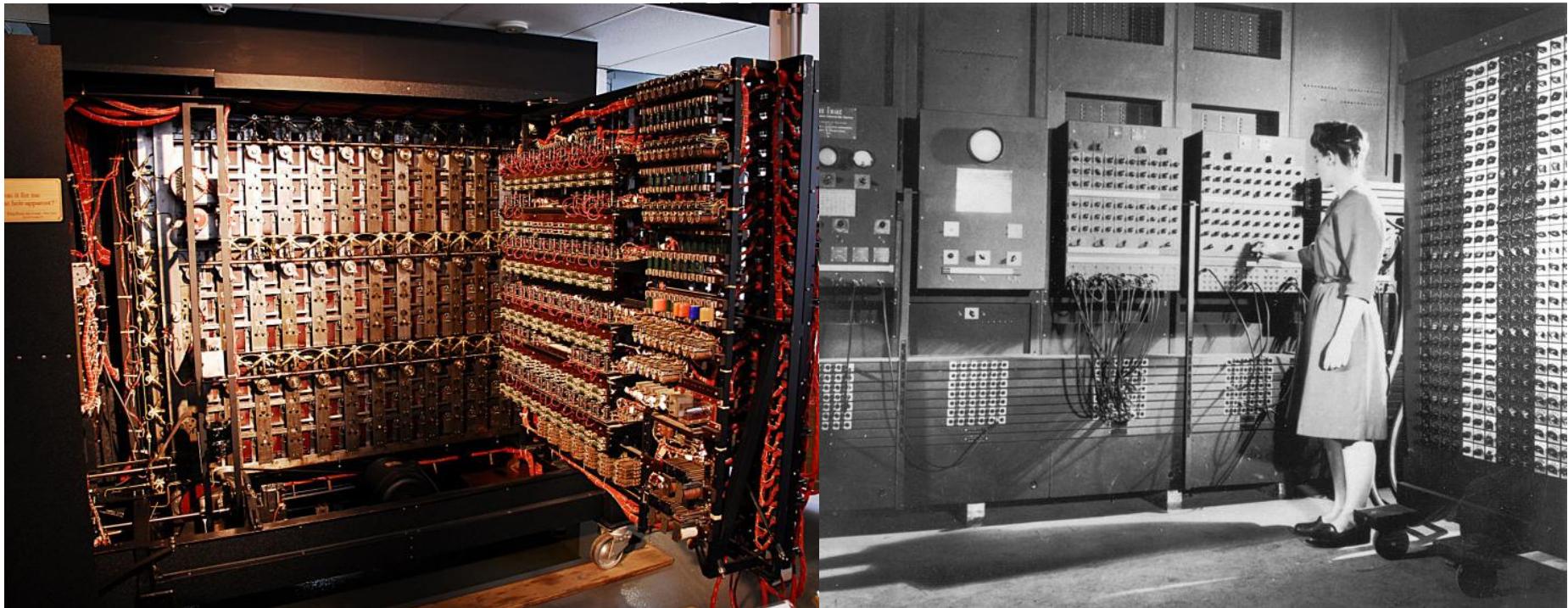


# 네트워크 보안



## ▶ 컴퓨터의 시작(1940s)

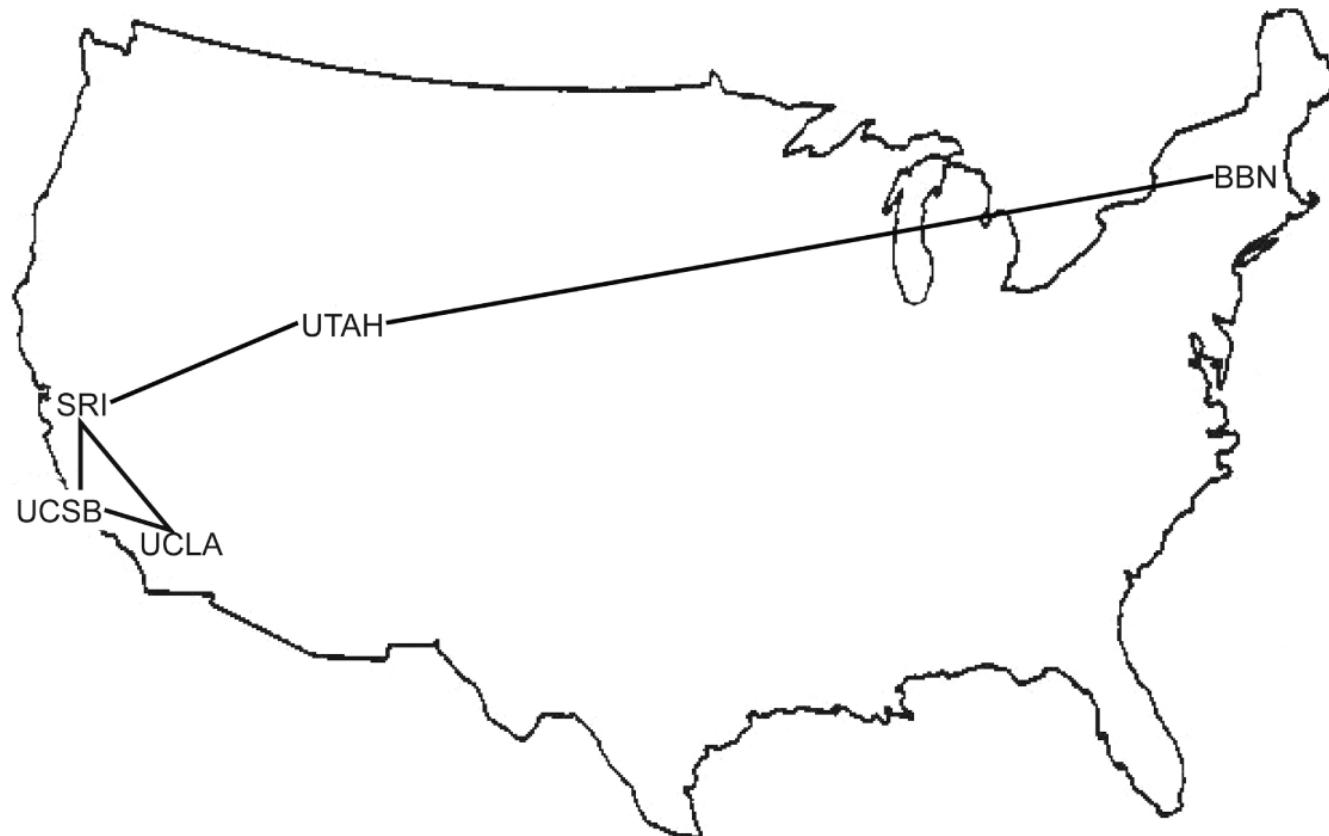
- Bombe와 Eniac



# 네트워크 보안

## ▶ 네트워크의 시작(1960s)

- Arpanet



# 네트워크 보안

## ▶▶ 운영체제의 시작(1970s)

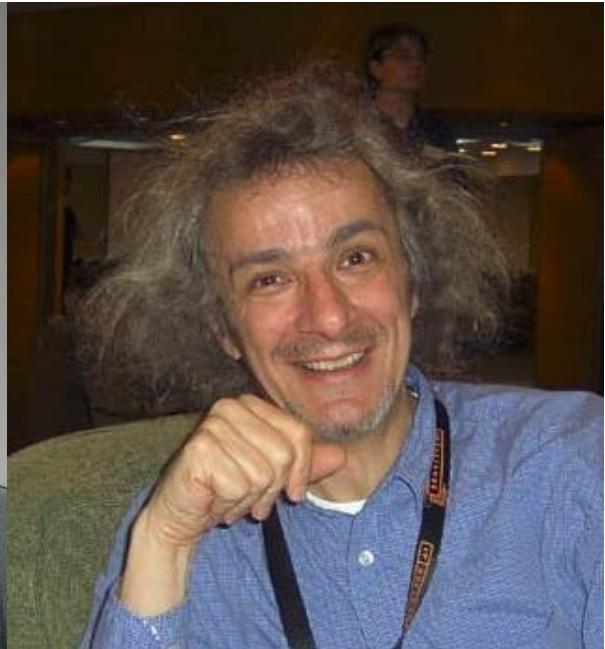
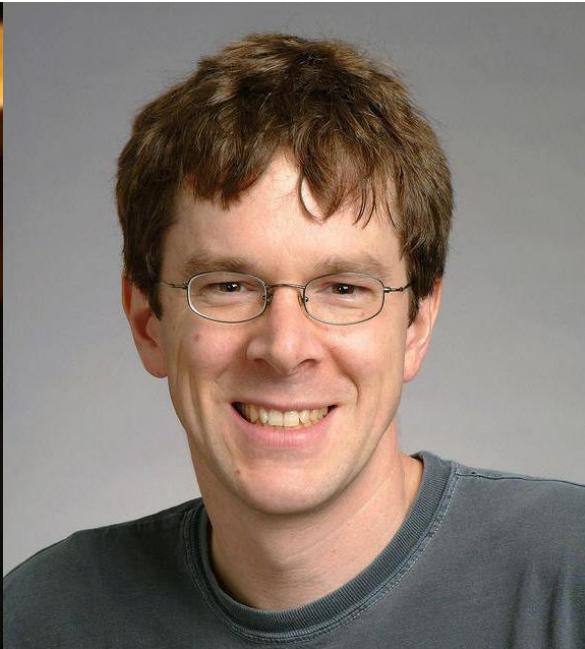
- Unix



# 네트워크 보안

## ▶▶ 해커의 시작(1980s)

- 캐빈 미트닉, 로버트 모리스, 클리포드 스톤

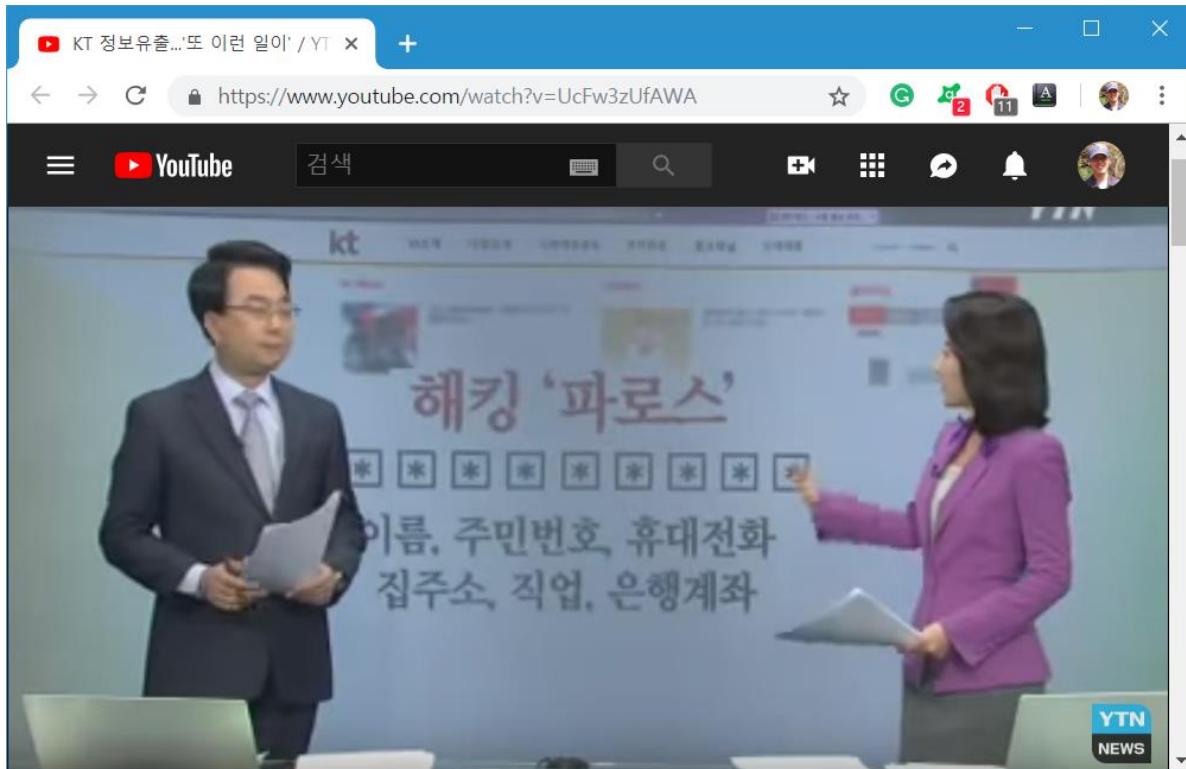


# 네트워크 보안



## ▶ 해킹 사례

- KT 고객정보 유출(2014)
- <https://www.youtube.com/watch?v=UcFw3zUfAWA>

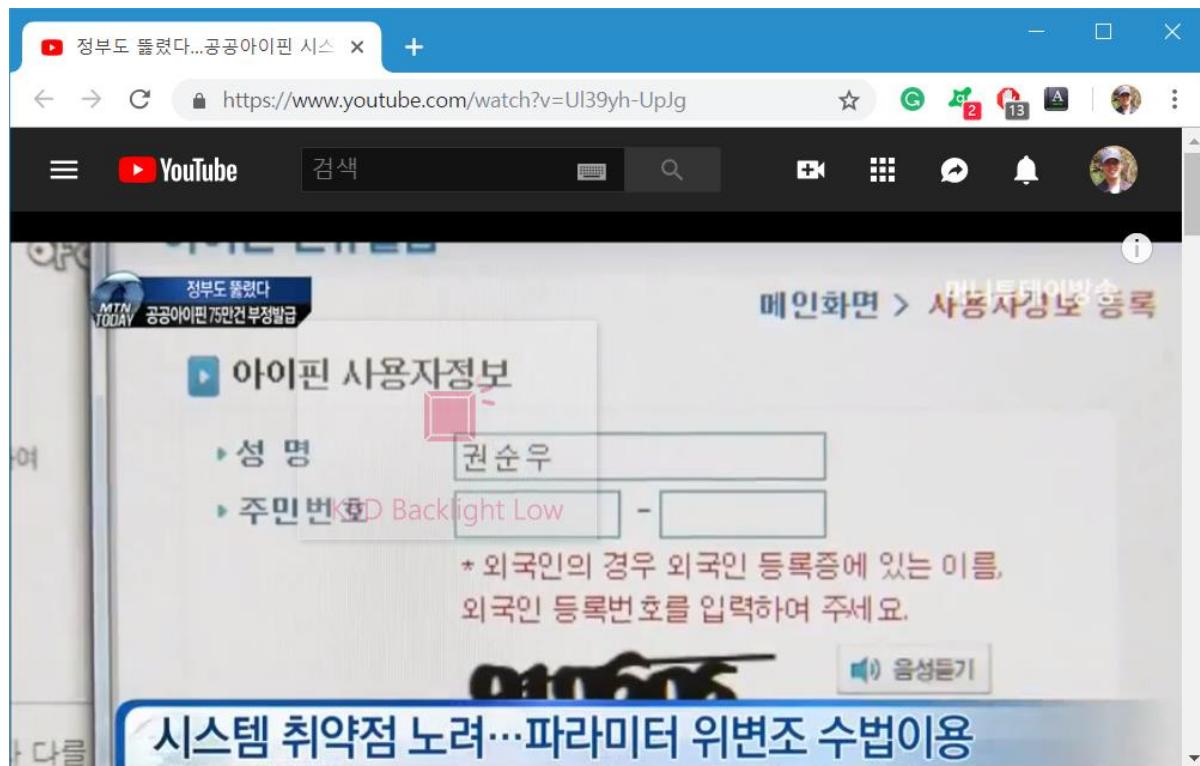


# 네트워크 보안



## ▶ 해킹 사례

- 아이핀 부정발급(2015)
- <https://www.youtube.com/watch?v=Ul39yh-UpJg>

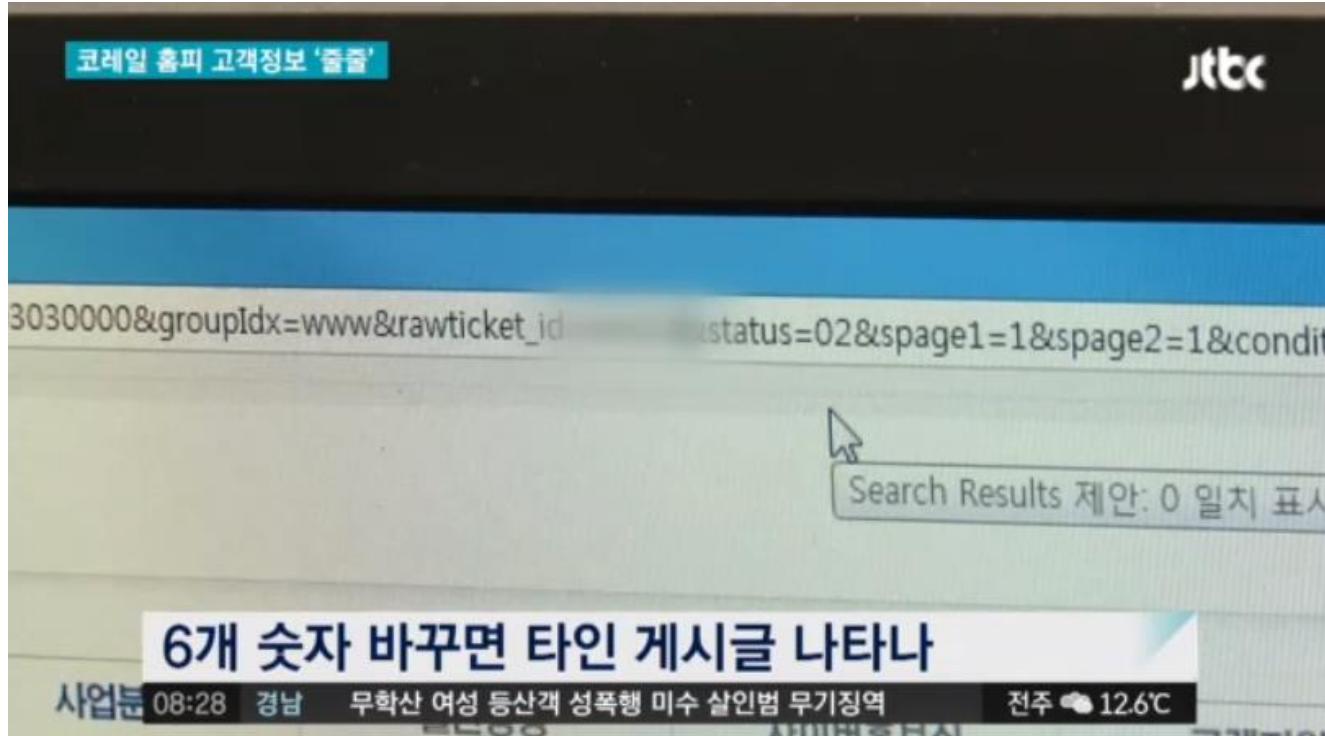


# 네트워크 보안



## ▶ 해킹 사례

- 코레일 타인 게시글 노출(2016)
- [http://news.jtbc.joins.com/article/article.aspx?news\\_id=NB11332590](http://news.jtbc.joins.com/article/article.aspx?news_id=NB11332590)



# 네트워크 보안



## ▶ 해킹 사례

- T-모바일 고객정보 유출(2017)
- [https://www.youtube.com/watch?v=3\\_gd3a077RU](https://www.youtube.com/watch?v=3_gd3a077RU)

The screenshot shows a JSON object representing customer data. The structure is as follows:

```
Object {  
  users: [Object],  
  lines: [Object]  
}  
  
users: [Object]  
  0: Object {  
    email: "rafael150813@ymail.com",  
    id: "U-eb71e003-9c15-4421-a630-8d7f1a562810"  
  }  
  
lines: [Object]  
  0: Object {  
    accountStatus: "N",  
    balance: "900186200",  
    customerType: "GSM_HD_P",  
    givenName: "Rafael",  
    lastName: "1312407509510",  
    lineable: "true",  
    number: "19352538993",  
    permissionType: "Owner/User",  
    ...  
  }  
  1: Object {  
    accountStatus: "N",  
    balance: "900186200",  
    customerType: "GSM_HD_P",  
    givenName: "Rafael",  
    lastName: "1312407509510",  
    lineable: "false",  
    number: "19352538993",  
    permissionType: "User/Read",  
    ...  
  }  
}
```

RECORDED WITH SCREENCASTOMATIC

# 네트워크 보안



## ▶ 공통점

- [edaily.co.kr/news/read?newsId=01115206609301680&mediaCodeNo=257](http://edaily.co.kr/news/read?newsId=01115206609301680&mediaCodeNo=257)

[이데일리 이승현 기자] 2월 28일부터 이달 2일까지 행정자치부가 관리하는 공공 아이핀(i-PIN) 시스템에서 75만건의 **아이핀이 부정 발급된 해킹사고는 1년 전 KT(030200)** 가 입자 1200만명의 개인정보가 유출된 사고와 **동일한 수법인** 것으로 나타났다. 이는 내부 시스템의 전송 데이터들에 대한 암호화와 검증절차 등 보완대책이 미비해 발생한 것으로 분석된다.

해커는 이러한 방식으로 **변수 값을 바꿔 인증절차인** △**개인정보 입력**(주민등록번호 등 입력)과 △**본인인증**(공인인증서 등 입력) 단계를 우회해 △**서버인증**을 받아 허위 아이핀을 발급한 것이다. 보안업계 한 관계자는 "서버인증 단계에서 파라미터 값을 받아야 하는데 (변조를 통해) 인증이 성공한 것으로 조작했다고 본다"고 말했다.

이 해킹수법은 지난해 3월 KT 홈페이지에서의 가입자 개인정보 유출사고 때 이용됐다. 해커들은 KT 홈페이지에 프록시 툴인 '파로스' 프로그램을 설치해 내부에서 오가는 정보들을 중간에 가로챘고, 무작위 대입공격(브루트 포스)을 통해 이에 부합하는 고객정보를 빼돌린 것이다.

# 네트워크 보안



## ▶ 파라메터

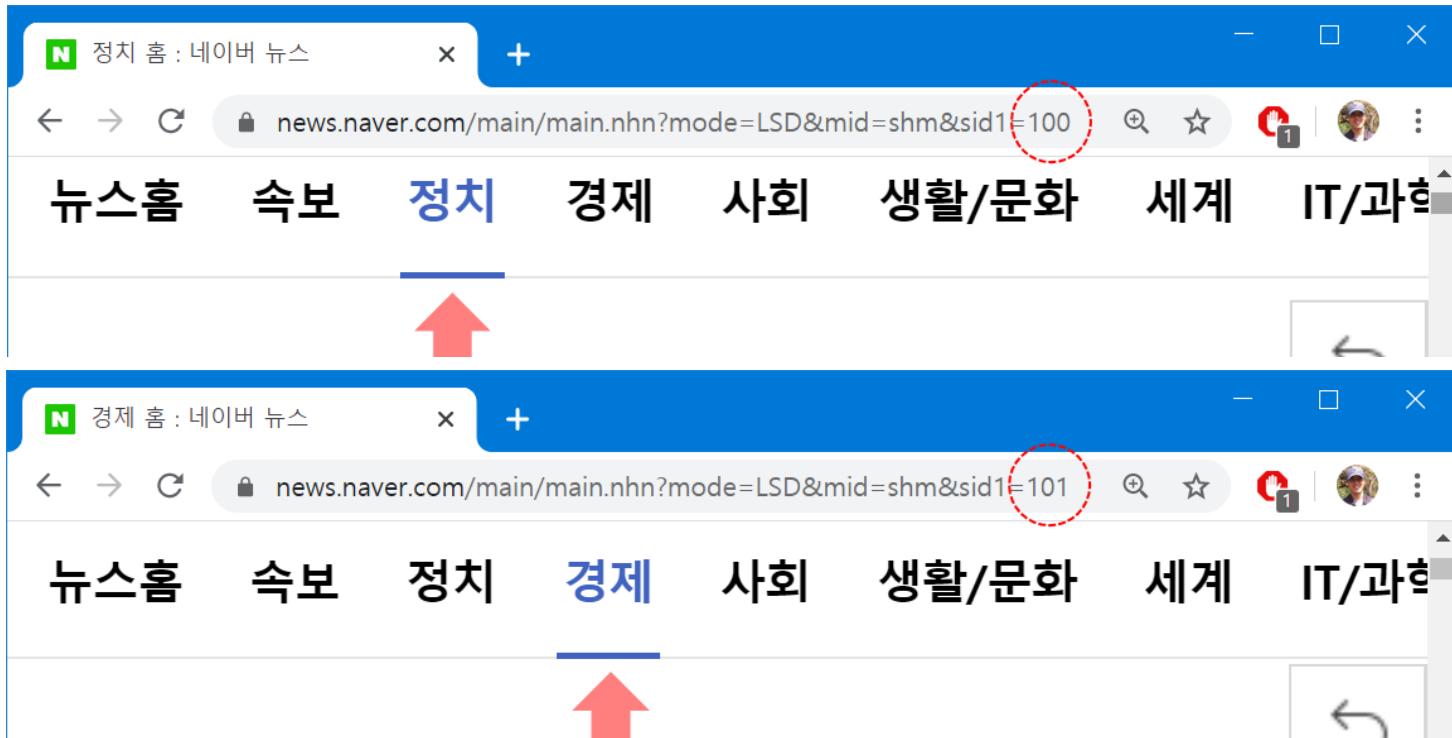
- 파라메터가 바뀌면 정보도 바뀐다



# 네트워크 보안

## ▶ 파라미터

- 파라미터가 바뀌면 정보도 바뀐다



# 네트워크 보안



## ▶ 파라메터

- Burp Suite(<https://portswigger.net/burp>)를 이용한 파라메터 변조

Request to https://search.naver.com:443 [43.250.153.7]

Forward Drop Intercept is on Action

Raw Params Headers Hex

Comment this item

GET /search.naver?sm=top\_hty&fbm=1&ie=utf8&query=superca HTTP/1.1  
Accept: text/html, application/xhtml+xml, image/jxr, \*/\*  
Host: search.naver.com  
Connection: close  
Accept-Language: ko-KR  
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko  
Accept-Encoding: gzip, deflate  
Cookie: npic=cksCbS8qHfE2lo3m/7RxfxWl5CfvbmlkLGwhQKA1/lnAcAjBN+gUdPK6YYz32D0ICA==; nx\_ssl=2; \_naver\_usersession\_=+m4o1cr6Y/QIXISG9deNDQ==; NNB=S0RIMT7H3UJVY

Type a search term 0 matches

# 네트워크 보안



## ▶ 보안의 시작

- 해외: 모리스웜(1988)
- 국내: 1.25대란(2003)

news.bizwatch.co.kr/article/mobile/2018/06/29/003...

국내 민간 분야의 사이버 보안은 2016년부터 최근까지 관심과 주의 단계에서 오가고 있다. 보안 경보 단계는 정상-관심-주의-경계-심각 다섯 단계로 나뉘어지는데 최근에는 우려할만한 수준은 아니라는 얘기다.

원래 KISC는 1990년대만해도 인터넷진흥원(당시 한국정보보호센터) 내의 해킹과 바이러스 상담지원을 담당하는 팀 단위 조직이었다. 2003년 1월25일 국내 인터넷망이 마비됐던 이른바 '1.25 대란'을 겪은 이후 인터넷침해대응지원센터를 개소하면서 조직이 확대됐고 이후 2013년 사이버테러 이후 지금의 본부 단위로 조직을 키웠다.

담당 업무도 1990년대에는 이메일 신고 및 접수 처리 정도였으나 1.25 대란을 겪으면서 상시 모니터링 업무를 맡게 되고 이후에는 디도스 대응과 감염PC 치료, 모바일 응급 사이버 치료 등으로 고도화하고 있다.

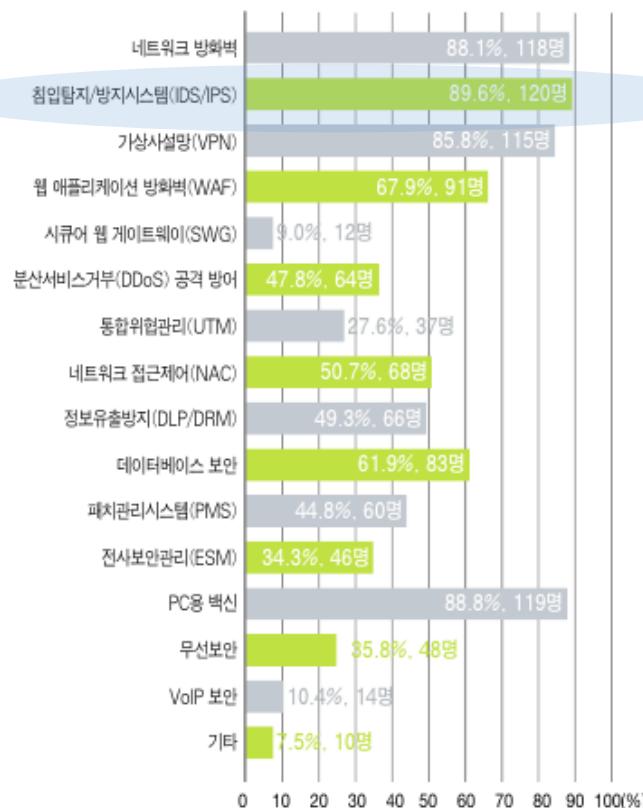
# 네트워크 보안



## ▶ 보안 업무 비중

- <http://pdf.datanet.co.kr/217/217141.PDF> (2011년)

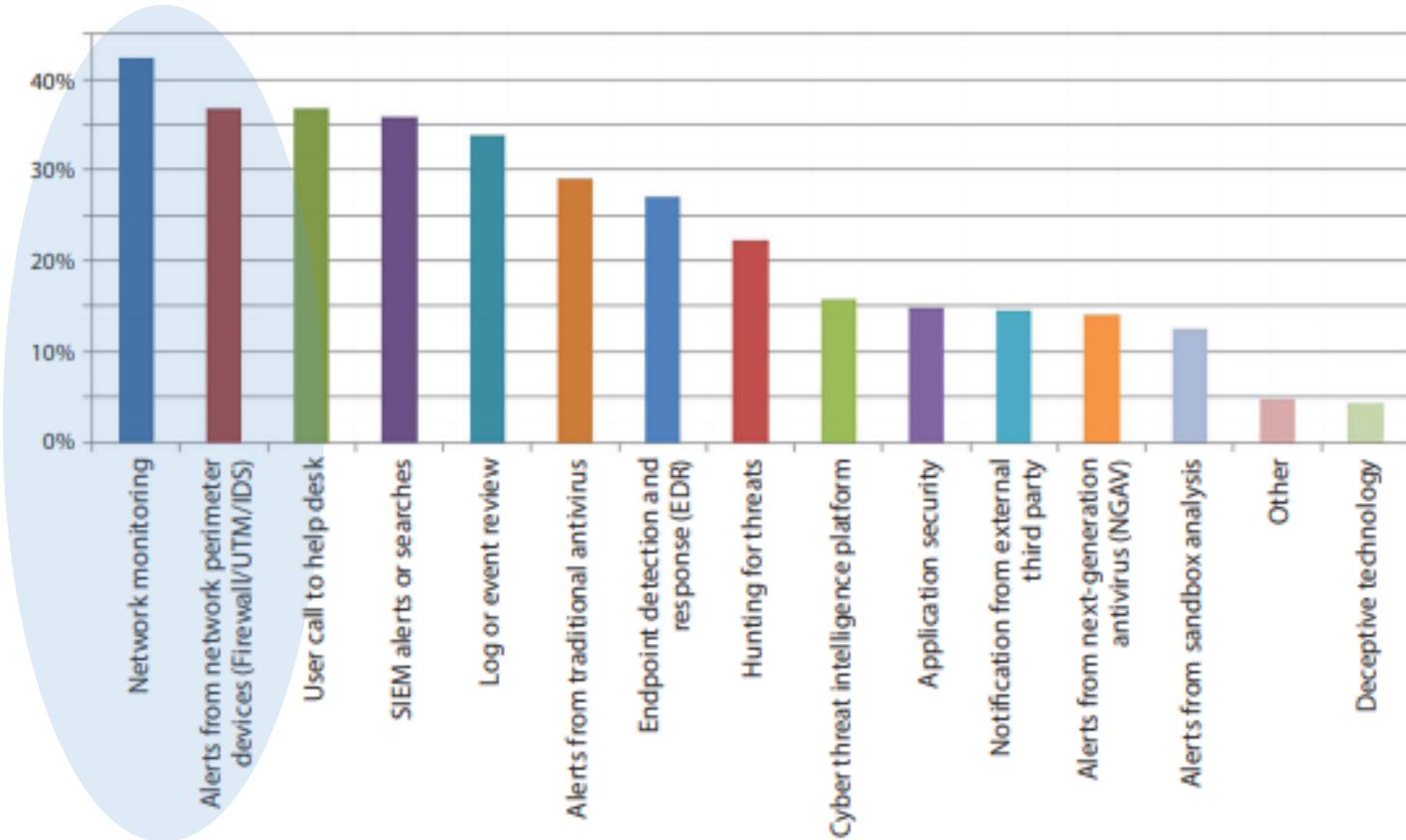
〈그림 8〉 현재 사용하고 있는 보안 제품군은. (100% = 134)



# 네트워크 보안

## ▶ 보안 업무 비중

- <https://www.sans.org/reading-room/whitepapers/threats/2017-threat-landscape-survey-users-front-line-37910>



# 네트워크 보안

## ▶▶ 왜 네트워크 보안?

- 톰 크루즈가 뭘 하고 있나요?



# 네트워크 보안



## ▶ 전세계가 연결된 네트워크

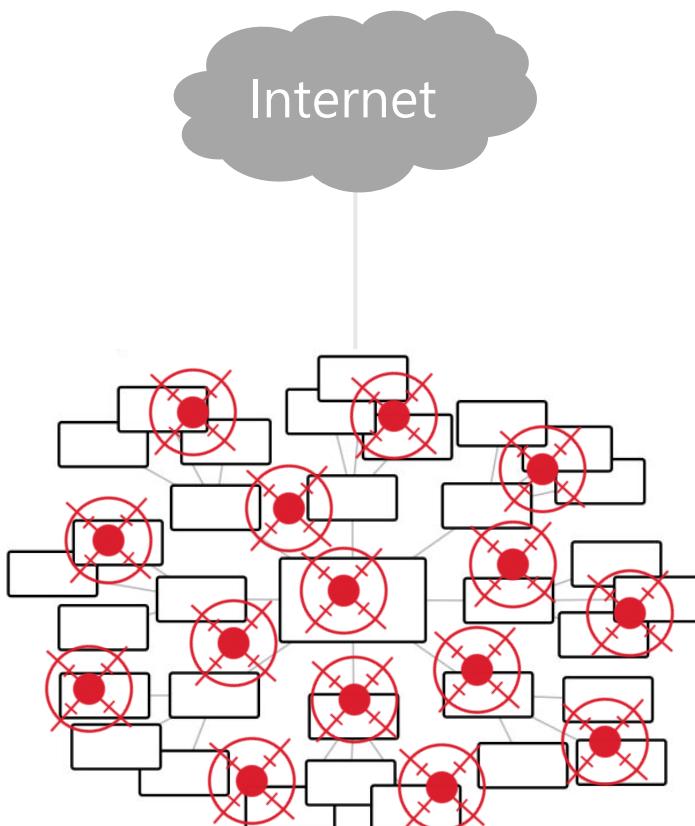
- 물리적 제약이 적다. 그래서 쉬운 해킹
- 피아 식별이 어렵다. 그만큼 어려운 방어



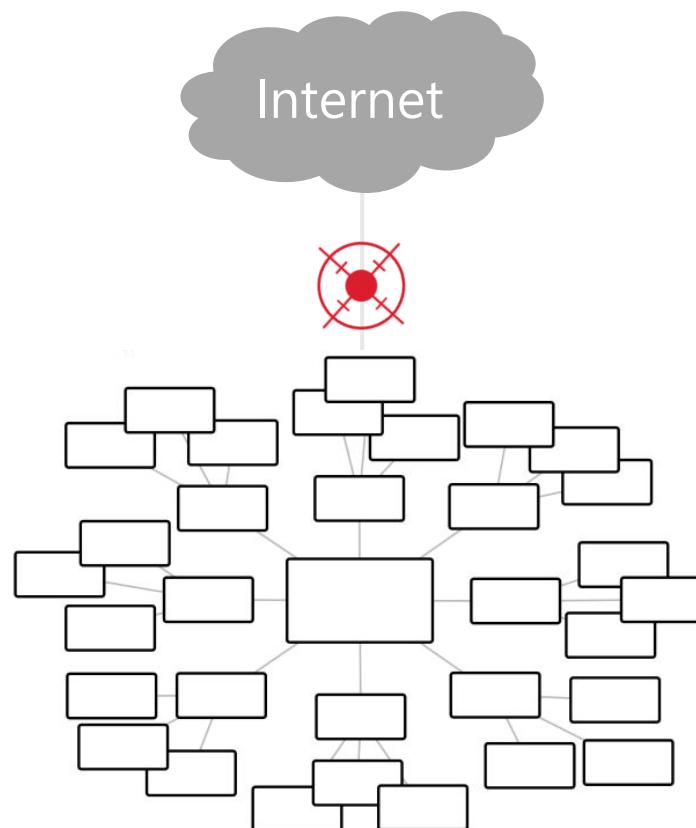
# 네트워크 보안

▶ 그럼에도 불구하고 효율적

- 모든 노드



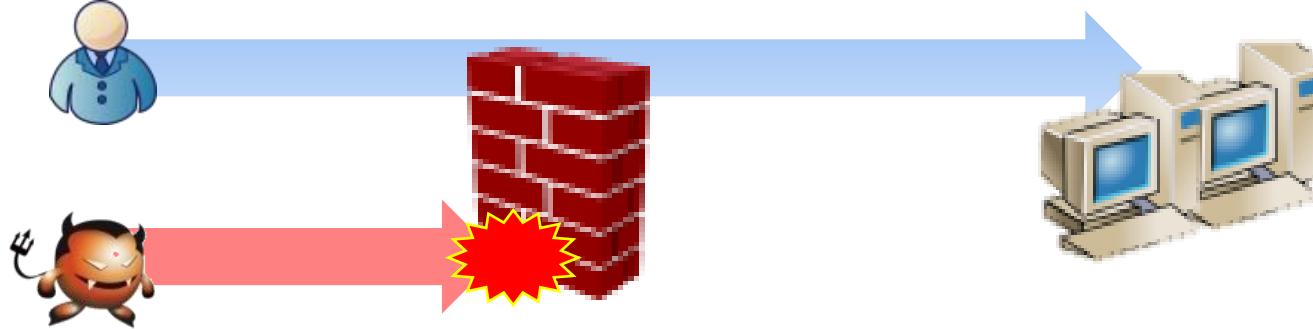
- 모든 데이터가 지나가는 길목



# 네트워크 보안

## ▶ 방화벽

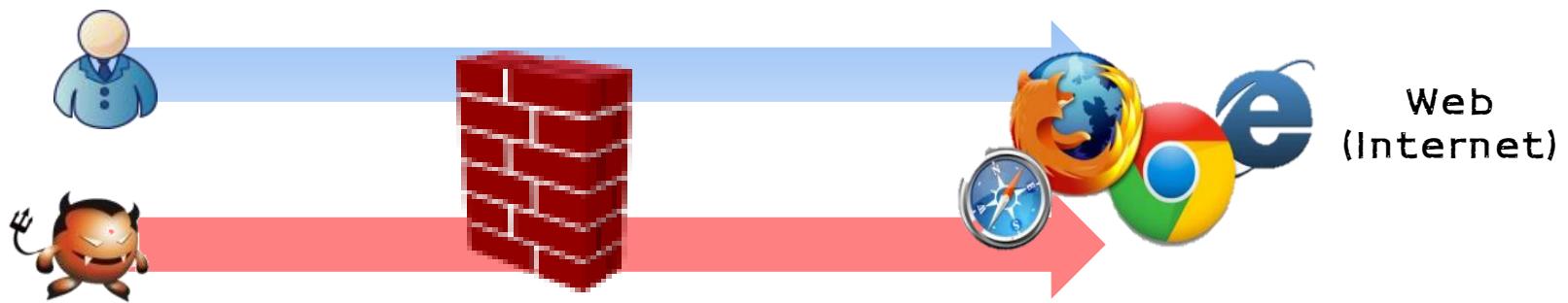
- 체크포인트 Firewall-1(1994년)
- IP/Port 기반으로 동작
- 허용 정책에 포함되지 않는 모든 IP/Port 차단





## ▶ 보안에 취약한 웹서비스

- www.yahoo.com(1995년)
- 방화벽이 차단할 수 없는 웹서비스 등장
- 웹 해킹 등장



# 네트워크 보안



## ▶▶ IDS(Intrusion Detection System)

- IP/Port 상위 개념의 접근제어 필요성 대두
- 애플리케이션 레이어의 데이터 패턴을 검사하는 snort(1998년)
- 패턴매칭!?



Firewall

vs



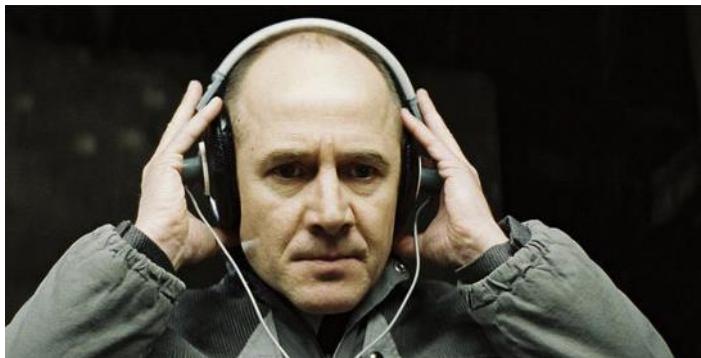
IDS

# 네트워크 보안



## ▶ 패턴매칭

- 인류가 경험한 **가장** 효과적인 데이터 수집/분석 방법



무차별 감시 강화  
프리즘 프로젝트



### 언론을 통해 드러난 NSA 도·감청 기술의 진화

1980, 90년대

- 전화기나 건물 내부에 삽입 또는 부착하는 도·감청 기기 활용
- 이동식 차량 활용해 도청 대상 근거리서 무선 감청
- 레이저와 전자파, 위성기술 활용해 목소리 도청
- 복제폰 만들어 실시간 도·감청
- 도청 위해 외국 대사관 지하에 비밀 터널 뚫거나 비둘기 발목에 소형 장치를 매달아 도청 대상 주변에 날리기도



2000년대 이후

- 첩보위성 및 무인기 활용해 전 세계 도·감청
- 해저케이블에 도·감청 장비 장착
- 슈퍼컴퓨터 이용해 암호 분석
- 특수 프로그램 통해 특정 단어 중심으로 문자메시지 통화내용 분석, 스마트폰 암호화 기능도 무력화



# 네트워크 보안

## ▶ 패턴매칭 특징

- 주고 받는 패턴으로 상호관계 파악 가능

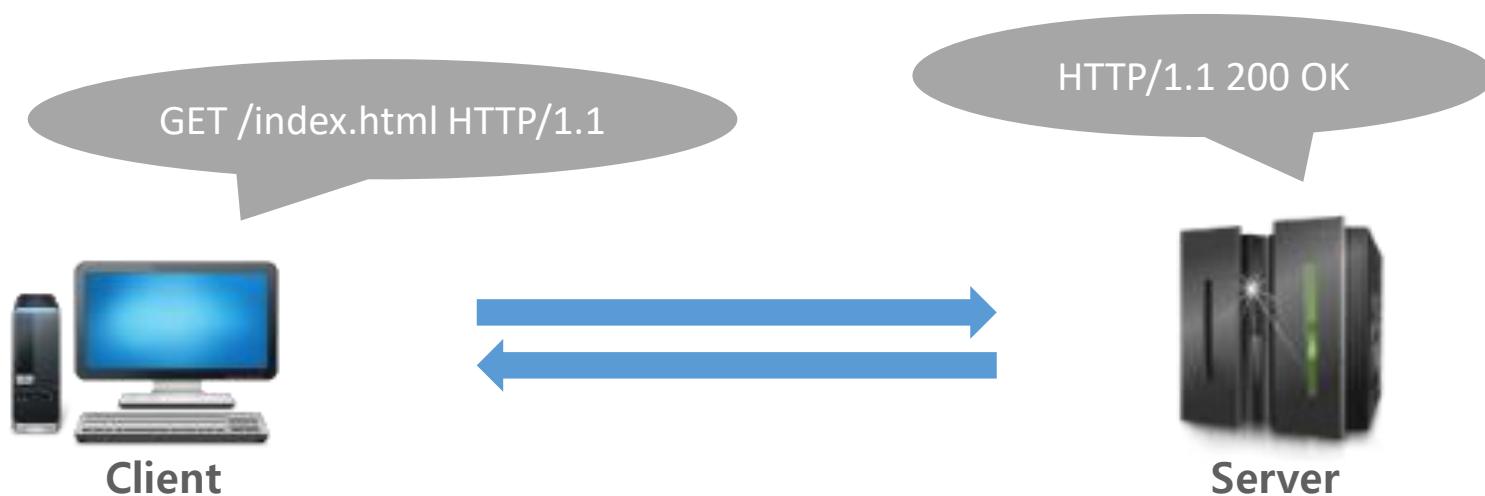


# 네트워크 보안



## ▶ 패턴매칭 특징

- 사람과 똑같은 방식으로 소통하는 컴퓨터
- 주고 받는 패턴으로 정상/비정상 파악 가능



# 네트워크 보안



## ▶ 침입탐지 모델

- An Intrusion Detection Model(1986년, Dorothy Denning)
- 패턴매칭/이상징후 분석

	패턴매칭(오용행위 분석)	이상징후(비정상행위 분석)
개념	정의된 <b>패턴</b> 을 갖는 공격만 막는다	정의된 <b>패턴</b> 을 갖는 정상만 허용한다
장점	(알려진) 공격 정의가 쉽다	정상만 정의하면 된다
단점	알려지지 않은 공격은?	정상 범위를 정의하기 어렵다
구현 예시	IDS, IPS (Black list)	방화벽, 웹방화벽 (White list)

- 오용(공격) 패턴을 찾는 패턴매칭 중심으로 발전
- 이상징후 분석 일부 수용? (패킷 임계치 초과 트래픽 탐지 등)

# 네트워크 보안

## ▶ 패턴매칭의 장점

- 쉽다

“개발이 용이하며, 개념적으로도 패킷에서 패턴을 찾는 것이 더 이해하기 쉬운 방법”

- 네트워크 보안 실무 (689 페이지)



RICHARD BEJTLICH  
Foreword by RON GULA,  
CTO, Tenable Network Security

- 검사 범위를 좁힐 수 있다

## 네트워크 트래픽

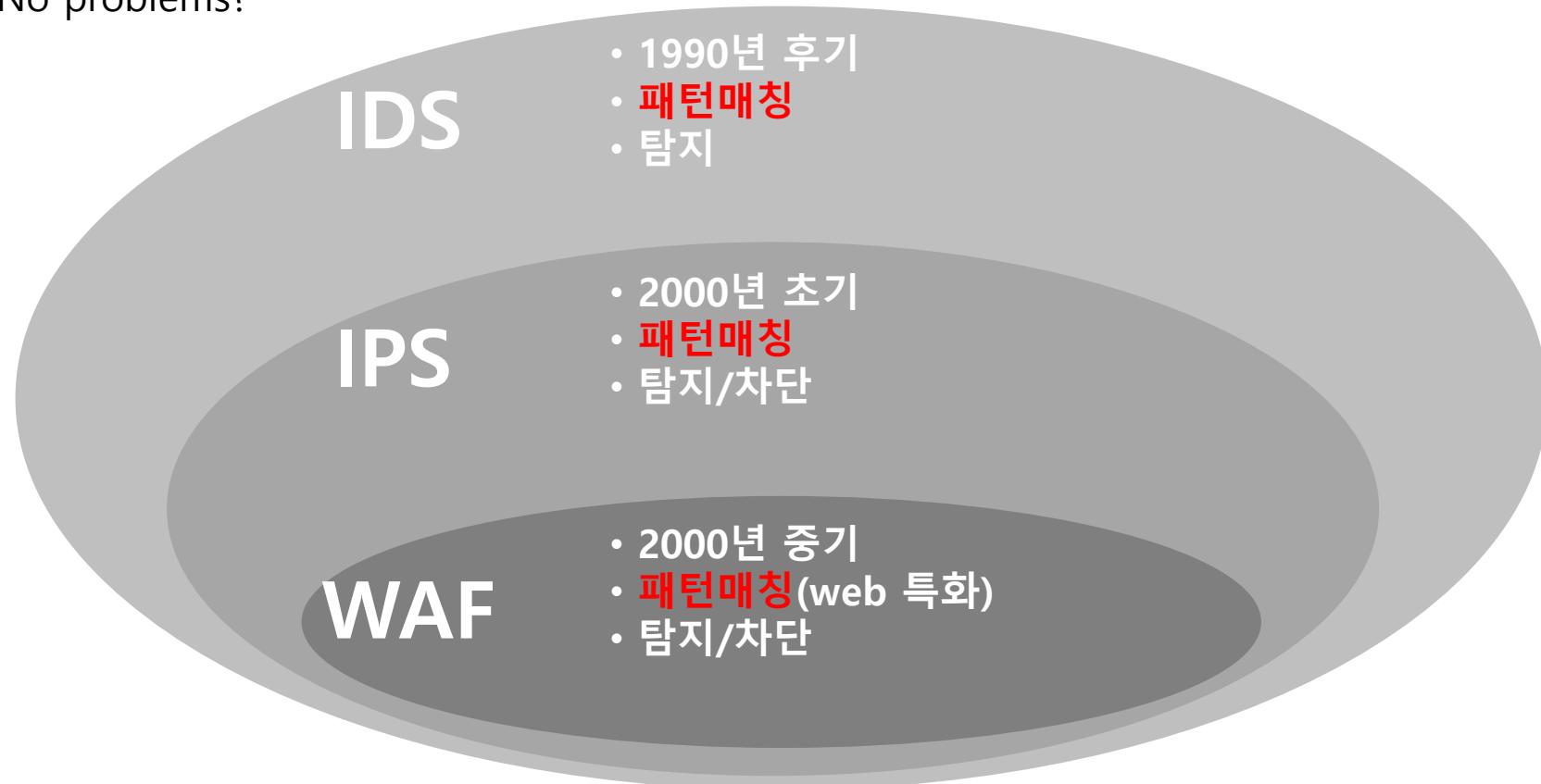


# 네트워크 보안



## ▶ 쉽고 범용적인 패턴매칭

- 네트워크 보안 기반 기술
- No problems?



## ▶ 오탐 문제 대두

- [http://web.archive.org/web/20031204111139/http://www.gartner.com/5\\_about/press\\_releases/pr11june2003c.jsp](http://web.archive.org/web/20031204111139/http://www.gartner.com/5_about/press_releases/pr11june2003c.jsp)

◀ ▶ C i web.archive.org/web/20031204111139/http://www.gartner.com/5\_about/press\_releases/pr11june2003c.jsp to the Gartner Information Security Hype Cycle research.

according to the Gartner, Inc., press release dated June 11, 2003, "The market for intrusion detection systems (IDS) has stalled, and vendors are now hyping intrusion prevention systems (IPS)."

**WayBack Machine** INTERN ARCHIVE will be obsoleted by the Wayback Machine. 38 cache promised by vendor 15 6월 03 - Go

**False positives and negatives**

An increased burden on the IS organization by requiring full-time monitoring (24 hours a day, seven days a week).

Gartner recommends that enterprises consider the following factors when evaluating IDS products: those offered by thought-leading firewall vendors (such as Cisco, Juniper Networks, and Fortinet) and integrated products.

"Intrusion detection systems are a market failure, and vendors are now hyping intrusion prevention systems, which have a long way to go before they can compete with firewalls," said Richard Stiennon, research vice president for Gartner. "Functionality is moving into firewalls, which are adding deep packet inspection for content and malicious traffic blocking, as well as antivirus activities."

According to the Gartner Information Security Hype Cycle research, some of the problems associated with IDS include:

- False positives and negatives
- An increased burden on the IS organization by requiring full-time monitoring (24 hours a day, seven days a week)

# 네트워크 보안



## ▶▶ 이슈들

- <http://symantec.com/connect/articles/evolution-intrusion-detection-systems>

Haystack 프로젝트(1988년)

*“searching through this large amount of data for one specific misuse was equivalent to looking for a needle in a haystack”*

- <http://www.boannnews.com/media/view.asp?idx=50128>

“증거값 시스템이 완벽한 건 아니다. 일단 경보가 대단히 많이 발생한다. 경보만 처리하고 분석하는 데에 사실 보안팀이 총력을 기울여야 될 정도라, 다른 일을 할 수가 없다. 증거값 기반 시스템이 ‘좋기는 하나 부담스러운’ 이유”

- <http://www.boannnews.com/media/view.asp?idx=37409>

“수십만, 수백만 라인의 로그를 하나하나 꼼꼼하게 확인하려면...”

# 네트워크 보안

▶ 검사 범위를 좁혔는데 (오탐) 로그는 왜 많을까?

- 도청이 성공하려면?



# 네트워크 보안



## ▶ 검사 범위를 좁혔는데 (오탐) 로그는 왜 많을까?

- 트래픽의 특성 : 수십 개 수준의 문자 기호로 수십만 이상의 문자열 조합 가능

### 한국경제

구독하기 오늘의신문 보기 이용자 한마디 01-02 16:30 편집

#### 女앵커와 불륜 '재벌 2세' 누군가 했더니…

자동차 뺑소니 사고로 사람을 죽이고, 방송사 앵커와 불륜을 맺고도 죄의식을 느끼지 않는…재벌가 안주인은 아들과 공모해 정치자금으…

OPINION 저절 정치 바꾸려면 국민이 정치를 끊어야 -사설-



#### 정몽구 회장 "車 연비 높이고 안전·성능 강화해야"

정몽구 현대자동차그룹 회장이 새해 신년사에서 "전 부문의 역량 강화를 통해 미래 성장의 기반을 조성하는 한 해로 만들어 가자"고 강조했다 정…



#### 빛에 조들린 30대女 "집이 맘해서, 알바 전전하다…" 경악한 이유가

女직원 "신입사원 연수 마지막 날 회장님의 부르더니…" 폭탄 고백



이건희 회장, 이부진 손 잡고 신년식 참석

이건희 삼성 그룹 회장 및 후계구도에 올라있는 삼성가(家) 3 남매가 일제히 그룹 신년하례식에 모습을 드러냈…

#### 핫 검색 이 가게 술 마시러 지하철역엔 왜? 이색 다이닝펍 코다차야

집권 2년차 대통령 '입'에 쏠린 눈…청와대 신임 대변인 누가 될까?

#### HOT NEWS

'차승원 아들' 차노아, 성폭행 女 직접 만나…깜짝

MBC 진짜사나이, 병사들 데리고 거짓 방송?

女아이돌 19금 도촬? 신체 일부를 은밀히…

수지, 방송 중 깜짝 발언에 선배들…'싸늘'

부하 女직원에 "임신했으면…" 막말 충격

10대 女조카 성폭행에 임신 시키더니…경악

女 민감 부위, 20대 중반부터 점점 변하더니…

오늘의 화제



비, 김태희와 결혼 질문에 "생각 없다"



강소라, 생방송 중 과감한 춤에 가슴이

이다도시, 이혼 후 생활고에…'충격'

[한국경제 사이트 바로가기 ▶](#)

# 네트워크 보안



## ▶▶ 검사 범위를 좁혔는데 (오탐) 로그는 왜 많을까?

- 네트워크 보안 실무(2004년, 미국)

“대량 경보에 의한, 대량 정보가 수집되면 이 정보의 홍수 속에 정말 중요한 정보가  
묻힐 가능성이 커진다. 이러한 대량 경보 발생 공격은 잘 만들어지지 않은 IDS  
시그니처가 그 원인인 경우가 대부분” (675 페이지)

- IDS와 보안과제의 완성(2013년, 한국)

“체계적인 룰 정확도 향상이 이루어지지 못하면서 부정확한 로그가 대량 발생하게  
되고, 분석 인력이 대량 투입되면서도 분석하지 못하는 로그는 오히려 더 늘어나는  
악순환이 되풀이” (453 페이지)

# 네트워크 보안



## ▶ 해법

- chatGPT에게 물어봤습니다

New chat x +

chat.openai.com/chat

+ New chat

IPS Challenges

Network Security Challenges

Quantum Computing Basics

Clear conversations

Is there a solution for false positives in intrusion prevention system?

Intrusion Prevention Systems (IPS) are designed to detect and prevent malicious network activity. False positives can occur when an IPS mistakenly identifies legitimate network traffic as malicious, resulting in unnecessary alerts and potentially blocking legitimate traffic. False positives can be caused by a number of factors, such as misconfigurations, outdated signatures, or errors in the IPS rules.

There are several strategies that can help reduce false positives in IPS:

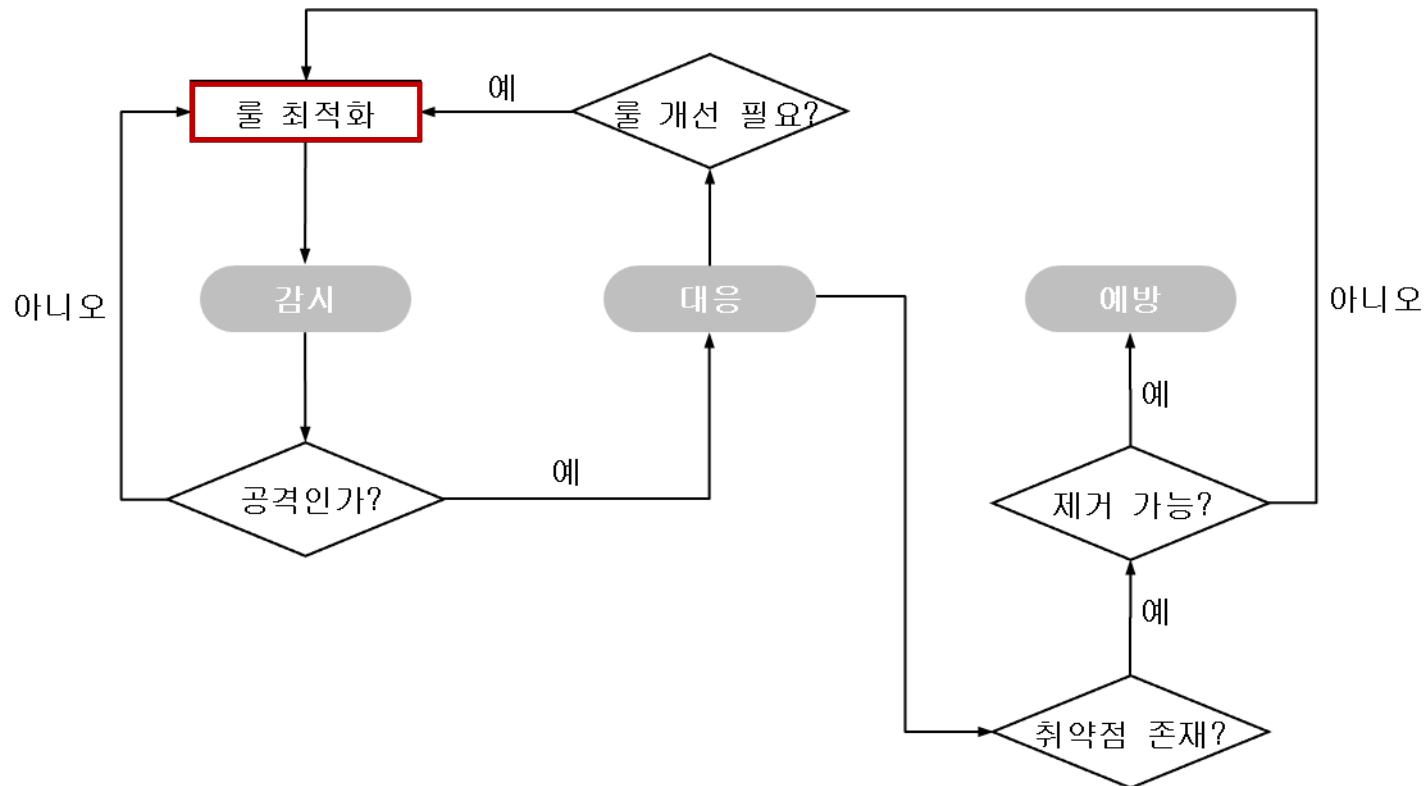
1. Keep the IPS up to date: Regularly update the IPS with the latest signatures and rules to ensure that it can detect the latest threats and reduce false positives.
2. Fine-tune the IPS: The IPS rules should be customized to match the specific network environment and traffic patterns. By tuning the rules, [REDACTED] can reduce the [REDACTED] yes.
3. Use multiple sources for threat intelligence. To reduce the risk of false [REDACTED]

# 네트워크 보안



## ▶ 보안관제 업무 프로세스

- 룰로 시작해서 룰로 끝나는 보안관제

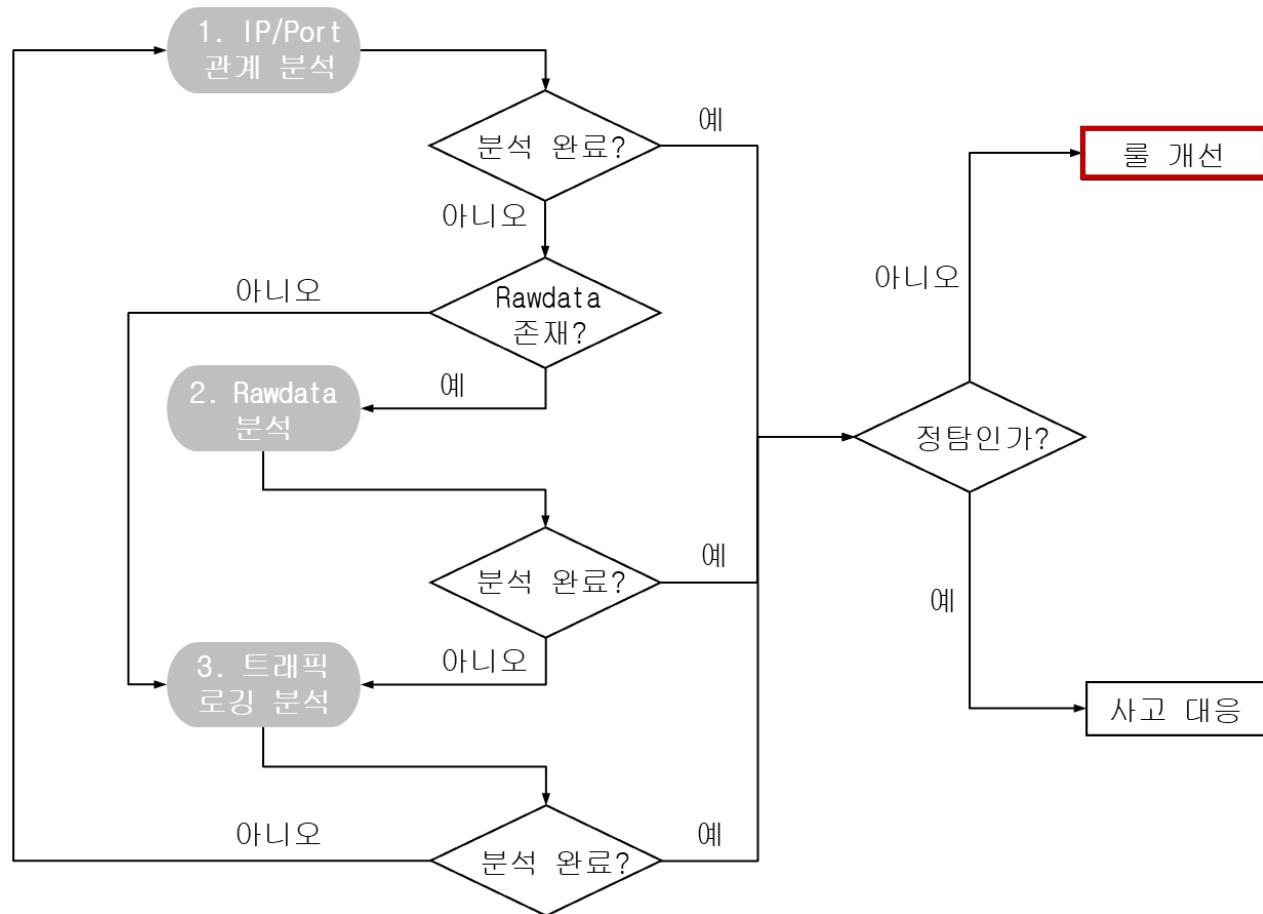


# 네트워크 보안

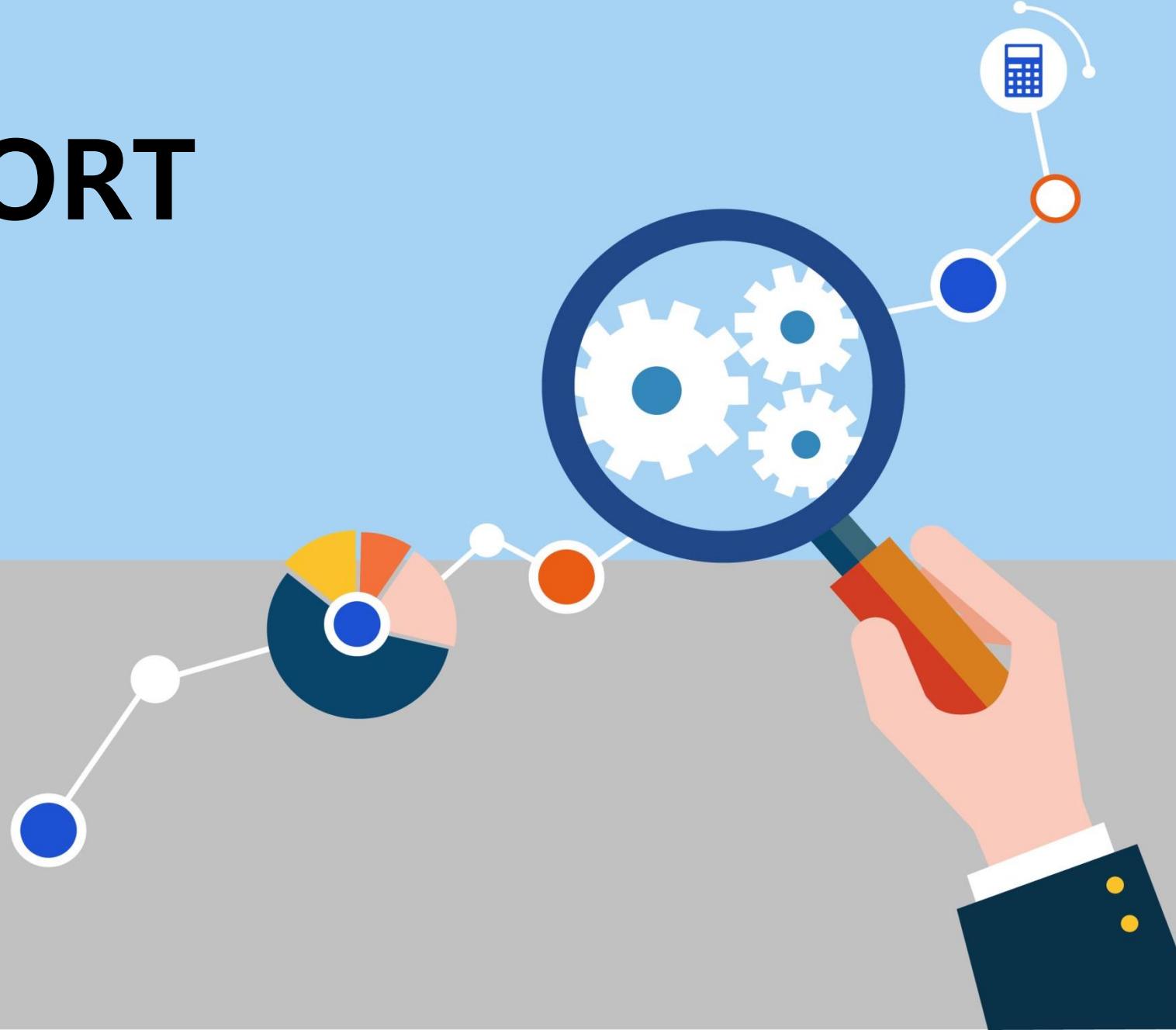


## ▶ 보안관제 업무 프로세스

- 룰로 시작해서 룰로 끝나는 보안장비(IDPS)



# SNORT



## ▶ VirtualBox

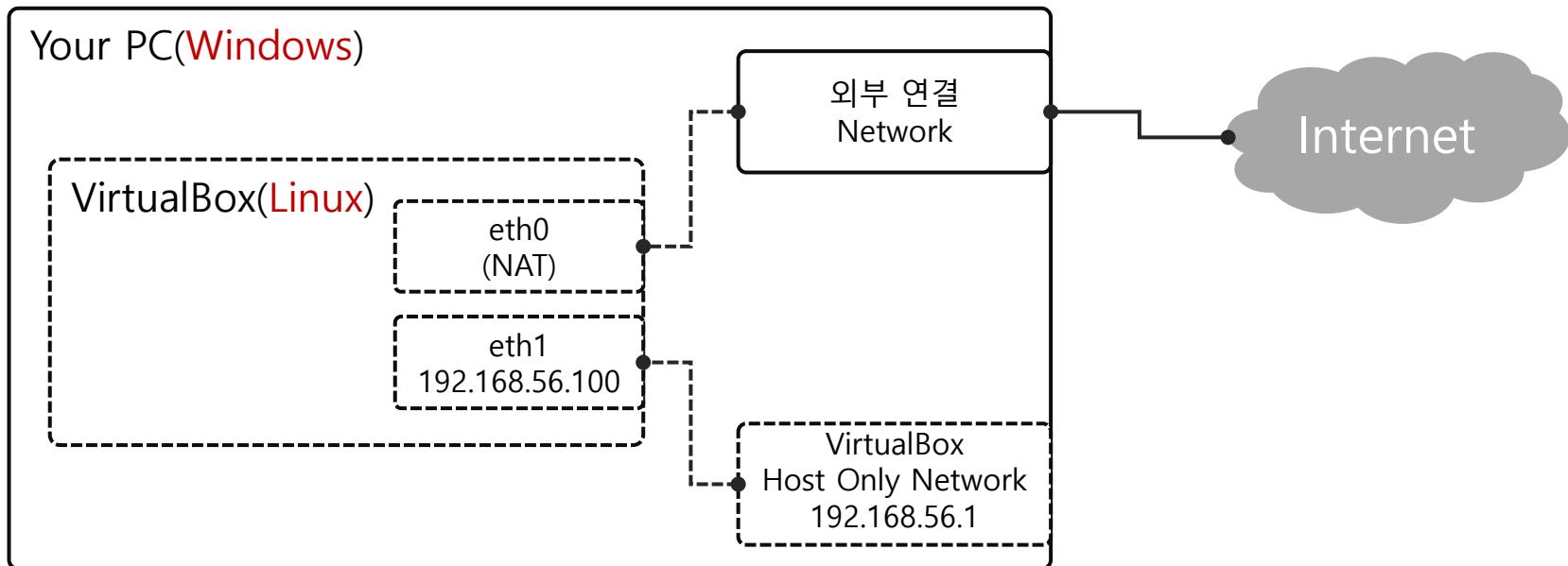
- 가상머신 생성



# SNORT

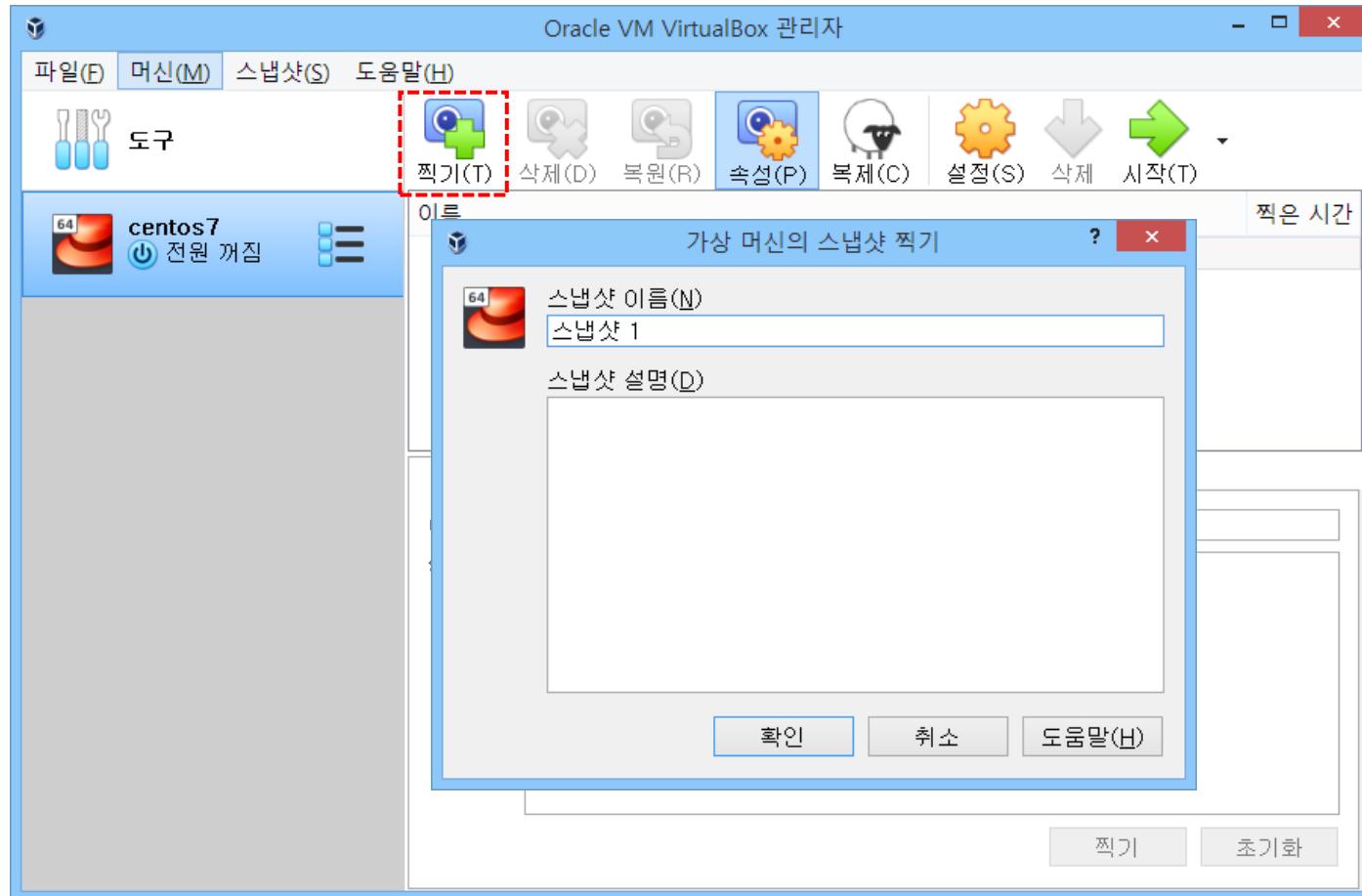
## ▶ VirtualBox

- 가상머신 네트워크



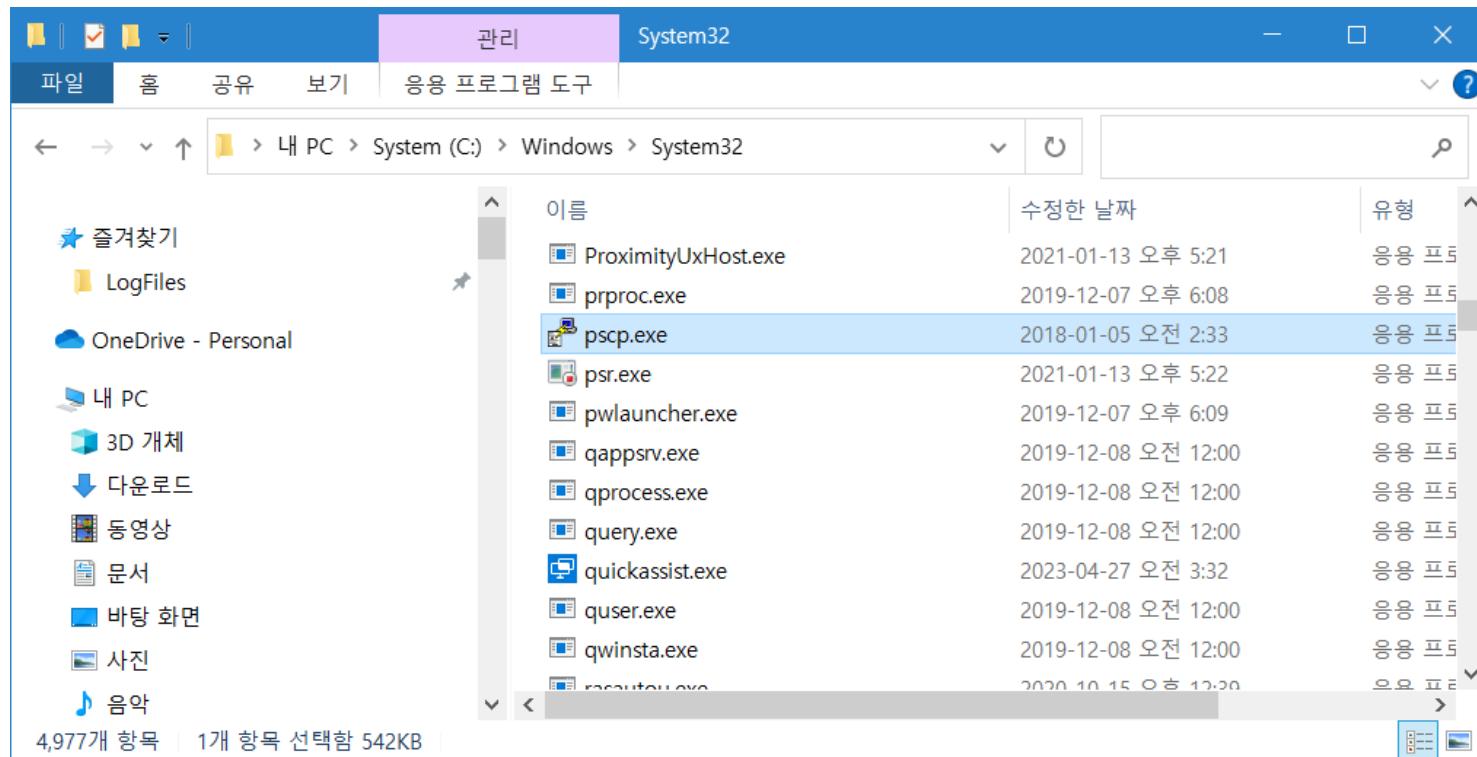
## ▶ VirtualBox

- 스냅샷



## ▶ 리눅스 원격 접속

- pscp.exe(윈도우/리눅스간 파일 전송 툴)

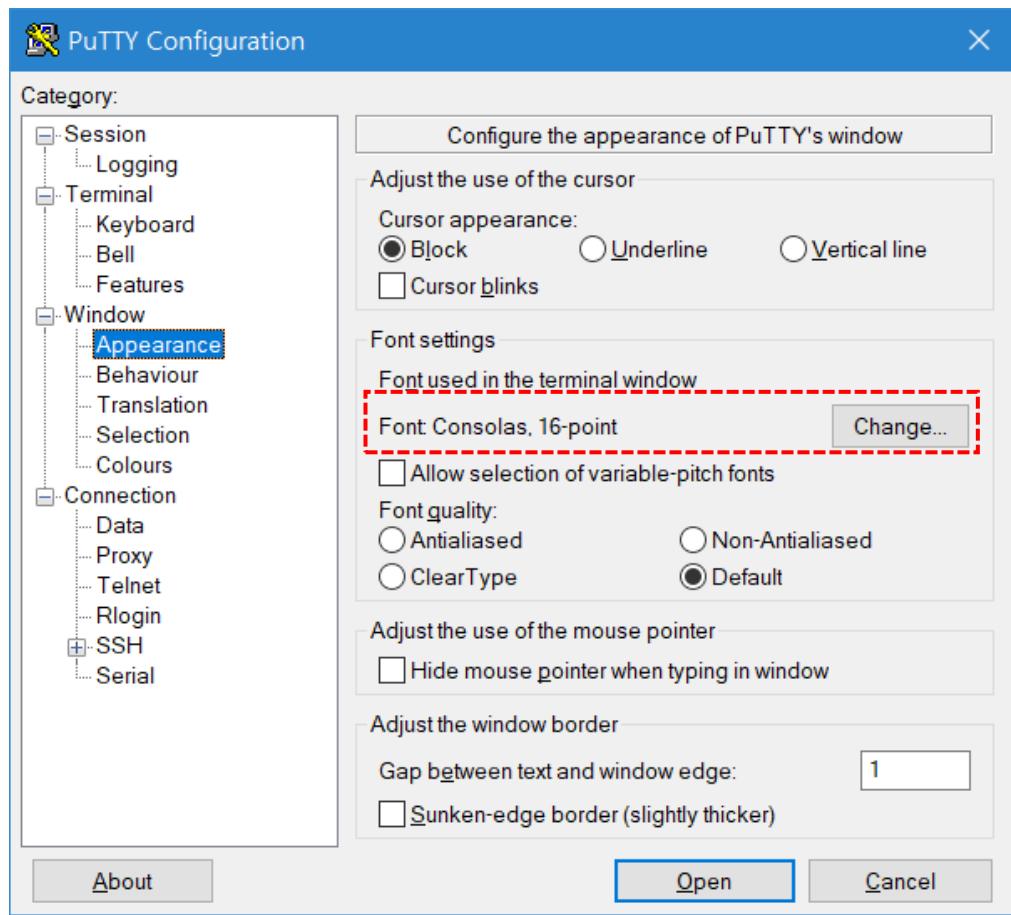


- 명령 구문: pscp FILE ID@IP:PATH



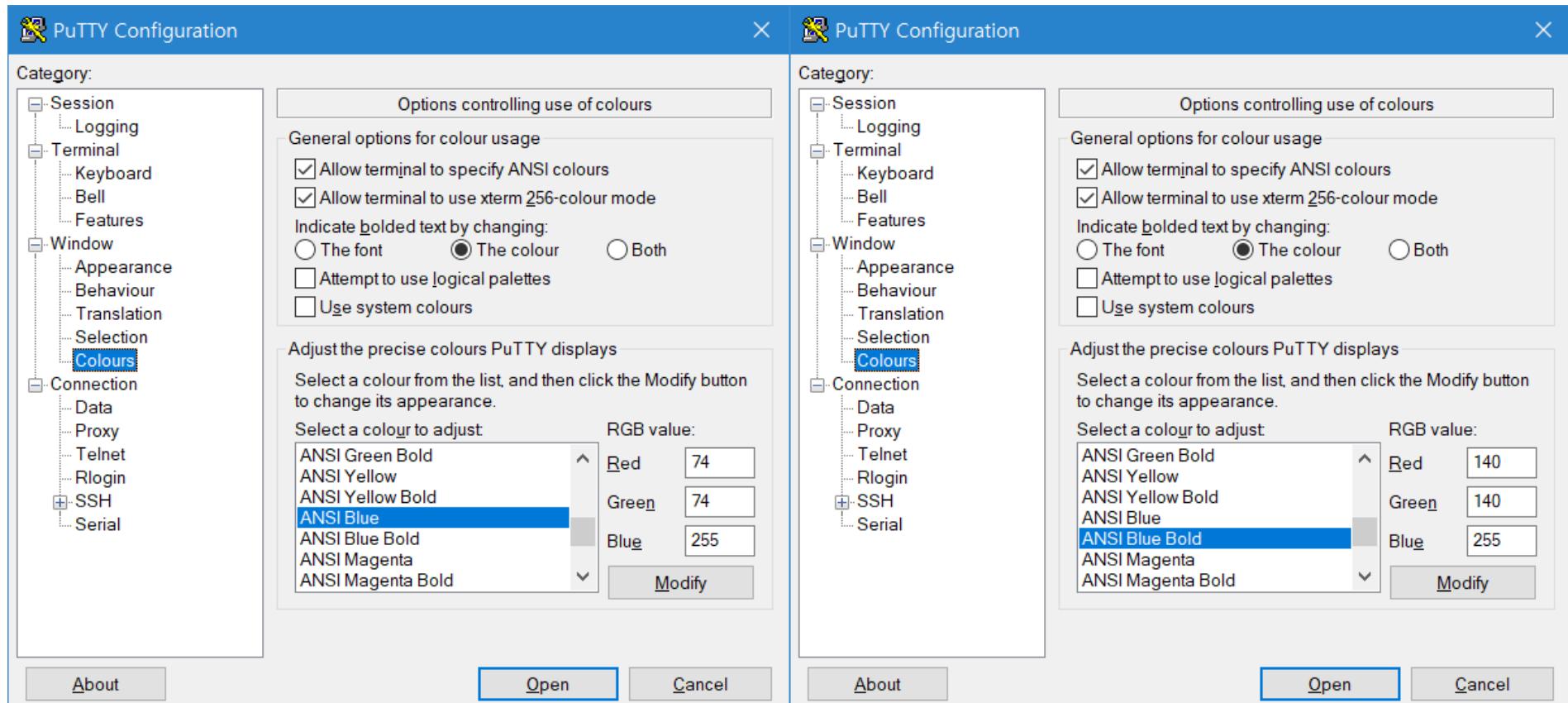
## ▶ 리눅스 원격 접속

- putty.exe 설정



## ▶ 리눅스 원격 접속

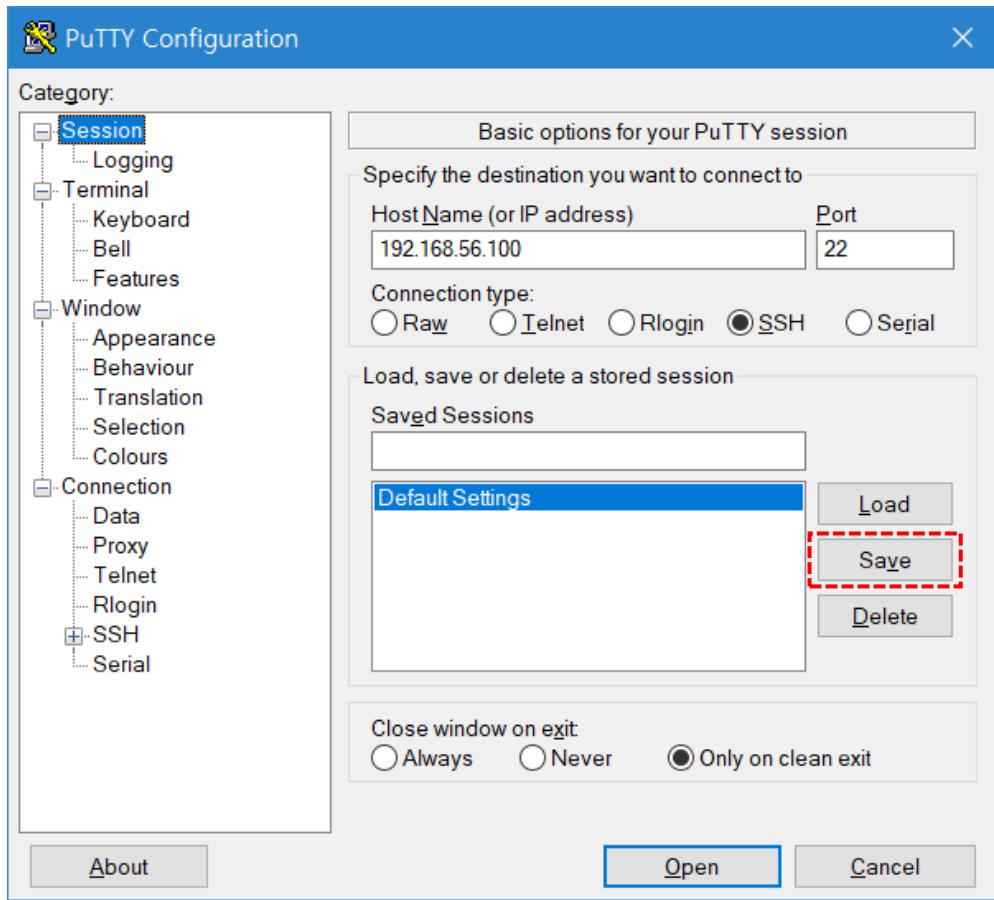
### ▪ putty.exe 설정





## ▶ 리눅스 원격 접속

- putty.exe 설정



# SNORT

## ▶▶ SNORT 설치

- 설치 매뉴얼

- ① yum install epel-release
- ② yum install wget gcc gcc-c++ libnetfilter\_queue-devel git flex bison zlib zlib-devel pcre pcre-devel libdnet\* libpcap\* nghttp2 xz-devel libtool libsfbspf\* daq\* -y
- ③ rpm -ivh [https://snort.org/downloads/snort/snort-2.9.20-1.centos.x86\\_64.rpm](https://snort.org/downloads/snort/snort-2.9.20-1.centos.x86_64.rpm)
- ④ ln -s /usr/lib64/libdnet.so.1.0.1 /usr/lib64/libdnet.1

- 사용자 매뉴얼

# SNORT

## ▶▶ SNORT 설치

- 설치 경로 확인

```
root@Centos7:~]# whereis snort
snort: /usr/sbin/snort /etc/snort /usr/share/man/man8/snort.8.gz
[root@Centos7 ~]#
[root@Centos7 ~]# ls /etc/snort/
classification.config    rules          unicode.map
gen-msg.map                snort.conf
reference.config           threshold.conf
[root@Centos7 ~]#
[root@Centos7 ~]# ls /var/log/snort/
[root@Centos7 ~]# 
```

# SNORT

## ▶▶ SNORT 설정

- 관계도

실행파일

/usr/sbin/snort



/etc/snort/snort.conf

설정파일



/etc/snort/rules/local.rules

룰파일



## ▶▶ SNORT 설정

### ▪ vim 사용 설명서

명령어	설명
i	현재 커서 위치에서 문자 입력
A	현재 커서가 위치한 줄 끝에서 문자 입력
ESC	일반 모드(읽기만 가능)로 전환
x	현재 커서 위치의 문자 삭제
dd	현재 커서가 위치한 줄(Line) 삭제
gg	첫번째 줄로 커서 이동
G	마지막 줄로 커서 이동
u	실행 취소(undo)
CTRL + r	재실행(redo)



## ▶▶ SNORT 설정

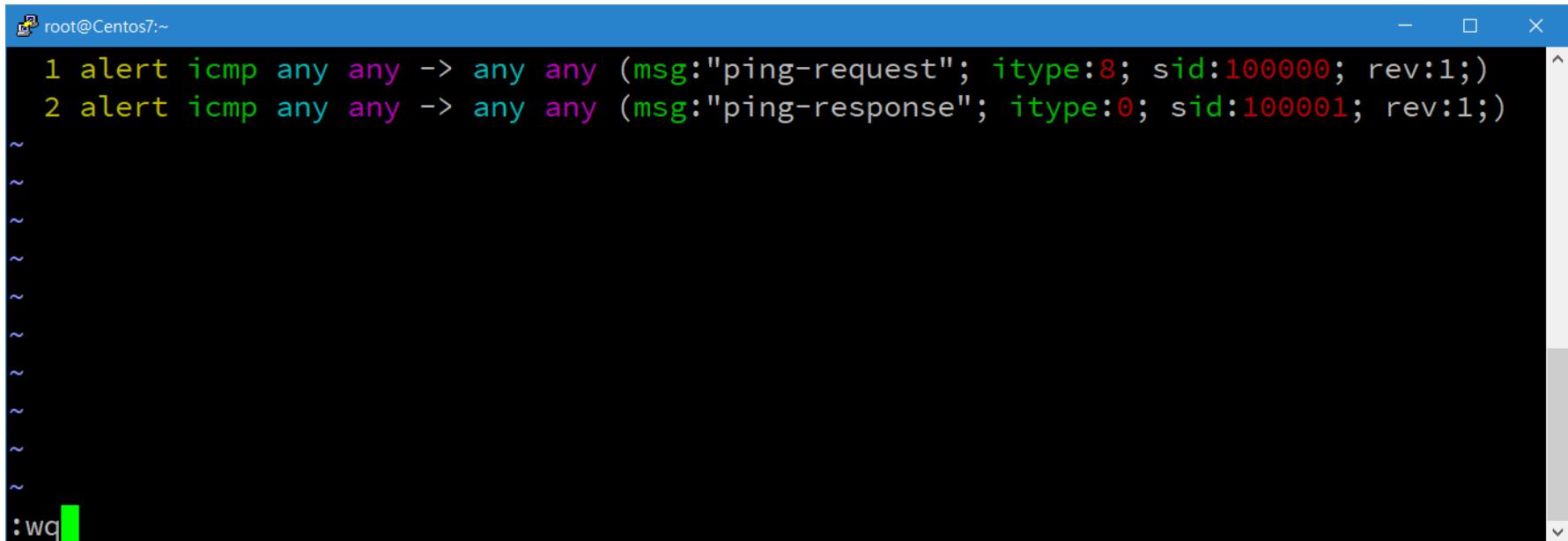
- vim 주요 명령어

검색 모드	명령어 설명
/	검색
n	위에서부터 검색어와 일치하는 문자의 위치로 커서 이동
N	아래에서부터 검색어와 일치하는 문자의 위치로 커서 이동

명령 모드	명령어 설명
:w	저장
:wq	저장 후 종료
:번호	해당 줄 번호로 커서 이동

## ▶▶ SNORT 설정

- vi /etc/snort/rules/local.rules



A screenshot of a terminal window titled "root@Centos7:~". The window contains two lines of Snort alert rules:

```
1 alert icmp any any -> any any (msg:"ping-request"; itype:8; sid:100000; rev:1;)  
2 alert icmp any any -> any any (msg:"ping-response"; itype:0; sid:100001; rev:1;)
```

The terminal window has a dark background with light-colored text. The bottom right corner shows a vertical scroll bar. The bottom left corner shows the command ":wq" followed by a green square cursor.



## ▶▶ SNORT 설정

- vi /etc/snort/snort.conf
- checksum\_mode 변경

```
root@Centos7:~  
139 # config enable_decode_oversized_alerts  
140  
141 # Same as above, but drop packet if in Inline mode (requires enable_decode_o  
versized_alerts)  
142 # config enable_decode_oversized_drops  
143  
144 # Configure IP / TCP checksum mode  
145 config checksum_mode: none  
146  
147 # Configure maximum number of flowbit references. For more information, see  
README.flowbits  
148 # config flowbits_size: 64  
149
```

145,23

20%

## ▶▶ SNORT 설정

- vi /etc/snort/snort.conf
- dynamic rules 설정 주석 처리

```
root@Centos7:~ 248
249 # path to base preprocessor engine
250 dynamicengine /usr/lib64/snort-2.9.9.0_dynamicengine/libsf_engine.so
251
252 # path to dynamic rules libraries
253 #dynamicdetection directory /usr/local/lib/snort_dynamicrules
254
255 ######
256 # Step #5: Configure preprocessors
257 # For more information, see the Snort Manual, Configuring Snort - Prepro-
cessors
258 #####
-- INSERT --                                253,1      36%
```



## ▶▶ SNORT 설정

- vi /etc/snort/snort.conf
- white/blacklist 설정 주석 처리

```
root@Centos7:~ 506 # Reputation preprocessor. For more information see README.reputation
507 processor reputation: \
508     memcap 500, \
509     priority whitelist, \
510     nested_ip inner #, \
511     #whitelist $WHITE_LIST_PATH/white_list.rules, \
512     #blacklist $BLACK_LIST_PATH/black_list.rules
513
514 ##### Step #6: Configure output plugins
515 # Step #6: Configure output plugins
516 # For more information, see Snort Manual, Configuring Snort - Output Modules
-- INSERT --                                     510,20   74%
```



## ▶▶ SNORT 설정

- vi /etc/snort/snort.conf
- 탐지로그 저장 설정

```
root@Centos7:~  
510     nested_ip inner  
511     #whitelist $WHITE_LIST_PATH/white_list.rules, \  
512     #blacklist $BLACK_LIST_PATH/black_list.rules  
513  
514 #####  
515 # Step #6: Configure output plugins  
516 # For more information, see Snort Manual, Configuring Snort - Output Modules  
517 #####  
518  
519 # unified2  
520 # Recommended for most installs  
521 output unified2: filename alert.log, limit 128  
522
```

521,1 75%

## ▶▶ SNORT 설정

- vi /etc/snort/snort.conf
- 룰 설정

```
root@Centos7:~ 542 # NOTE: All categories are enabled in this conf file
543 #####544
545 # site specific rules
546 include $RULE_PATH/local.rules
547
548 #include $RULE_PATH/app-detect.rules
549 #include $RULE_PATH/attack-responses.rules
550 #include $RULE_PATH/backdoor.rules
551 #include $RULE_PATH/bad-traffic.rules
552 #include $RULE_PATH/blacklist.rules
553 #include $RULE_PATH/botnet-cnc.rules
554 #include $RULE_PATH/browser-chrome.rules
:548,651norm i#
```

# SNORT

## ▶▶ SNORT 실행 유형

- 경고/패킷 데이터 분리

- ① snort -i eth0 -c /etc/snort/snort.conf -A fast (IP 헤더 수준 기록)
- ② snort -i eth0 -c /etc/snort/snort.conf -A full (프로토콜 헤더 수준 기록)
- ③ snort -i eth0 -c /etc/snort/snort.conf -K ascii (출발지별 프로토콜 헤더 수준 기록)

- 경고/패킷 데이터 통합

- ① snort -i eth0 -c /etc/snort/snort.conf (unified2 설정)

## ▶▶ mysql 설치

- 설치 및 서비스 시작

- ① rpm -ivh <https://dev.mysql.com/get/mysql80-community-release-el7-11.noarch.rpm>
- ② yum install mysql-server mysql-devel -y
- ③ service mysqld start
- ④ 패스워드 및 외부 접속 설정



## ▶▶ barnyard2 설치

- 설치 매뉴얼

- ① wget https://github.com/firnsy/barnyard2/archive/master.tar.gz -O barnyard2.tar.gz  
tar xvzf barnyard2.tar.gz
- ② cd barnyard2-master
- ③ sed -i 's/my\_bool/bool/' src/output-plugins/spo\_database.h
- ④ ./autogen.sh
- ⑤ ./configure --with-mysql --with-mysql-libraries=/usr/lib64/mysql && make && make install
- ⑥ mkdir /var/log/barnyard2
- ⑦ touch /var/log/snort/by.temp
- ⑧ cp /usr/local/etc/barnyard2.conf /etc/snort/
- ⑨ mysql -u root -p -e "create database snort";
- ⑩ mysql -u root -p -D snort < schemas/create\_mysql



## ▶▶ barnyard2 설정

- vi /etc/snort/barnyard2.conf
- 데이터베이스 저장 설정

```
root@Snort:~  
346 #  
347 # Purpose: This output module provides logging ability to a variety of databases  
348 # See doc/README.database for additional information.  
349 #  
350 # Examples:  
351     output database: log, mysql, user=root password=n01ids dbname=snort host=  
           localhost  
352 #     output database: alert, postgresql, user=snort dbname=snort  
353 #     output database: log, odbc, user=snort dbname=snort  
354 #     output database: log, mssql, dbname=snort user=snort password=test  
355 #     output database: log, oracle, dbname=snort user=snort password=test  
356 #
```

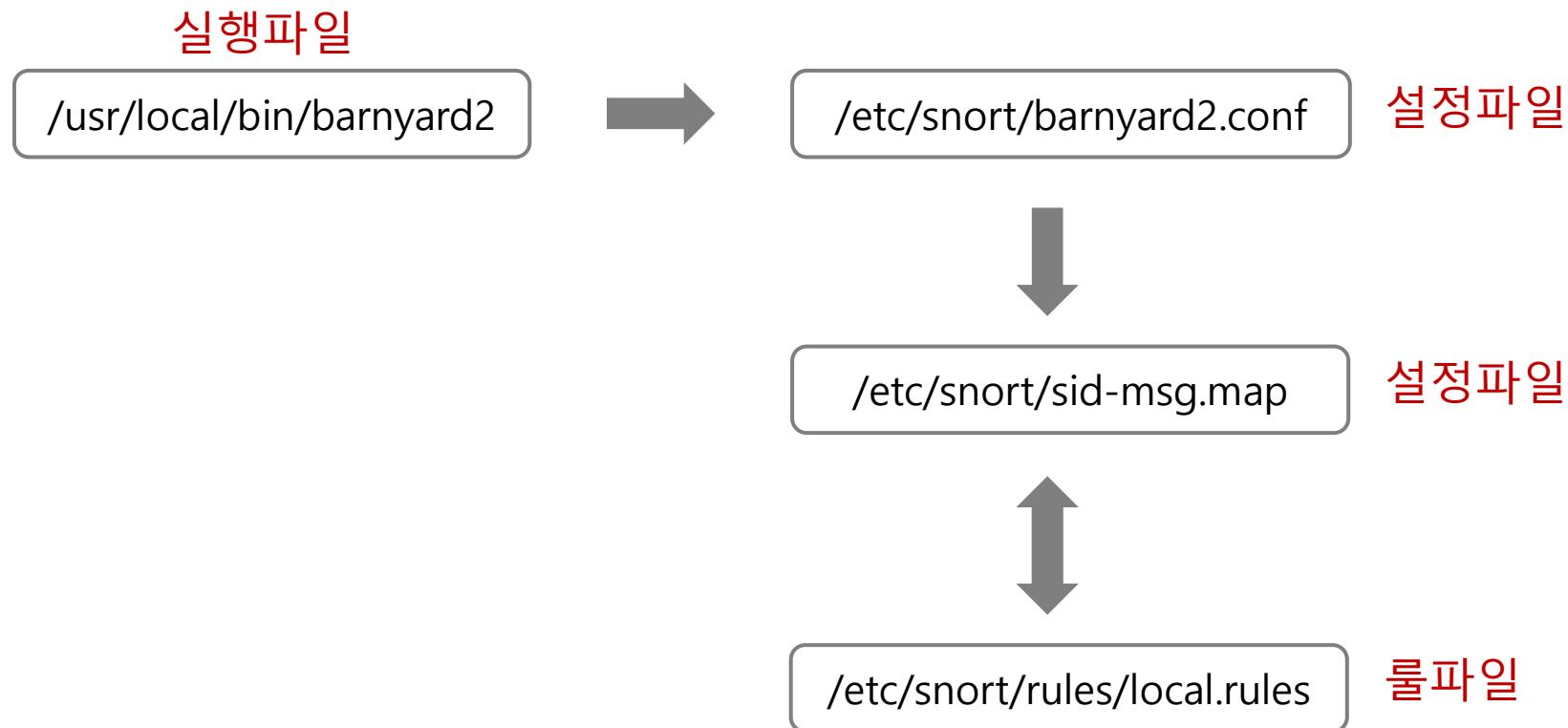
351,26

95%

# SNORT

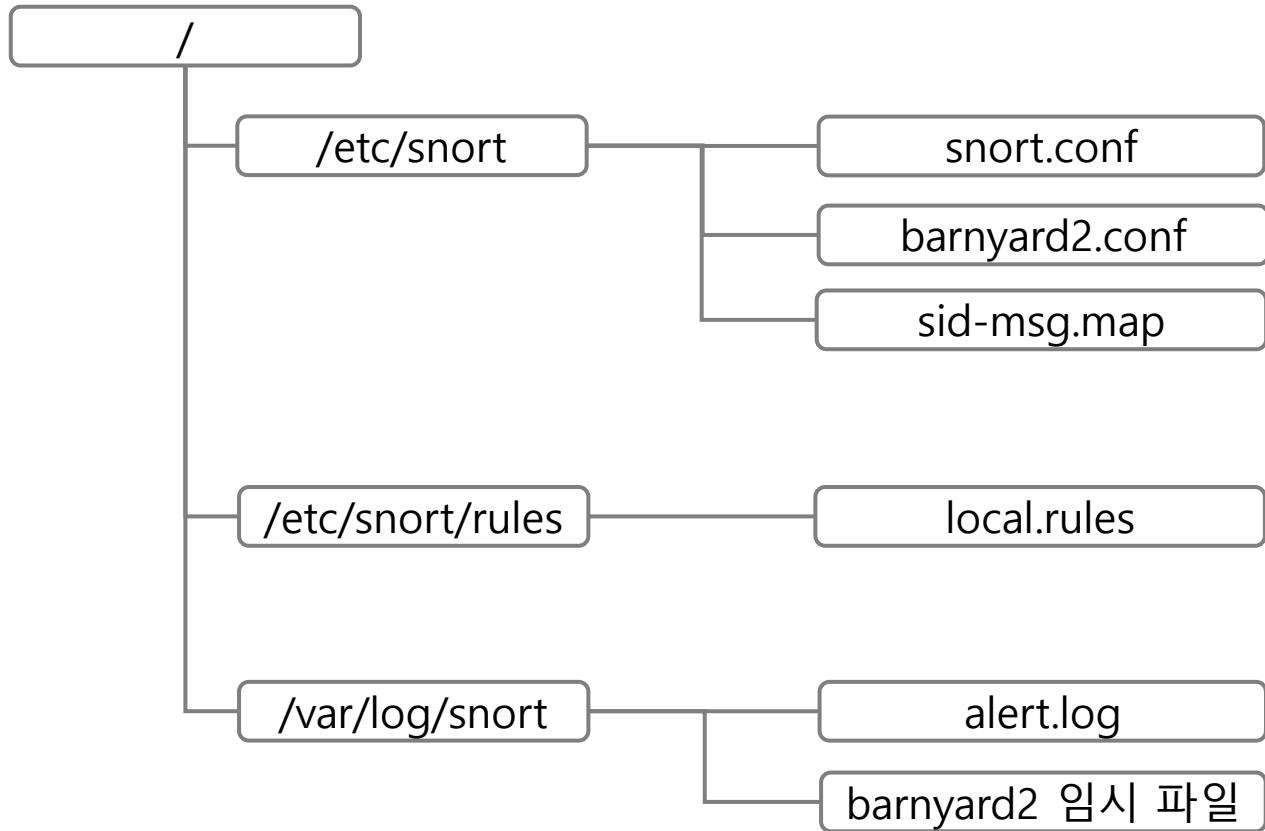
## ▶▶ barnyard2 설정

- 관계도





## ▶ SNORT 경로/주요 설정



## ▶▶ SNORT 스택

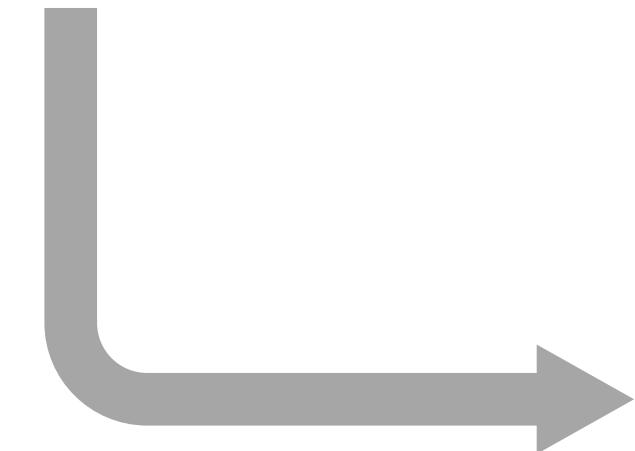
트래픽

전처리

병렬 패턴매칭  
(Aho-Corasick)

룰 패턴매칭

snort



barnyard2

mysql

# SNORT

## ▶▶ SNORT 테스트

- vi /etc/snort/rules/local.rules

```
alert icmp any any -> any any (msg:"ping-request"; itype:8; sid:1000000; rev:1;)
```

```
alert icmp any any -> any any (msg:"ping-response"; itype:0; sid:1000001; rev:1;)
```

- rev 옵션을 추가해야만 barnyard2 연동 가능

- vi /etc/snort/sid-msg.map

```
1000000 || ping-request
```

```
1000001 || ping-response
```

- 룰 파일과 map 파일이 일치할 때만 barnyard2 연동 가능

# SNORT



## ▶▶ barnyard2 실행

- barnyard2 **-c** ‘설정파일’ **-d** ‘snort 로그 경로’ **-f** ‘snort 로그 파일’ **-w** ‘임시 파일’
- barnyard2 **-c** /etc/snort/barnyard2.conf **-d** /var/log/snort/ **-f** alert.log **-w** /var/log/snort/barnyard2.waldo

## ▶▶ SNORT 실행

- snort **-i** ‘랜카드’ **-c** ‘설정파일’
- snort **-i** eth1 **-c** /etc/snort/snort.conf
- snort **-r** sample.pcap **-c** /etc/snort/snort.conf (캡쳐 트래픽 읽기 모드)



## ▶▶ SNORT 테스트 결과 확인

- 데이터베이스 확인
- 발생시간, 발생 탐지명 조회 쿼리문

```
select a.timestamp, b.sig_name  
from event a, signature b  
where a.signature = b.sig_id
```



## ▶▶ SQL(Structured Query Language)

- 데이터 조회 및 가공 등에 사용하는 구조적 질의 언어

## ▶▶ 기본 구조

- select : 어떤 필드를 보겠다
- from : 어떤 테이블에서 보겠다
- where : 어떤 조건으로 보겠다

▪ select 성명  
from student  
where 학번 = 11

학번	성명
10	장길산
11	홍길동
12	임꺽정

student 테이블



## ▶▶ SNORT 스키마

- 주요 테이블/필드 내역

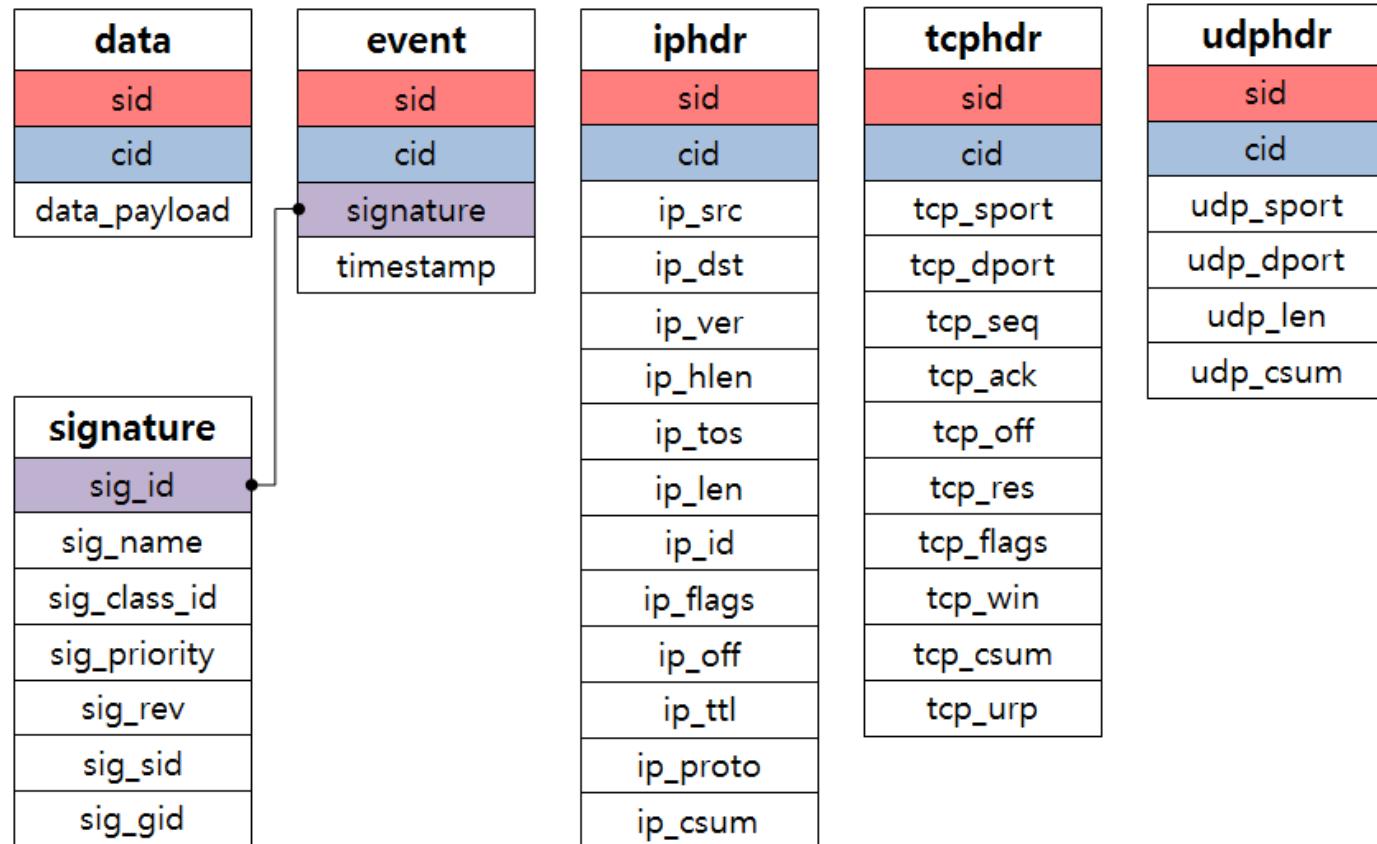
테이블	주요 필드	설명	동기화 필드
event	timestamp	발생 시간	
	signature sig_id	룰 일련번호 룰 이름	signature = sig_id
signature	sig_name		

테이블	주요 필드	설명	동기화 필드
event	timestamp	발생 시간	
	signature	룰 일련번호	
data	data_payload	패킷 페이로드	
iphdr	ip_src	출발지	sid(센서 일련번호), cid(로그 일련번호)
	ip_dst	목적지	
tcphdr	tcp_sport	출발지포트	
	tcp_dport	목적지포트	



## ▶▶ SNORT 스키마

- 주요 테이블/필드 구성



# SNORT

## ▶▶ SNORT 스키마

- event/signature 테이블 관계

SQLyog Community Edition- MySQL GUI - [snort - root@192.168.56.100\*]

File Edit Favorites DB Table Objects Tools Window Help

snort

Query

```
1 select * from signature
```

1 Result 2 Messages 3 Table Data 4 Objects 5 History

sig_id	sig_name	sig_class_id	sig_priority	sig_rev	sig_sid
3284	icmp echo request		0	0	1 1000000

Query batch completed successfully 0 ms 1 row(s) Ln 1, Col 1 Connections : 1 Want more Power? Get Enterprise

Query

```
1 select * from event
```

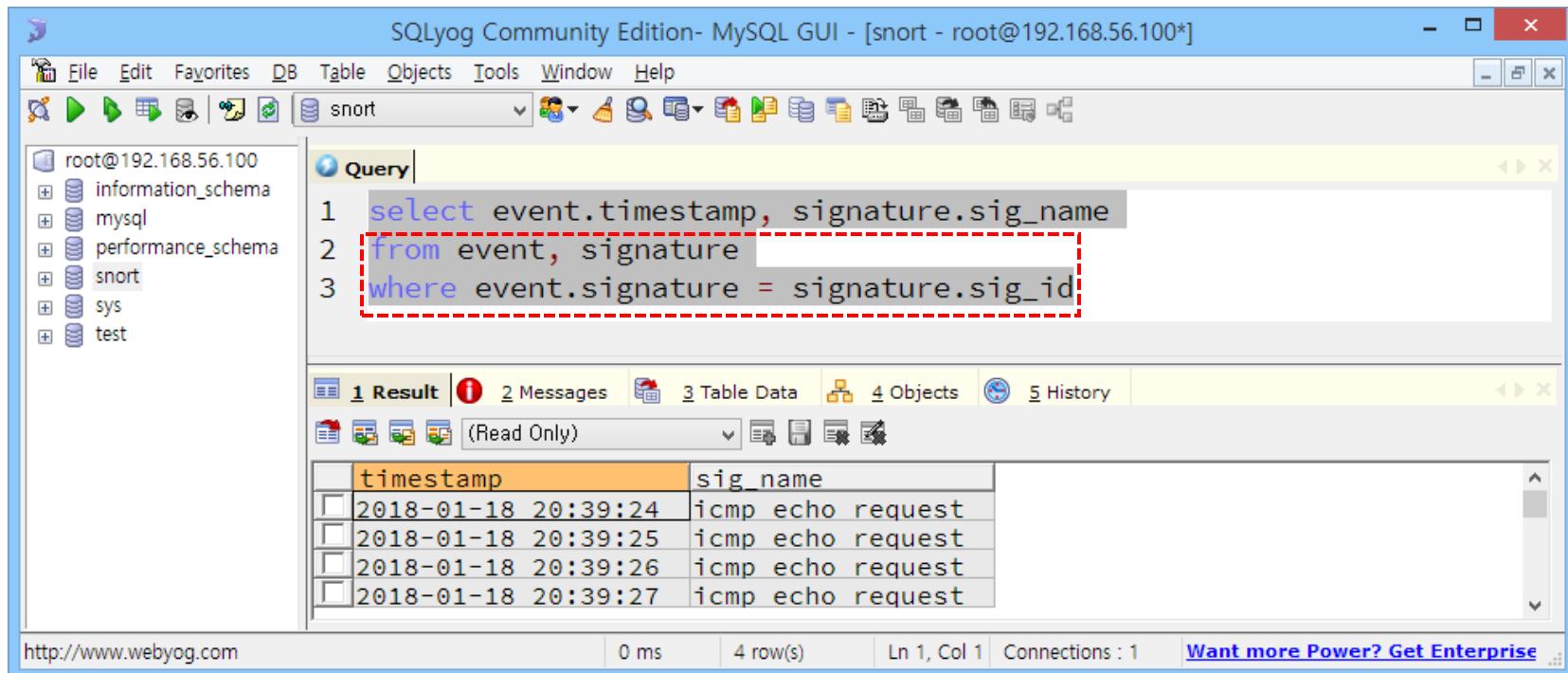
1 Result 2 Messages 3 Table Data 4 Objects 5 History

sid	cid	signature	timestamp
1	6687	3284	2018-01-18 20:39:24
1	6688	3284	2018-01-18 20:39:25
1	6689	3284	2018-01-18 20:39:26
1	6690	3284	2018-01-18 20:39:27

Query batch completed successfully 0 ms 4 row(s) Ln 1, Col 1 Connections : 1 Want more Power? Get Enterprise

## ▶▶ SNORT 스키마

- ‘event.signature = signature.sig\_id’ 조건으로 동기화



SQLyog Community Edition- MySQL GUI - [snort - root@192.168.56.100\*]

Query

```
1 select event.timestamp, signature.sig_name
2 from event, signature
3 where event.signature = signature.sig_id
```

Result

timestamp	sig_name
2018-01-18 20:39:24	icmp echo request
2018-01-18 20:39:25	icmp echo request
2018-01-18 20:39:26	icmp echo request
2018-01-18 20:39:27	icmp echo request

http://www.webyog.com | 0 ms | 4 row(s) | Ln 1, Col 1 | Connections : 1 | Want more Power? Get Enterprise



## ▶▶ SNORT 스키마

- 테이블명 별칭 부여를 통해 쿼리문 간소화 가능
- ‘a.signature = b.sig\_id’ 조건으로 동기화

The screenshot shows the SQLyog Community Edition interface. The title bar reads "SQLyog Community Edition- MySQL GUI - [snort - root@192.168.56.100\*]". The left sidebar lists databases: root@192.168.56.100, information\_schema, mysql, performance\_schema, snort, sys, and test. The main area has a "Query" tab open with the following SQL code:

```
1 select a.timestamp, b.sig_name
2 from event a, signature b
3 where a.signature = b.sig_id
```

The code is highlighted with a red dashed rectangle around the "from" and "where" clauses. Below the query is a "Result" tab showing the output:

timestamp	sig_name
2018-01-18 20:39:24	icmp echo request
2018-01-18 20:39:25	icmp echo request
2018-01-18 20:39:26	icmp echo request
2018-01-18 20:39:27	icmp echo request

At the bottom, the status bar says "Query batch completed successfully" and "Want more Power? Get Enterprise".



## ▶▶ SNORT 스키마

- 테이블 3개를 동기화해주는 쿼리문

The screenshot shows the SQLyog Community Edition interface. The left sidebar lists databases: root@192.168.56.100, information\_schema, mysql, performance\_schema, snort, sys, and test. The main window has a 'Query' tab open with the following SQL code:

```
1 select a.timestamp, b.sig_name, c.data_payload
2 from event a, signature b, data c
3 where a.signature = b.sig_id
4 and a.sid = c.sid and a.cid = c.cid
```

The last three lines of the query (the join conditions) are highlighted with a red dashed box. Below the query is a 'Result' tab showing a table with four rows of data:

timestamp	sig_name	data_payload
2018-01-18 20:39:24	icmp echo request	6162636465666768696A6B6C6...
2018-01-18 20:39:25	icmp echo request	6162636465666768696A6B6C6...
2018-01-18 20:39:26	icmp echo request	6162636465666768696A6B6C6...
2018-01-18 20:39:27	icmp echo request	6162636465666768696A6B6C6...

At the bottom, the status bar indicates: 'Query batch completed successfully'.

## ▶▶ SNORT 스키마

- 16진수를 디코딩해주는 ‘unhex 함수’

The screenshot shows the SQLyog Community Edition interface. The left sidebar lists databases: root@192.168.56.100, information\_schema, mysql, performance\_schema, snort, sys, and test. The current database is 'snort'. The main window has a 'Query' tab with the following SQL code:

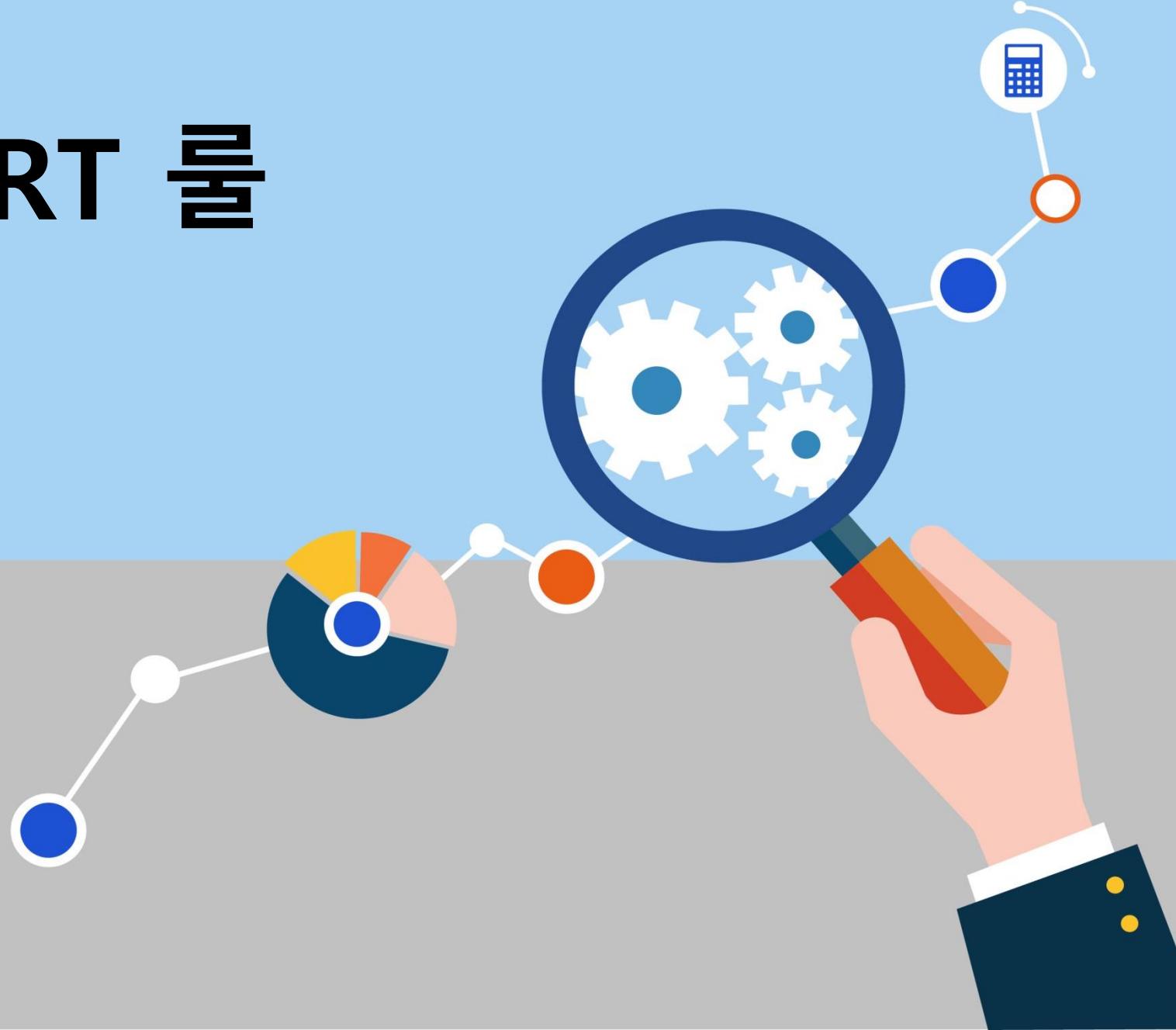
```
1 select a.timestamp, b.sig_name, unhex(c.data_payload)
2 from event a, signature b, data c
3 where a.signature = b.sig_id
4 and a.sid = c.sid and a.cid = c.cid
```

The 'Result' tab displays the query results in a table:

timestamp	sig_name	unhex(c.data_payload)
2018-01-18 20:39:24	icmp echo request	abcdefghijklmnoprstuvwxyzabcdefghi
2018-01-18 20:39:25	icmp echo request	abcdefghijklmnoprstuvwxyzabcdefghi
2018-01-18 20:39:26	icmp echo request	abcdefghijklmnoprstuvwxyzabcdefghi
2018-01-18 20:39:27	icmp echo request	abcdefghijklmnoprstuvwxyzabcdefghi

At the bottom, the status bar shows: Query batch completed successfully | 0 ms | 4 row(s) | Ln 1, Col 1 | Connections : 1 | Want more Power? Get Enterprise

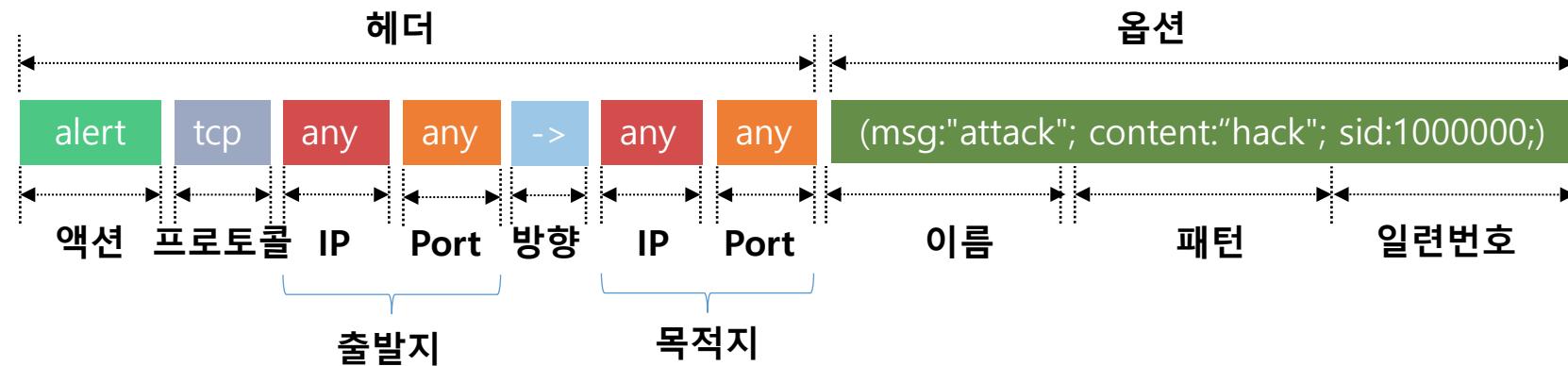
# SNORT



# SNORT 룰

## ▶ 룰

- 헤더: 트래픽 발생 주체 및 방향 정의
- 옵션: 트래픽 세부 특징 정의



# SNORT 룰

## ▶ 룰 옵션

- 패킷 헤더(물리적 특성) 및 페이로드(내용) 정의

```
Ethernet II, Src: 5c:ac:4c:65:3c:44 (5c:ac:4c:65:3c:44)
Internet Protocol Version 4, Src: 192.168.0.3 (192.168.
Transmission Control Protocol, Src Port: 27780 (27780),
Hypertext Transfer Protocol
+ GET /ad/adview?secid=0AT02 HTTP/1.1\r\n
  Host: amsv2.daum.net\r\n
  Connection: keep-alive\r\n
  User-Agent: mozilla/5.0\r\n
  Accept: text/html,application/xhtml+xml,application/>
  Referer: http://www.daum.net/?t__nil_logo=daum\r\n
```

header

payload

# SNORT 를



## ▶ 패킷 헤더(Non-Payload) 검사 를 옵션

- <http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node33.html>
- 패킷 헤더의 물리적 특성을 이용한 검사

옵션	설명
flow	트래픽 방향 선택(SYN 플래그 기준)
flowbit	세션 추적
flags	TCP flag bit 검사
dsize	페이지 사이즈 검사

## ▶ 권고사항

- Payload 검사 옵션보다 선행할 때 성능 향상에 유리
- alert tcp any any -> any any ( ① 패킷 헤더(Non-Payload) 검사    ② Payload 검사)

# SNORT 를



## ▶ 패킷 헤더(Non-Payload) 검사 를 옵션

- flow(트래픽 방향 구분)
- 방향 구분 기준 : TCP SYN 플래그

옵션	설명	비고
to_server	to_SYN 플래그 수신자	from_client와 동일
to_client	to_SYN 플래그 송신자	from_server와 동일
established	세션 수립 후	
not_established	세션 수립 전	

## ▶ 예시

- `flow:to_server; content:"aaa";`
- TCP SYN 플래그를 수신한 서버로 향하는 트래픽만 검사

# SNORT 를



## ▶ 패킷 헤더(Non-Payload) 검사 를 옵션

- flow
- 누가 단독 syn 요청을 하는가?

Time	Protocol	sIP	dIP	sPort	dPort	Info
2011-04...	TCP	10.168.0.3	10.239.204.205	62553	80	62553 → 80 [SYN] Seq=0...
2011-04...	TCP	10.239.204.205	10.168.0.3	80	62553	80 → 62553 [SYN, ACK] ...
2011-04...	TCP	10.168.0.3	10.239.204.205	62553	80	62553 → 80 [ACK] Seq=1...
2011-04...	HTTP	10.168.0.3	10.239.204.205	62553	80	POST http://update.vir...
2011-04...	TCP	10.239.204.205	10.168.0.3	80	62553	80 → 62553 [ACK] Seq=1...
2011-04...	HTTP	10.239.204.205	10.168.0.3	80	62553	HTTP/1.1 200 OK (text...
2011-04...	TCP	10.168.0.3	10.239.204.205	62553	80	62553 → 80 [ACK] Seq=2...
2011-04...	TCP	10.239.204.205	10.168.0.3	80	62553	80 → 62553 [FIN, ACK] ...
2011-04...	TCP	10.168.0.3	10.239.204.205	62553	80	62553 → 80 [ACK] Seq=2...
2011-04...	TCP	10.168.0.3	10.239.204.205	62553	80	62553 → 80 [FIN, ACK] ...
2011-04...	TCP	10.239.204.205	10.168.0.3	80	62553	80 → 62553 [ACK] Seq=3...

# SNORT 를



## ▶▶ 패킷 헤더(Non-Payload) 검사 를 옵션

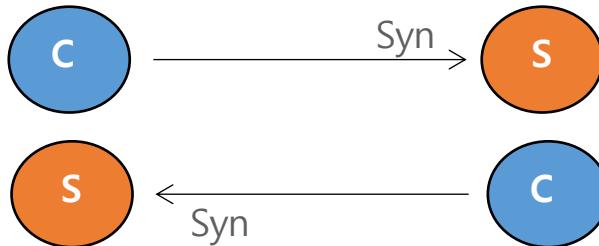
- flow
- 단점 : Client <-> Server 구분이 명확한 네트워크에서만 사용 가능

## ▶▶ 예시

- alert tcp any any -> any any (flow:to\_server;)
- 단순 네트워크



- 복잡 네트워크



# SNORT 를



## ▶ 패킷 헤더(Non-Payload) 검사 를 옵션

- flowbit(세션 추적)
- content:"securecrt.html"; flowbits:set, ‘세션 이름’ ; flowbits:noalert;

No.	Time	Protocol	sIP	dIP	sPort	dPort	Info
→	2011-04-10 ...	HTTP	10.168.0.3	10.184.27.17	62669	80	GET /pricing/corporate/
←	2011-04-10 ...	HTTP	10.184.27.17	10.168.0.3	80	62669	HTTP/1.1 200 OK (text/)

↳ Hypertext Transfer Protocol

  ↳ GET /pricing/corporate/applications/securecrt.html HTTP/1.1\r\n\r\n

    Accept: text/html, application/xhtml+xml, \*/\*\r\n\r\n

    Accept-Language: ko-KR\r\n\r\n

- content:"VanDyke"; flowbits:isset, ‘세션 이름’ ;

No.	Time	Protocol	sIP	dIP	sPort	dPort	Info
→	2011-04-10 ...	HTTP	10.168.0.3	10.184.27.17	62669	80	GET /pricing/corporate/
←	2011-04-10 ...	HTTP	10.184.27.17	10.168.0.3	80	62669	HTTP/1.1 200 OK (text/)

<html>\n<head>\n<title>VanDyke Software - Redirection</title>\n

# SNORT 룰



## ▶ 패킷 헤더(Non-Payload) 검사 룰 옵션

- flowbit 테스트
- alert tcp any any -> any any (msg:"flowbits-set"; content:"securecrt.html"; flowbits:set,test; flowbits:noalert; sid:1000001; rev:1;)
- alert tcp any any -> any any (msg:"flowbits-test"; flowbits:isset,test; content:"VanDyke"; sid:1000002; rev:1;)

# SNORT 를



## ▶▶ 패킷 헤더(Non-Payload) 검사 를 옵션

- flags(TCP 흐름 제어 검사)
- 옵션

Flags	C(WR)	E(CE)	U(rgent)	A(ck)	P(ush)	R(st)	S(yn)	F(in)
2진수	10000000	1000000	100000	10000	1000	100	10	1
10진수	128	64	32	16	8	4	2	1

## ▶▶ 예시

- 좋은 예시 : flags:S; content:"aaa";
- 나쁜 예시 : content:"aaa"; flags:S;
- 더 나쁜 예시 : content:"aaa"; flags:PA;

# SNORT 를



## ▶ 패킷 헤더(Non-Payload) 검사 를 옵션

- flags
- Ack, Push 플래그 설정 패킷

tcp.flags==24

No.	Time	Protocol	sIP	dIP	sPort	dPort	Info
4	2011-0...	HTTP	10...	10...	62553	80	POST http://update.virtualbox.org/query.php?platfo
6	2011-0...	HTTP	10...	10...	80	62553	HTTP/1.1 200 OK (text/html)
11	2011-0...	TCP	10...	10...	62555	80	62555 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17160 Len=52
12	2011-0...	HTTP	10...	10...	62555	80	POST /service/update?33w-3...com OnVRinHa_hhVclUTvD

Flags: 0x018 (PSH, ACK)

- 000. .... .... = Reserved: Not set
- ...0 .... .... = Nonce: Not set
- .... 0.... .... = Congestion Window Reduced (CWR): Not set
- .... .0.... .... = ECN-Echo: Not set
- .... .0..... = Urgent: Not set
- .... ...1 .... = Acknowledgment: Set
- .... .... 1... = Push: Set
- .... .... .0.. = Reset: Not set
- .... .... ..0. = Syn: Not set
- .... .... ...0 = Fin: Not set

[TCP Flags: .....AP....]

Window size value: 12288

[Calculated window size: 49152]

Flags (12 bits) (tcp.flags), 2 bytes

Packets: 5030 · Displayed: 854 (17.0%) · Load time: 0:0,220 · Profile: Default

# SNORT 를



## ▶ 패킷 헤더(Non-Payload) 검사 를 옵션

- dsize(payload 사이즈 검사)
- 옵션

옵션	설명
dsize:300	300byte
dsize:>300	300byte 초과
dsize:<300	300byte 미만
dsize:199->301	200 ~ 300byte

## ▶ 예시

- alert tcp any any -> any any (msg:"flags-test"; dsize:<100; content:"HTTP/"; sid:1000004; rev:1;)

# SNORT 를



## ▶ 패킷 헤더(Non-Payload) 검사 를 옵션

- dsize
- $328 - 54(\text{패킷 헤더 사이즈}) = 274$

The screenshot shows the Wireshark interface with the file "web\_test.pcap" open. The packet list pane shows several HTTP and TCP packets. A red dashed box highlights the details of a selected TCP packet (Frame 4). The expanded details pane shows the following information:

- Frame 4: 328 bytes on wire (2624 bits), 328 bytes captured (2624 bits)
- Ethernet II, Src: Sc:ac:4c:65:3c:44, Dst: 00:26:66:13:c7:69
- Internet Protocol Version 4, Src: 10.168.0.3, Dst: 10.239.204.205
- Transmission Control Protocol, Src Port: 62553, Dst Port: 80, Seq: 1, Ack: 1, Len: 274
- Source Port: 62553
- Destination Port: 80
- [Stream index: 0]
- [TCP Segment Len: 274]
- Sequence number: 1 (relative sequence number)
- [Next sequence number: 275 (relative sequence number)]
- Urgent pointer: 0
- [SEQ/ACK analysis]
- TCP payload (274 bytes)

A callout box labeled "전체 패킷 사이즈" points to the total frame size (328 bytes). Another callout box labeled "페이지 사이즈" points to the TCP payload size (274 bytes).

# SNORT 룰



## ▶ 룰 옵션

- threshold(패킷 발생량 측정)
- alert icmp any any -> any any (msg:"threshold-test"; itype:8; threshold:type threshold, track by\_dst, count 2, seconds 3; sid:1000005; rev:1;)
- ‘track by\_dst, count 2, seconds 3;’ 조건에서 10초간 패킷 10개 발생 시

분류	옵션	로그 발생 단위	로그 발생
유형	threshold:type <b>threshold</b>	패킷 임계치(임계 시간 무시)	5개
	threshold:type <b>limit</b>	임계 시간 동안의 패킷 임계치(=발생 개수)	7개
	threshold:type <b>both</b>	임계 시간	3개
IP 조건	track by_src	출발지 IP	
	track by_dst	목적지 IP	
패킷 발생량	count		
시간 조건	seconds		

# SNORT 룰

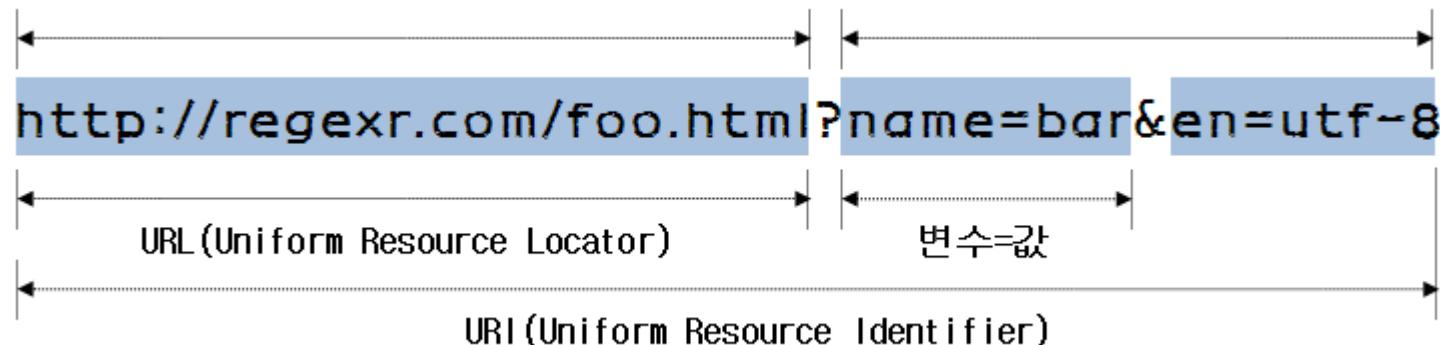


## ▶ 룰 옵션

- Payload 검사 옵션
- <http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node32.html>

옵션	검사 범위	비고
content	페이지 전체	
uricontent	URI	순수 문자열 검사만 가능
pcre	페이지 전체	정규표현식 지원

- URI 구조



# SNORT 룰



## ▶▶ Payload 검사 룰 옵션

- 수정자(Modifier)
- content/uricontent 검사 방식, 범위 등을 수정

옵션	설명	비고
<b>nocase</b>	대소문자 구분 해제	
<b>offset</b>	검사 시작 위치	절대 위치
<b>depth</b>	검사 범위	
<b>distance</b>	검사 시작 위치	상대 위치
<b>within</b>	검사 범위	
<b>fast_pattern</b>	Aho-Corasick 패턴매칭 우선순위 조정	

# SNORT 를



## ▶▶ Payload 검사 를 옵션

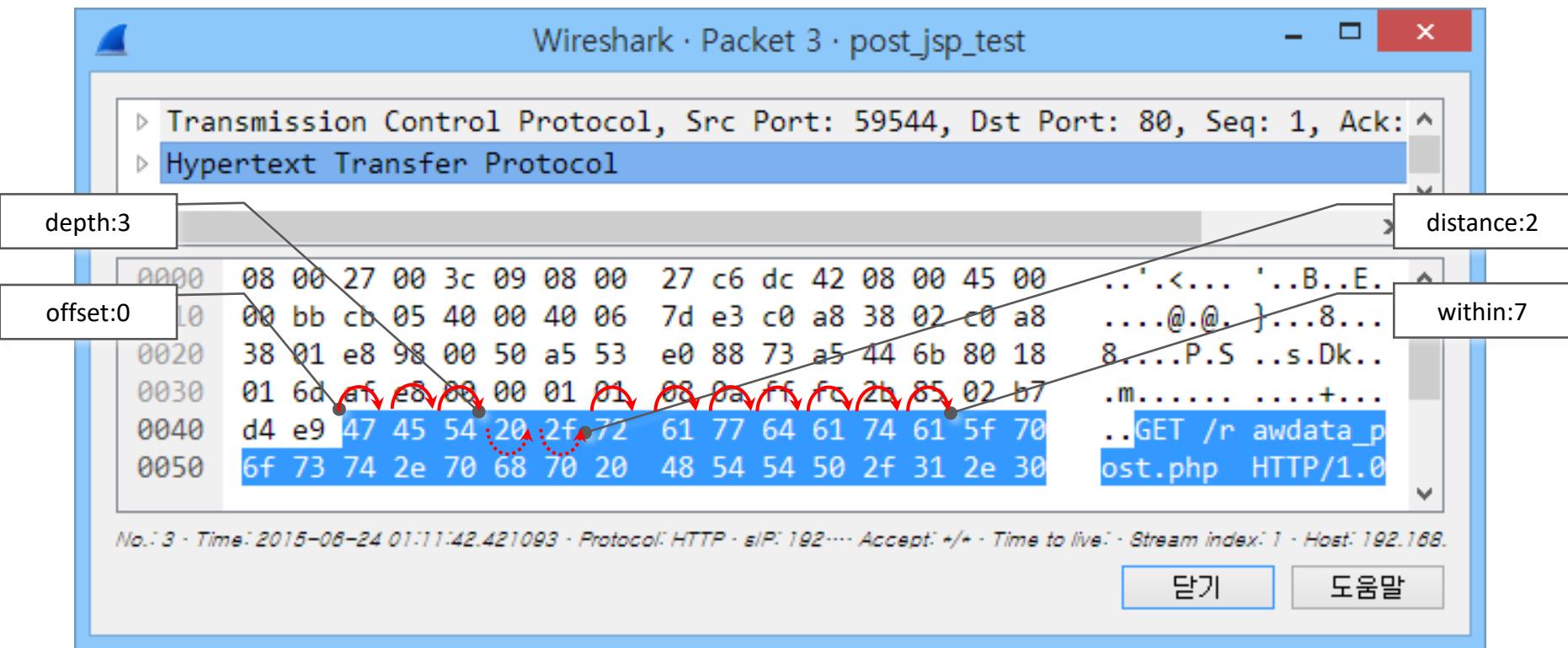
- 추가 수정자

옵션	검사 범위	비고
http_uri	웹 요청 URI	uricontent와 동일
http_header	웹 요청 헤더	
http_method	웹 요청 method	
http_cookie	웹 요청 cookie	
http_client_body	POST method로 전송되는 데이터	
http_stat_code	웹 응답 코드번호	
http_stat_msg	웹 응답 메시지	

# SNORT 룰

## ▶▶ Payload 검사 룰 옵션

- 검사 위치 및 범위 수정자
- content:"GET"; offset:0; depth:3; content:"rawdata"; distance:2; within:7;



# SNORT 룰



## ▶▶ Payload 검사 룰 옵션

- pcre(perl compatible regular expression)
- 정규표현식 사용 가능

## ▶▶ content 와 차이점

- 한 글자 알파벳 이름을 갖는 exe 파일 검사 시 26개의 룰 생성 필요
- content:"a.exe"; ... content:"z.exe";
- pcre옵션을 이용하면 1개의 룰만 생성
- pcre:"/[a-z]\.exe/";

# SNORT 룰



## ▶▶ Payload 검사 룰 옵션

- pcre 수정자

Option	Description	Example
/i	대소문자 구분 안 함	pcre:"/GET.*exe/i";
/s	줄바꿈 문자 검사	pcre:"/GET.*exe/s";
/m	앵커 문자의 줄 구분 해제	pcre:"/^GET.*exe\$/m";
/R (Snort only)	'distance:0' 과 동일	content:"GET /"; pcre:"/\.exe\$/R";

# SNORT 를



## ▶▶ Payload 검사 를 옵션

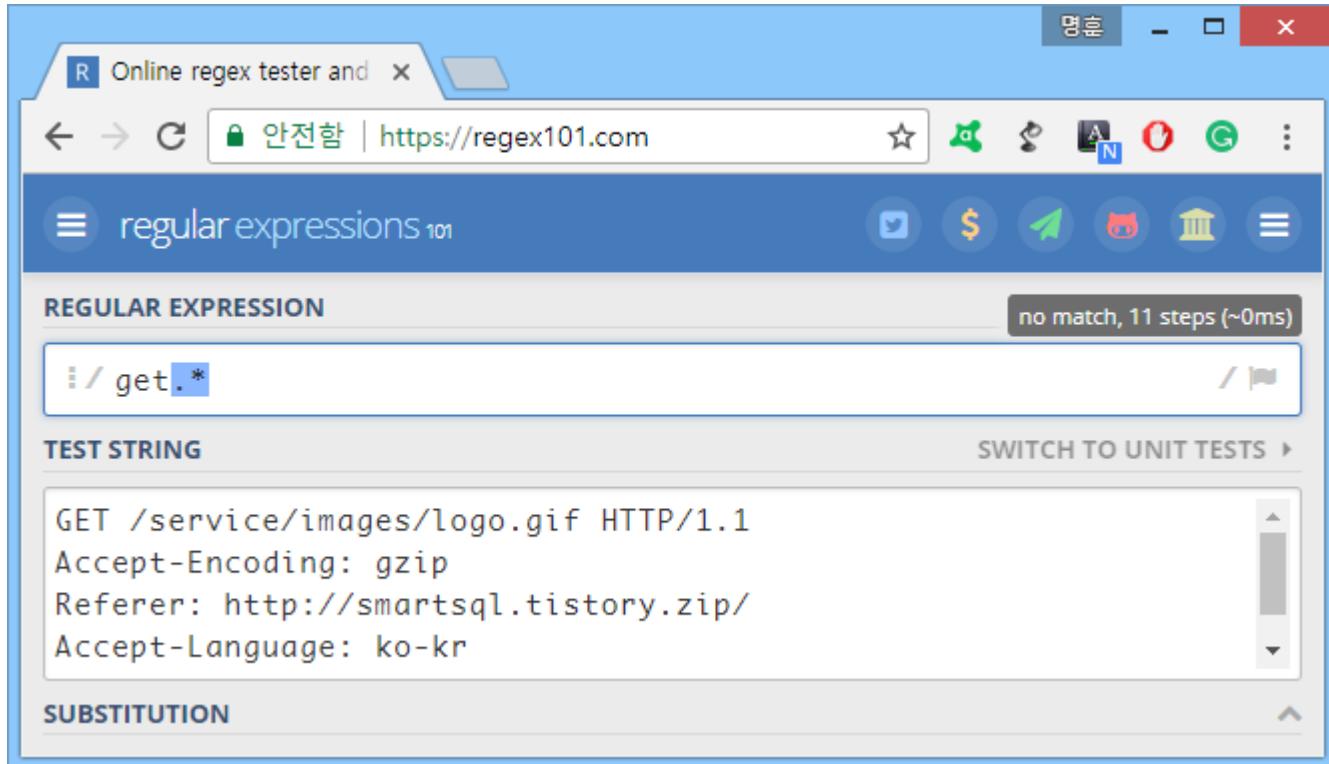
- 추가 수정자(Snort only)

옵션	검사 범위
/U	웹 요청 URI
/H	웹 요청 헤더
/M	웹 요청 method
/C	웹 요청 cookie
/P	POST method로 전송되는 데이터
/S	웹 응답 코드번호
/Y	웹 응답 메시지

# SNORT 룰

## ▶▶ Payload 검사 룰 옵션

- <https://regex101.com/r/d1rj4a/1>
- 대소문자를 구분하는 상태



# SNORT 룰



## ▶▶ Payload 검사 룰 옵션

- pcre 수정자
- 대소문자 구분 해제 수정자(i) 사용

The screenshot shows the 'Online regex tester and visualizer' at <https://regex101.com>. The regular expression input field contains '/get.\*' with a red circle highlighting the 'i' modifier button. The test string below contains a series of HTTP headers. The substitution section is empty.

REGULAR EXPRESSION: /get.\*

TEST STRING:

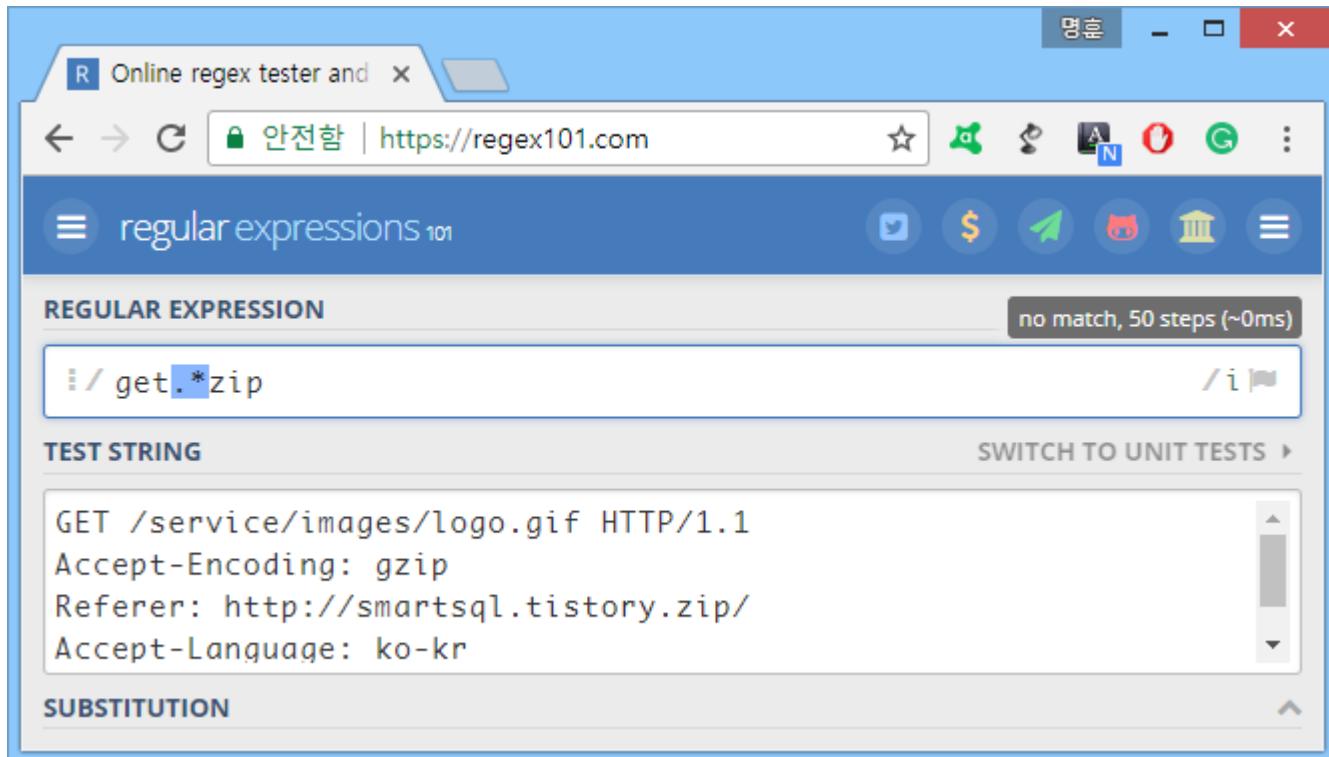
```
GET /service/images/logo.gif HTTP/1.1
Accept-Encoding: gzip
Referer: http://smartsqql.tistory.zip/
Accept-Language: ko-kr
```

SUBSTITUTION:

# SNORT 를

## ▶▶ Payload 검사 를 옵션

- pcre 수정자
- 줄바꿈 문자를 검사할 수 없는 상태



# SNORT 룰



## ▶▶ Payload 검사 룰 옵션

- pcre 수정자
- 줄바꿈 문자 검사 수정자(s) 사용

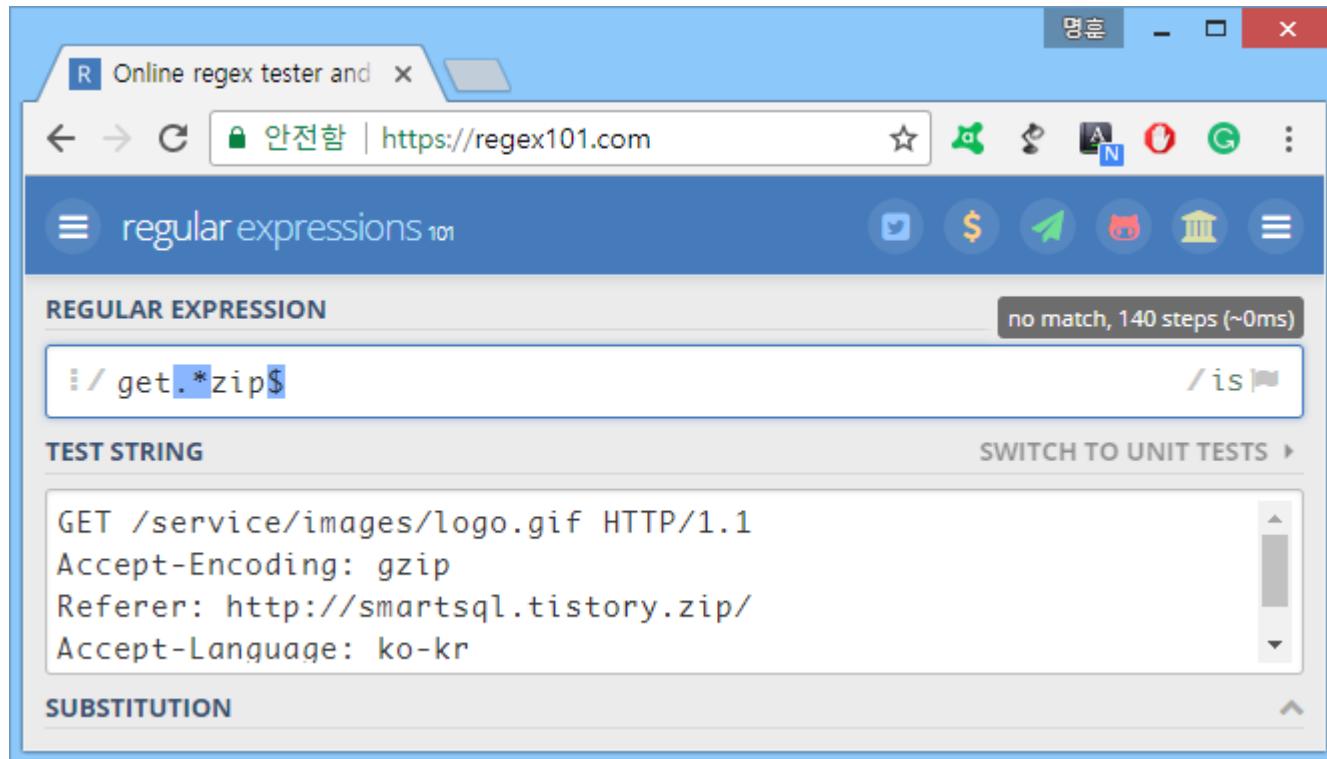
The screenshot shows the 'Online regex tester and' interface at <https://regex101.com>. The regular expression input field contains `/get.*zip`, and the test string input field contains:  
GET /service/images/logo.gif HTTP/1.1  
Accept-Encoding: gzip  
Referer: http://smartsqql.tistory.zip/  
Accept-Language: ko-kr

A red circle highlights the modifier `/is` in the regular expression input field.

# SNORT 를

## ▶▶ Payload 검사 를 옵션

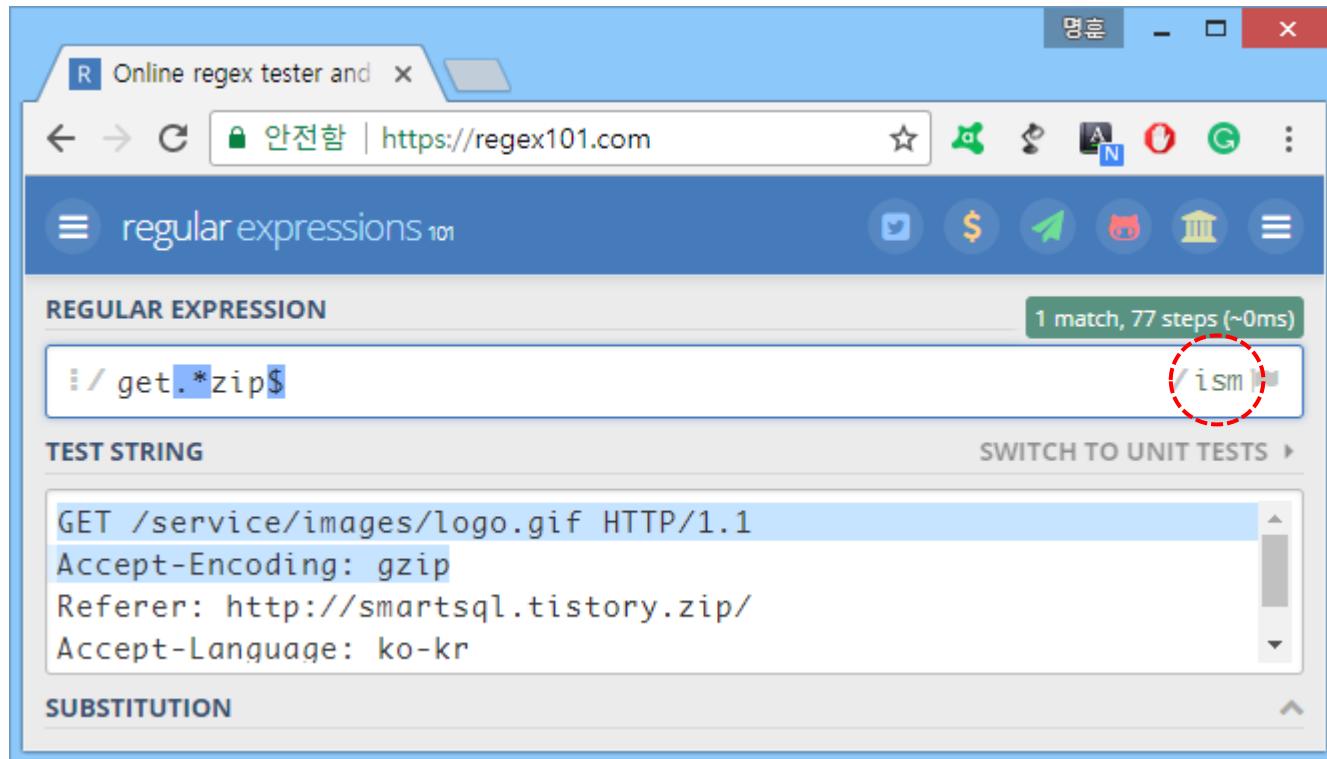
- pcre 수정자
- 앵커문자(^, \$) 줄 구분 상태



# SNORT 를

## ▶▶ Payload 검사 를 옵션

- pcre 수정자
- 앵커문자(^, \$) 줄 구분 해제 수정자(m) 사용





## ▶▶ Payload 검사 를 옵션

- Wireshark 문자열 검사 옵션 ‘contains’
- 대소문자 구분, 정규표현식 사용 불가

No.	Time	Protocol	Sip	Dip	Sport	Dport	Info
1...	2011-04-10 01:25:20...	HTTP	10.184.27.17	10.168.0.3	80	62674	HTTP/1.1 200 OK (t
2...	2011-04-10 01:25:21...	HTTP	10.184.27.17	10.168.0.3	80	62686	HTTP/1.1 200 OK (a
2...	2011-04-10 01:25:21...	HTTP	10.184.27.17	10.168.0.3	80	62685	HTTP/1.1 200 OK (a
4...	2011-04-10 01:26:11...	HTTP	10.239.17.244	10.168.0.3	80	62743	HTTP/1.1 200 OK (t
5...	2011-04-10 01:27:21...	HTTP	10.222.1.43	10.168.0.3	80	62796	HTTP/1.1 200 OK (t
6...	2011-04-10 01:27:23...	HTTP	10.68.15.236	10.168.0.3	80	62802	HTTP/1.1 200 OK (t
7...	2011-04-10 01:27:23...	HTTP	10.68.15.236	10.168.0.3	80	62813	HTTP/1.1 200 OK (a
8...	2011-04-10 01:27:23...	HTTP	10.68.15.236	10.168.0.3	80	62811	HTTP/1.1 200 OK (a
9...	2011-04-10 01:28:34...	HTTP	10.222.1.43	10.168.0.3	80	62862	HTTP/1.1 200 OK (t
9...	2011-04-10 01:28:34...	HTTP	10.67.222.162	10.168.0.3	80	62865	HTTP/1.1 200 OK (t
1...	2011-04-10 01:28:35...	HTTP	10.67.222.162	10.168.0.3	80	62872	HTTP/1.1 200 OK (t
2...	2011-04-10 01:28:57...	HTTP	10.34.181.60	10.168.0.3	80	63021	HTTP/1.1 200 OK (t
2...	2011-04-10 01:28:58...	HTTP	10.34.181.60	10.168.0.3	80	63028	HTTP/1.1 200 OK (t
2...	2011-04-10 01:29:04...	HTTP	10.168.0.3	10.34.181.96	63032	80	GET /commun/update/
2...	2011-04-10 01:31:10...	HTTP/XML	10.46.49.132	10.168.0.3	80	63123	HTTP/1.1 200 OK
2...	2011-04-10 01:31:10...	HTTP/XML	10.46.49.132	10.168.0.3	80	63121	HTTP/1.1 200 OK
2...	2011-04-10 01:31:10...	HTTP/XML	10.46.49.132	10.168.0.3	80	63126	HTTP/1.1 200 OK

# SNORT 를



## ▶▶ Payload 검사 를 옵션

- Wireshark 문자열 검사 옵션 ‘matches’
- 정규표현식 사용 가능, 대소문자 구분 안함

No.	Time	Protocol	Sip	Dip	Sport	Dport	Info
24	2011-04-10 01:23:02...	HTTP	10.168.0.3	10.121.177.30	62582	80	GET /updates/ueupda
34	2011-04-10 01:23:41...	HTTP	10.168.0.3	10.191.95.153	62602	80	GET /deploy/visual-
44	2011-04-10 01:24:35...	HTTP	10.168.0.3	10.254.192.211	62631	80	GET / HTTP/1.1
54	2011-04-10 01:24:35...	HTTP	10.168.0.3	10.254.192.211	62632	80	GET /service/images
57	2011-04-10 01:24:35...	HTTP	10.168.0.3	10.254.251.115	62633	80	GET /magpie/opencou
78	2011-04-10 01:24:35...	HTTP	10.168.0.3	10.254.192.211	62634	80	GET /service/images
1...	2011-04-10 01:25:19...	HTTP	10.168.0.3	10.184.27.17	62669	80	GET /pricing/corpor
1...	2011-04-10 01:25:19...	HTTP	10.168.0.3	10.184.27.17	62670	80	GET /favicon.ico HT
1...	2011-04-10 01:25:20...	HTTP	10.168.0.3	10.184.27.17	62674	80	GET /pricing/corpor
1...	2011-04-10 01:25:20...	HTTP	10.184.27.17	10.168.0.3	80	62674	HTTP/1.1 200 OK (t
1...	2011-04-10 01:25:21...	HTTP	10.168.0.3	10.184.27.17	62686	80	GET /css/printdate.
1...	2011-04-10 01:25:21...	HTTP	10.168.0.3	10.184.27.17	62685	80	GET /css/MM_script.
1...	2011-04-10 01:25:21...	HTTP	10.168.0.3	10.184.27.17	62678	80	GET /css/vdsw.css H
1...	2011-04-10 01:25:21...	HTTP	10.168.0.3	10.184.27.17	62687	80	GET /css/product.cs
1...	2011-04-10 01:25:21...	HTTP	10.168.0.3	10.184.27.17	62682	80	GET /__utm.js HTTP/
1...	2011-04-10 01:25:21...	HTTP	10.168.0.3	10.184.27.17	62688	80	GET /images/home/va
1...	2011-04-10 01:25:21...	HTTP	10.168.0.3	10.184.27.17	62690	80	GET /images/clear.g

# SNORT 를



## ▶▶ Payload 검사 를 옵션

- Wireshark 문자열 검사 옵션 ‘matches’
- 줄바꿈 문자를 검사할 수 없는 상태

The screenshot shows the Wireshark interface with the following details:

- File Menu:** File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help.
- Toolbar:** File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help.
- Search Bar:** http matches "get.\*zip"
- Table Headers:** No., Time, Protocol, Sip, Dip, Sport, Dport, Info.
- Table Data:** A list of 8 captured packets related to HTTP requests for files ending in ".zip".
- Packet Details:** Shows the selected packet's details, including:
  - Frame 2445: 683 bytes on wire (5464 bits), 683 bytes captured (5464 bits).
  - Ethernet II, Src: HonHaiPr\_65:3c:44 (5c:ac:4c:65:3c:44), Dst: 00:0c:29:00:00:00 (00:0c:29:00:00:00)
  - Internet Protocol Version 4, Src: 10.168.0.3, Dst: 10.90.198.96
  - Transmission Control Protocol, Src Port: 63159, Dst Port: 80
  - HyperText Transfer Protocol
- Hex Editor:** Displays the raw hex and ASCII data of the selected packet.
- Status Bar:** Packets: 5044 · Displayed: 8 (0.2%) · Profile: Default



## ▶▶ Payload 검사 를 옵션

- Wireshark 문자열 검사 옵션 ‘matches’
- 줄바꿈 문자 검사 수정자(s) 사용

web\_test.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http matches "(?s)get.\*zip"

No.	Time	Protocol	Sip	Dip	Sport	Dport	Info
34	2011-04-10 01:23:41...	HTTP	10.168.0.3	10.191.95.153	62602	80	GET /deploy/visual-
44	2011-04-10 01:24:35...	HTTP	10.168.0.3	10.254.192.211	62631	80	GET / HTTP/1.1
54	2011-04-10 01:24:35...	HTTP	10.168.0.3	10.254.192.211	62632	80	GET /service/images
57	2011-04-10 01:24:35...	HTTP	10.168.0.3	10.254.251.115	62633	80	GET /magpie/opencou
78	2011-04-10 01:24:35...	HTTP	10.168.0.3	10.254.192.211	62634	80	GET /service/images
1...	2011-04-10 01:25:19...	HTTP	10.168.0.3	10.184.27.17	62669	80	GET /pricing/corpor
1...	2011-04-10 01:25:19...	HTTP	10.168.0.3	10.184.27.17	62670	80	GET /favicon.ico HT

> Frame 44: 383 bytes on wire (3064 bits), 383 bytes on wire (3064 bits)  
> Ethernet II, Src: HonHaiPr\_65:3c:44 (5c:ac:4c:65:3c:44), Dst: 00:0c:29:1f:00:00 (00:0c:29:1f:00:00)  
> Internet Protocol Version 4, Src: 10.168.0.3, Dst: 10.254.192.211  
> Transmission Control Protocol, Src Port: 62631, Dst Port: 80  
< Hypertext Transfer Protocol  
 > GET / HTTP/1.1\r\n  
 Accept: \*/\*\r\n  
 Accept-Language: ko-KR\r\n  
 Accept-Encoding: gzip, deflate\r\n  
 User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET4.0C; .NET4.0E)

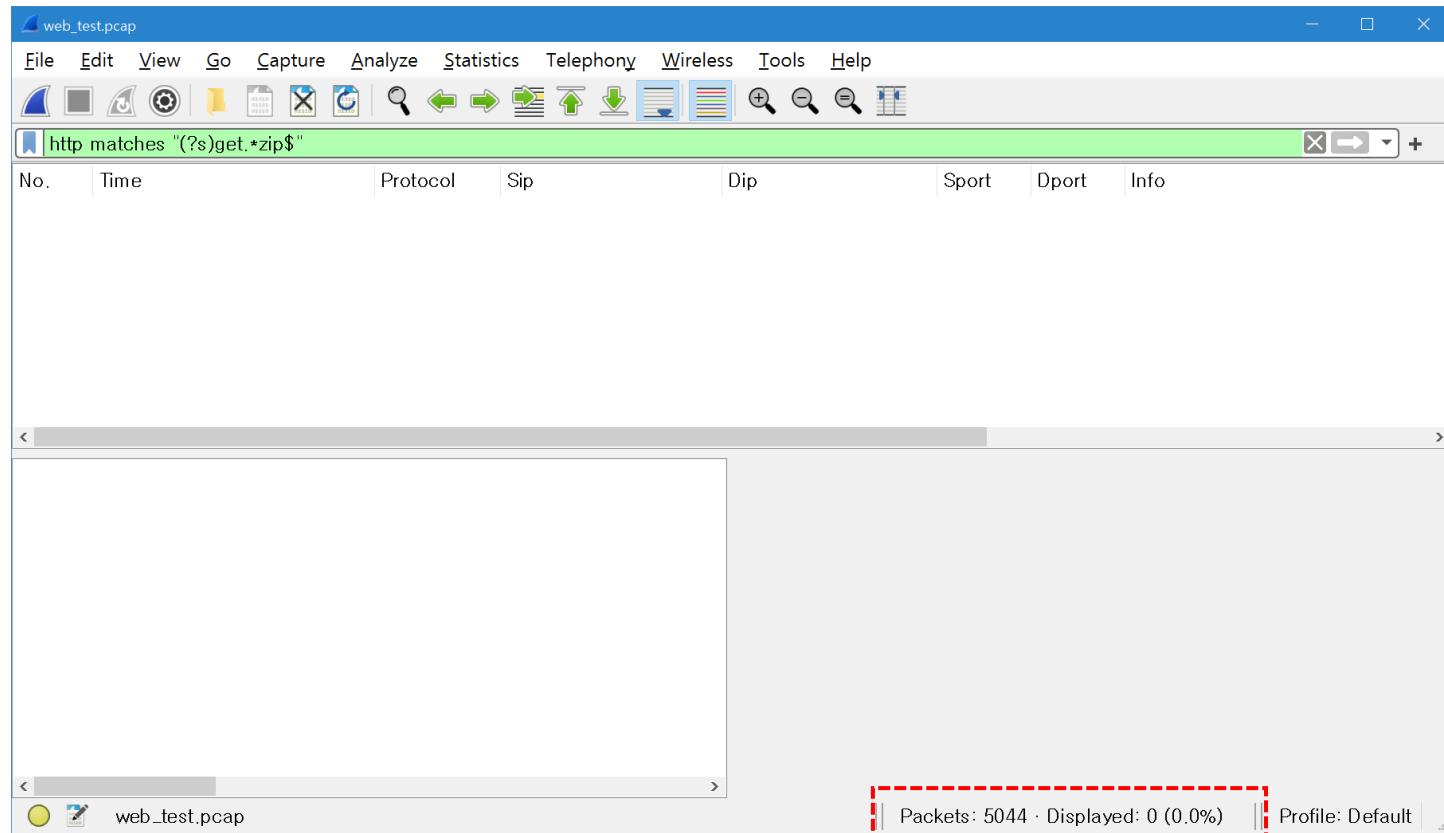
0000 00 26 66 13 c7 69 5c ac 4c 65 3c 44  
0010 01 71 10 db 40 00 80 06 12 30 0a a8  
0020 c0 d3 f4 a7 00 50 da 85 24 af 8d 27  
0030 11 1c 2f 46 00 00 47 45 54 20 2f 20  
0040 2f 31 2e 31 0d 0a 41 63 63 65 70 74  
0050 2a 0d 0a 41 63 63 65 70 74 2d 4c 61  
0060 67 65 3a 20 6b 6f 2d 4b 52 0d 0a 41  
0070 74 2d 45 6e 63 6f 64 69 6e 67 3a 20  
0080 2c 20 64 65 66 6c 61 74 65 0d 0a 55  
0090 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c  
00a0 2e 30 20 28 63 6f 6d 70 61 74 69 62

Packets: 5044 · Displayed: 265 (5.3%) Profile: Default

# SNORT 를

## ▶▶ Payload 검사 를 옵션

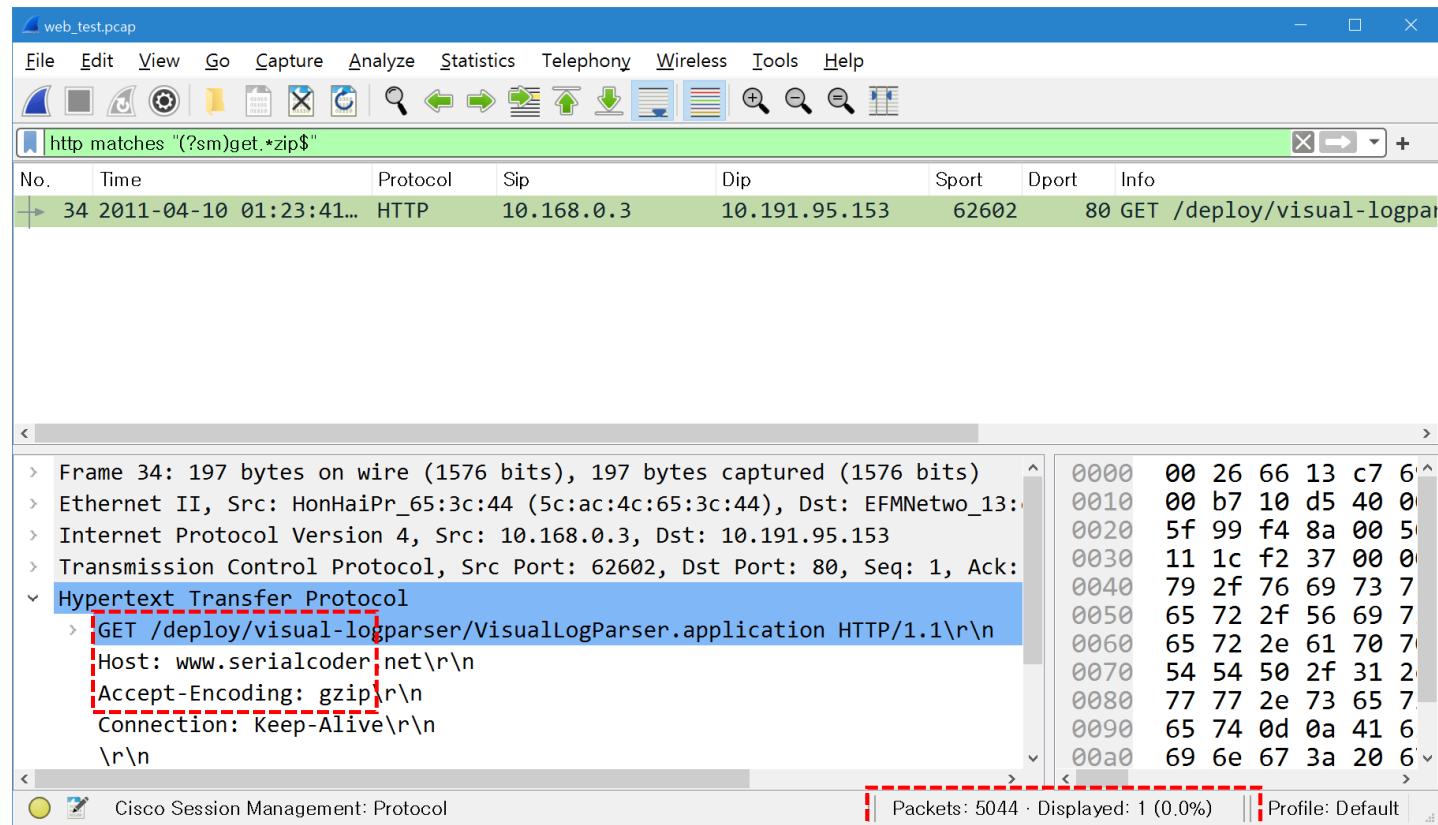
- Wireshark 문자열 검사 옵션 ‘matches’
- 앵커문자(^, \$) 줄 구분 상태



# SNORT 를

## ▶▶ Payload 검사 를 옵션

- Wireshark 문자열 검사 옵션 ‘matches’
- 앵커문자(^, \$) 줄 구분 해제 수정자(m) 사용



# SNORT 를



## ▶▶ Payload 검사 를 옵션

- `fast_pattern` (Aho-Corasick 패턴매칭 우선 순위 조정)

## ▶▶ 예시

- `content:"bb"; fast_pattern; content:"aaa";`  
'bb'로 Aho-Corasick 패턴매칭 후, 'bb'와 'aaa'를 순서대로 룰 패턴매칭 검사

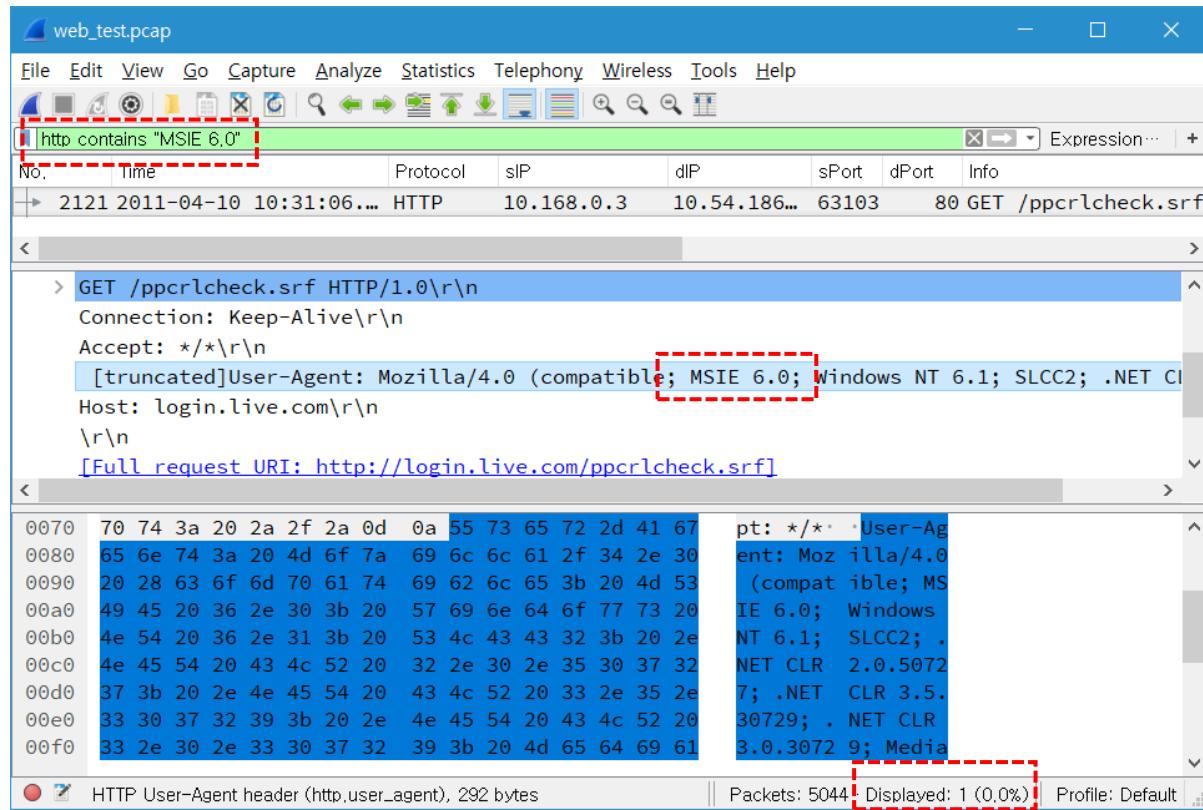
- `content:"bb"; fast_pattern:only; content:"aaa";`  
'bb'로 Aho-Corasick 패턴매칭 후, 'aaa'만 룰 패턴매칭 검사

# SNORT 를



## ▶▶ Payload 검사 를 옵션

- file\_data
- 이진 파일 및 BASE64 인코딩 등 텍스트가 아닌 영역까지 검사 확장



# SNORT 를



## ▶▶ Payload 검사 를 옵션

- file\_data
- alert tcp any any -> any any (msg:"file\_data-test"; file\_data; content:"MSIE 6.0"; sid:1000007; rev:1;)

The screenshot shows a Wireshark capture window titled "web\_test.pcap". A search bar at the top contains the expression "http.file\_data contains \"MSIE 6.0\"". The packet list table shows a single packet (No. 32) from 2011-04-10 10:33:23. The packet details pane shows the content of the file data:

```
[Request in frame: 3180]
[Next request in frame: 3377]
[Next response in frame: 3443]
Content-encoded entity body (gzip): 1244 bytes -> 3136 bytes
File Data: 3136 bytes
Line-based text data: application/x-javascript (1 lines)
[truncated]/* Copyright (C) 2011 Microsoft Corporation */if(!window.Silverlight)window.!
0360 7d 3b 69 66 28 62 2e 69 6e 64 65 78 4f 66 28 22
0370 57 69 6e 64 6f 77 73 20 4e 54 22 29 3e 3d 30 7c
0380 7c 62 2e 69 6e 64 65 78 4f 66 28 22 4d 6f 7a 69
0390 6c 6c 61 2f 34 2e 30 20 28 63 6f 6d 70 61 74 69
03a0 62 6c 65 3b 20 4d 53 49 45 20 36 2e 30 29 22 29
03b0 3e 3d 30 29 61 2e 4f 53 3d 22 57 69 6e 64 6f 77
03c0 73 22 3b 65 6c 73 65 20 69 66 28 62 2e 69 6e 64
03d0 65 78 4f 66 28 22 50 50 43 20 4d 61 63 20 4f 53
};if(b.indexOf("Windows NT")>=0||b.indexOf("Mozilla/4.0 (compatible; MSIE 6.0)")>=0||b.indexOf("Windows")>=0||b.indexOf("Mac OS X")>=0);else if(b.indexOf("PPC Mac OS")>=0){
```

At the bottom, it says "Frame (259 bytes) Reassembled TCP (1665 bytes) Uncompressed entity body (3136 bytes)".

# SNORT 룰



## ▶▶ Payload 검사 룰 옵션

- byte\_test(특정 위치의 byte 크기 검사)

byte\_test:**convert\_scope**, **operator**, **value**, **offset**

②

③

④

①

- operator(연산자) : >, <, >=, <=, =, &(AND), ^(OR)

## ▶▶ 예시

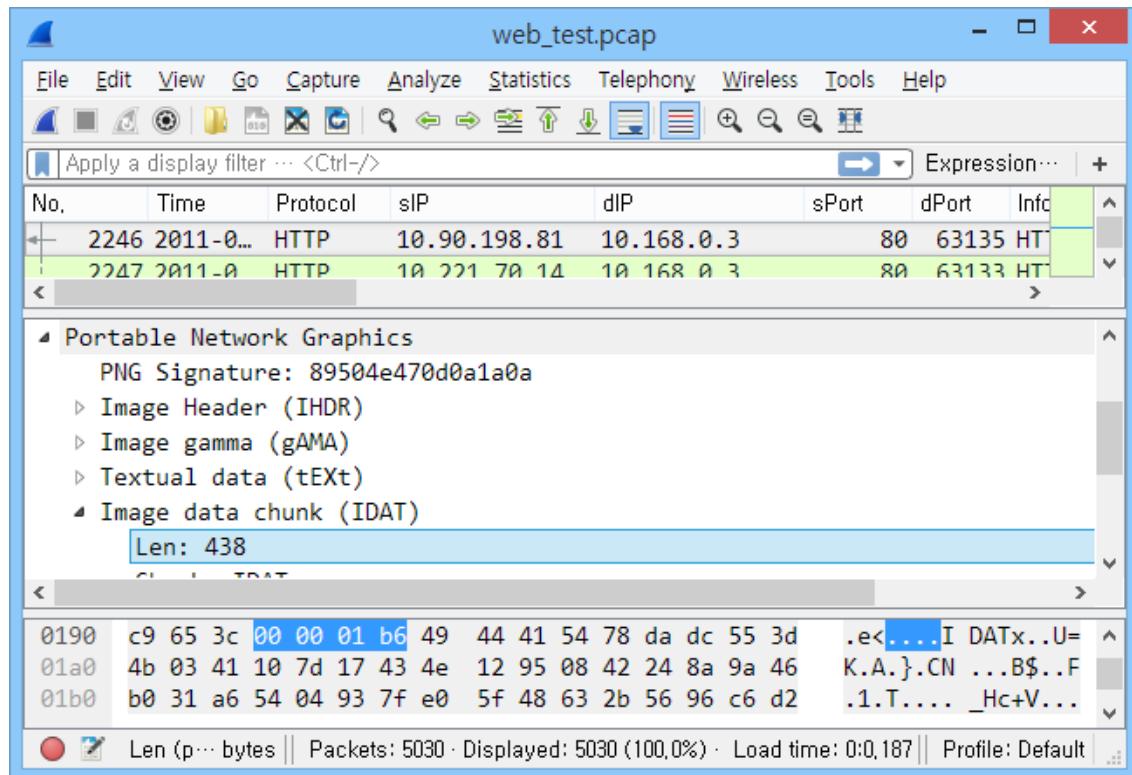
- content:"IDAT"; byte\_test:4,>,400,-8,relative;
- relative : ‘distance:0’ 과 동일
- alert tcp any any -> any any (msg:"byte\_test-test"; content:"IDAT"; byte\_test:4,>,400,-8,relative; sid:1000008; rev:1;)

# SNORT 룰



## ▶▶ Payload 검사 룰 옵션

- byte\_test
- content:"IDAT"; byte\_test:4,>,400,-8,relative;

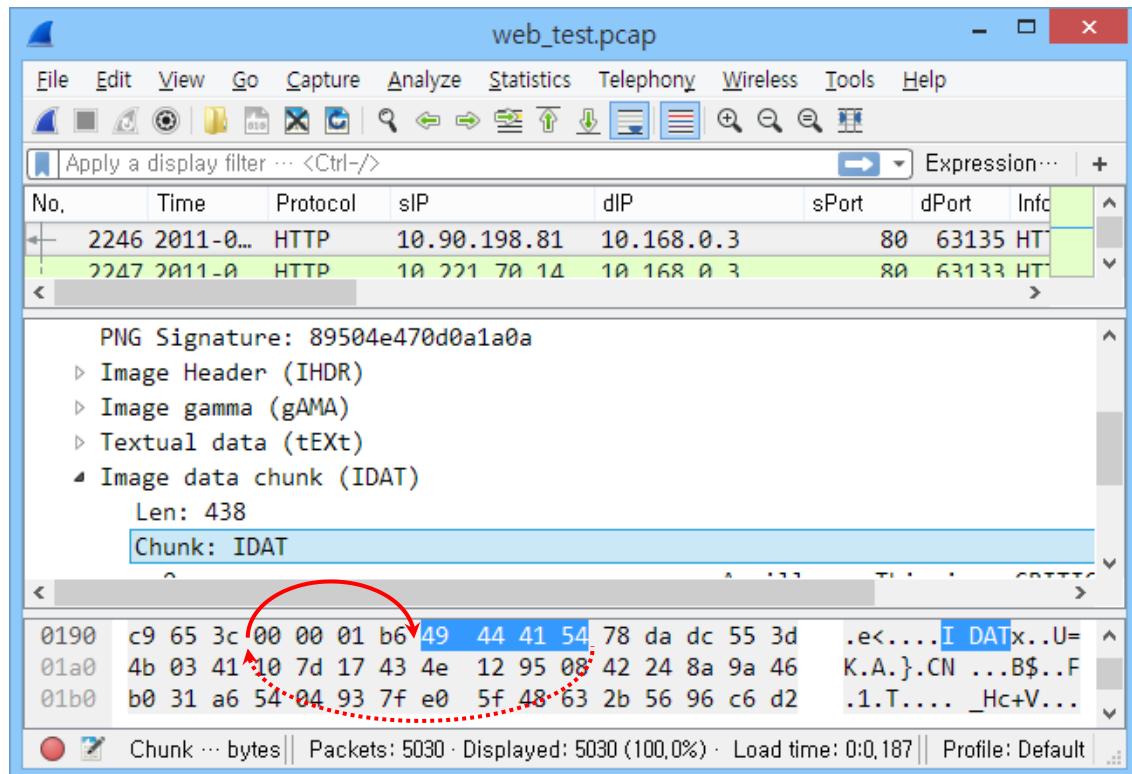


# SNORT 룰



## ▶▶ Payload 검사 룰 옵션

- byte\_test
- content:"IDAT"; byte\_test:4,>,400,-8,relative;



# SNORT 룰



## ▶▶ Payload 검사 룰 옵션

- byte\_jump(특정 위치의 byte 크기만큼 검사 위치 이동)

byte\_jump:<sup>2</sup>convert\_scope, <sup>1</sup>offset

## ▶▶ 예시

- content:"IHDR"; content:"tEXt"; distance:0;
- content:"IHDR"; byte jump:4,-8,relative; content:"tEXt"; distance:0;

0140	47 0d 0a 1a 0a 00 00 00 0d 49 48 44 52 00 00 00	G..... [IHDR]...
0150	18 00 00 00 18 08 06 00 00 00 e0 77 3d f8 00 00	..... . . . w=...
0160	00 19 74 45 58 74 53 6f 66 74 77 61 72 65 00 41	.tEXtSo ftware.A
0170	64 6f 62 65 20 49 6d 61 67 65 52 65 61 64 79 71	dobe Ima geReadyq

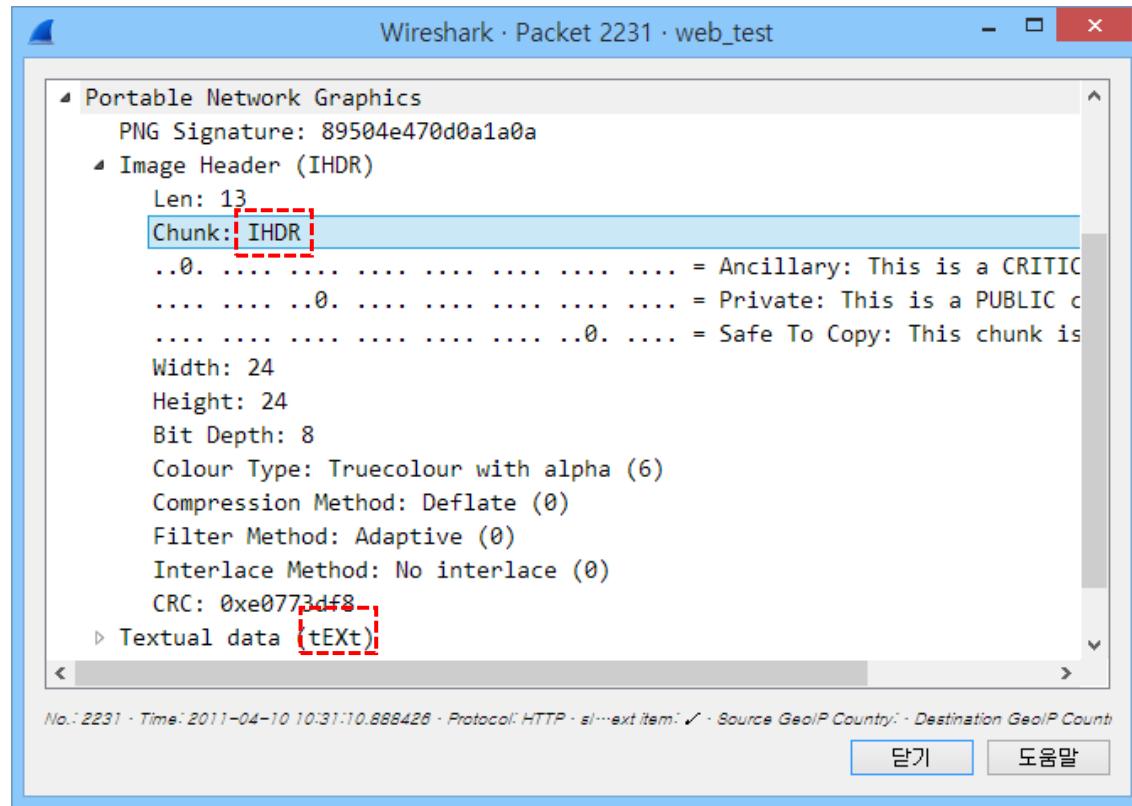
Frame (339 bytes) Reassembled TCP (1745 bytes)

Len (png,...,4 bytes) Packets: 5030 · Displayed: 21 (0.4%) · Load time: 0:0,204 · Profile: Default

# SNORT 를

## ▶▶ Payload 검사 를 옵션

- byte\_jump
- content:"IHDR"; byte\_jump:4,-8,relative; content:"tEXt"; distance:0;

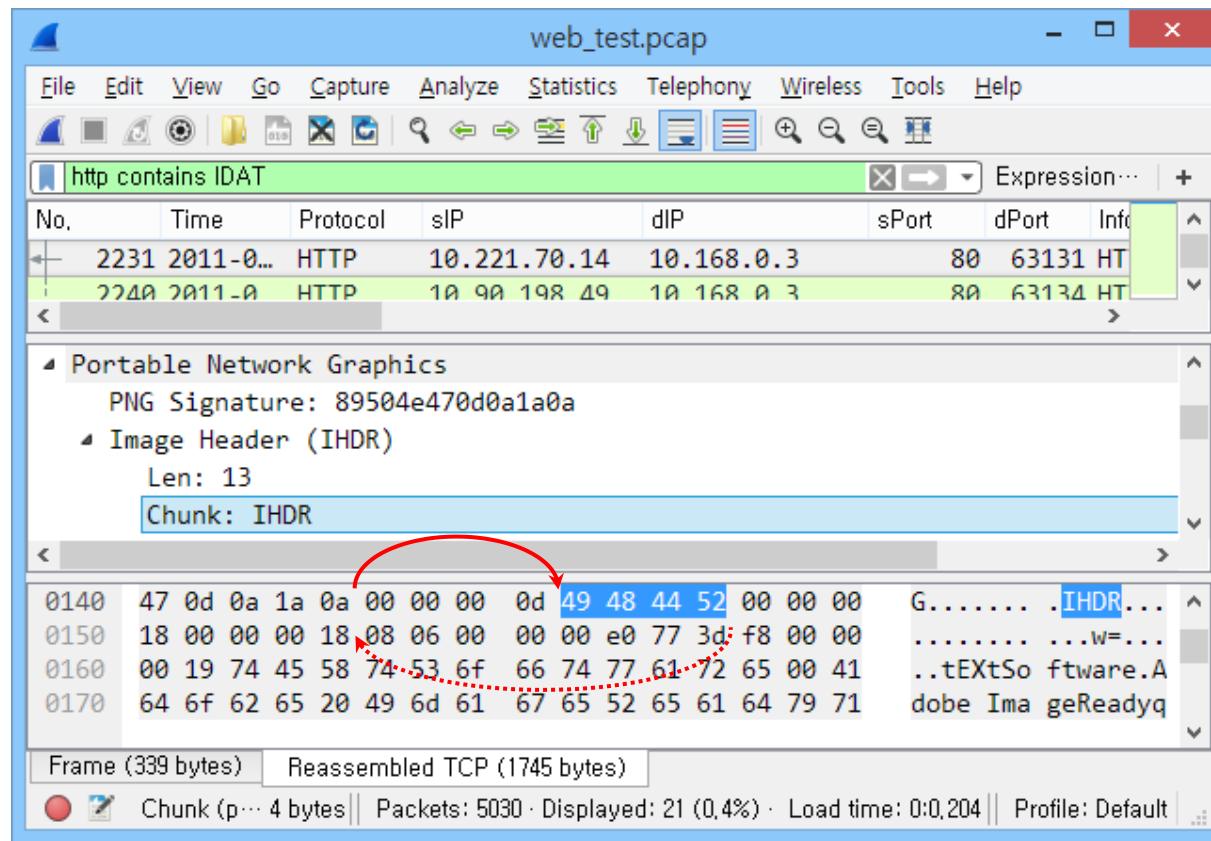


# SNORT 를



## ▶▶ Payload 검사 를 옵션

- byte\_jump
- content:"IHDR"; byte\_jump:4,-8,relative; content:"tEXt"; distance:0;



# PCRE



# 정규표현식



## ▶ 데이터와 정규표현식

- 무한 조합의 문자열을 일정한 규칙으로 표현

```
transforms.conf (C:\Splunk\etc\system\default) - GVIM1
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
파일 문서 새 문서 새 창 편집 터미널 험금 캐시 히스토리 키보드 설정 도움말 ? ?
126 [url]
127 # matches a url of the form proto://domain.tld/uri
128 # Extracts: url, proto, domain, uri
129 REGEX = (?<url>[[alphanum:proto]]://(?<domain>[a-zA-Z0-9\-.:]++) (?<uri>/[^\\s"]*)? )
130
131 [simple_uri]
132 # matches a uri of the form /path/to/resource?query
133 # Extracts: uri, uri_path, uri_query
134 REGEX = (?<uri>(?<uri_path>[^\\s"?]+)(?:\\?(?<uri_query>[^\\s"]+))?)?
135
136 [uri]
137 # uri = path optionally followed by query [/this/path/file.js?query=part&other=var]
138 # path = root part followed by file      [/root/part/file.part]
139 # Extracts: uri, uri_path, uri_root, uri_file, uri_query, uri_domain (optional if in proxy mode)
140 REGEX = (?<uri>(?:\\w++://(?<uri_domain>[^\\s"]++))?(?<uri_path>(?<uri_root>/+(?:[^\\s"?=/]*+/+)*)(?<uri_file>[^\\s"?=/]*+))(?:\\?(?<uri_query>[^\\s"]+))?)?
```

134,1 31%

# 정규표현식



## ▶▶ 메타문자

- 정규표현식의 자음과 모음

## ▶▶ 분류

유형	동작 방식
문자열 검사	고정된 문자열 범위를 검사 가변적인 문자열 범위를 검사 검사(시작 또는 끝나는) 위치 검사
검사 방식 조절	검사 방식 지정(수정자) 검사 수량 지정(수량자)

# 정규표현식



## ▶ 메타문자

- 고정된 문자열 범위를 검사하는 메타문자
- <https://regex101.com/r/c7B3tk/1>

PCRE	VIM	검사 범위	비고
.		'줄바꿈 문자'를 제외한 모든 문자(공백 포함)	
\s		'탭, 줄바꿈 문자' 등을 포함한 모든 공백 문자 (VIM은 '줄바꿈 문자'를 포함하지 않음)	
\S		공백을 제외한 모든 문자	
\d		숫자	
\D		숫자를 제외한 모든 문자	대소문자를 이용해서 검사범위 반전
\w		알파벳 대소문자와 숫자, _	
\W		\w의 검사 범위를 제외한 모든 문자	
\r		새 줄을 시작하는 제어문자	
\n		줄이 끝났음을 표시하는 제어문자	줄바꿈 문자

# 정규표현식



## ▶ 메타문자

- 줄바꿈 문자

Word	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
Dec	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Hex	0	1	2	3	4	5	6	7	8	9	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
Word	SPA CE	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
Dec	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Hex	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	3A	3B	3C	3D	3E	3F	
Word	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
Dec	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
Hex	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
Word	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
Dec	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
Hex	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F

# 정규표현식



## ▶ 메타문자

- 줄바꿈 문자

Wireshark · Packet 6 · web\_test

▀ Hypertext Transfer Protocol

  ▷ HTTP/1.1 200 OK\r\n

    Date: Sun, 10 Apr 2011 01:21:55 GMT\r\n

    Server: Apache/2.2.9 (Debian) PHP/5.2.6-1+lenny9 with Suhosin-Patch mod\_ss

    X-Powered-By: PHP/5.2.6-1+lenny9\r\n

  ▷ Content-Length: 84\r\n

00c0 65 6e 53 53 4c 2f 30 2e 39 2e 38 67 0d 0a 58 2d enSSL/0. 9.8g..X-\n00d0 50 6f 77 65 72 65 64 2d 42 79 3a 20 50 48 50 2f Powered- By: PHP/\n00e0 35 2e 32 2e 36 2d 31 2b 6c 65 6e 6e 79 30 0d 0a 5.2.6-1+ lenny9..\n00f0 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 Content- Length:\n0100 38 34 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 84..Cont ent-Type\n0110 3a 20 74 65 78 74 2f 68 74 6d 6c 3b 20 63 68 61 : text/h tml; cha

No. 6 · Time: 2011-04-10 10:21:55.974300 · Protocol: HTTP · s/P: 10. .... Text item: ✓ · Source GeoIP Country: · Destination GeoIP Count

닫기      도움말

# 정규표현식



## ▶ 메타문자

- 가변적 문자열 범위를 검사하는 메타문자
- 문자열 범위 지정 가능

PCRE	명칭	설명
[]	문자 클래스	[ ] 안에 나열된 문자 중 하나를 검사
[^]		[ ] 안에 나열된 문자를 제외한 모든 문자 검사
()	캡쳐 그룹	
(?<이름>)	Named 캡쳐 그룹	( ) 안에 나열된 문자 그룹 검사
(?:)	Not 캡쳐 그룹	
	Alternation	OR 연산자

# 정규표현식



## ▶ 메타문자

- 가변적 문자열 범위를 검사하는 메타문자
- 문자 클래스

The screenshot shows two instances of the regex101.com web application. Both instances have the URL <https://regex101.com> in the address bar.

**Top Window (Left):**

- REGULAR EXPRESSION:** `/abc/`
- TEST STRING:** `abc`
- SUBSTITUTION:** (empty)
- Status:** 1 match, 2 steps (~0ms)

**Bottom Window (Right):**

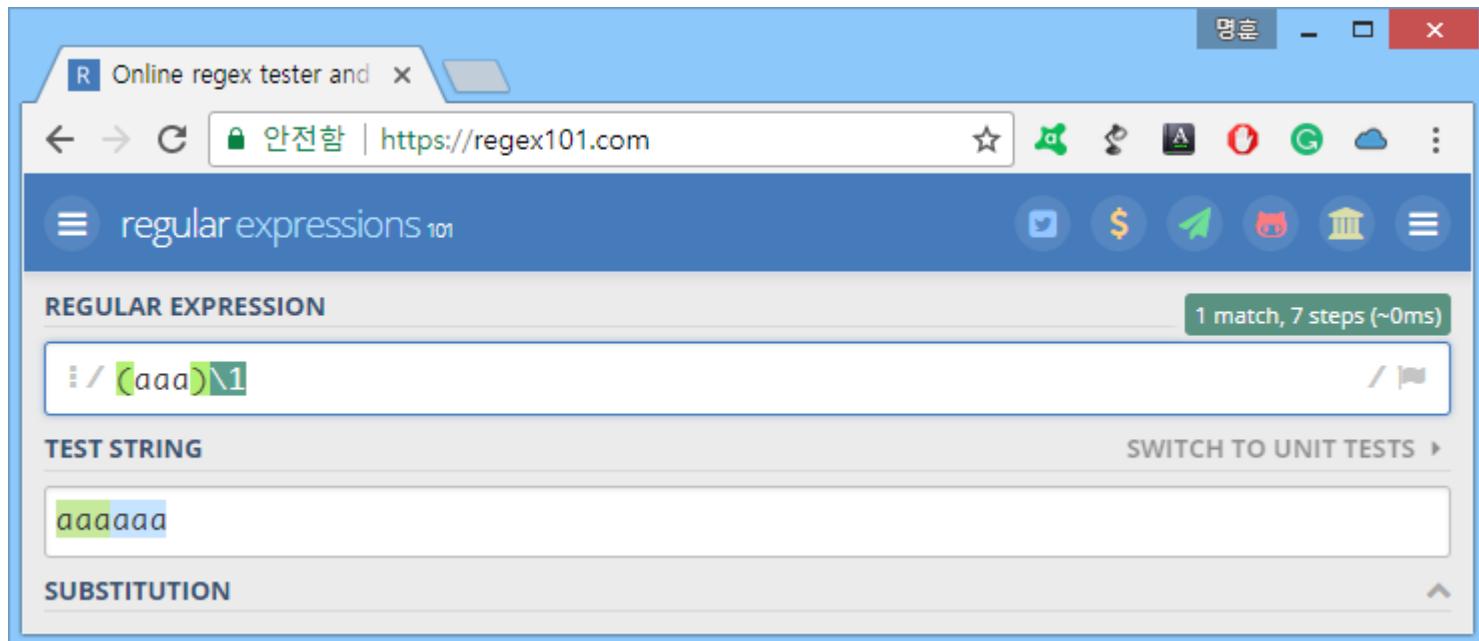
- REGULAR EXPRESSION:** `/abc/g`
- TEST STRING:** `abc`
- SUBSTITUTION:** (empty)
- Status:** 3 matches, 7 steps (~1ms)

A red dashed circle highlights the `/g` character in the regular expression input field of the bottom window.

# 정규표현식

## ▶ 메타문자

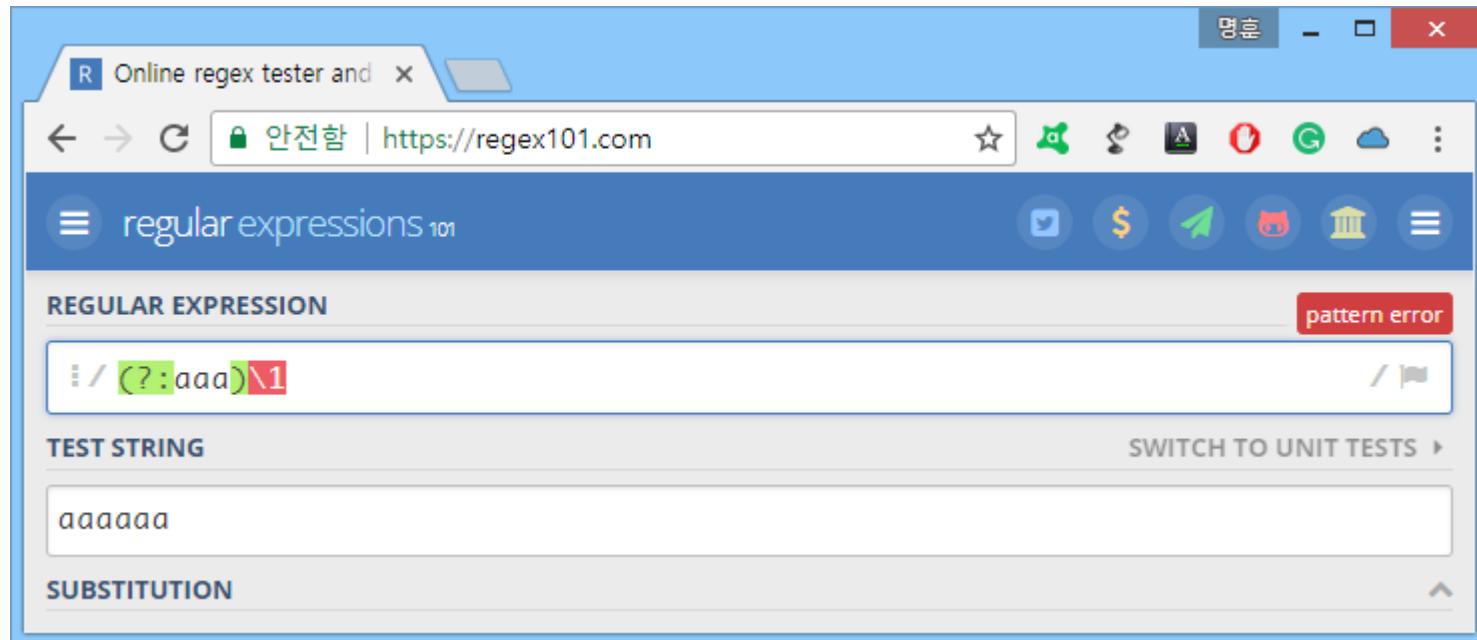
- 가변적 문자열 범위를 검사하는 메타문자
- 캡쳐 그룹



# 정규표현식

## ▶ 메타문자

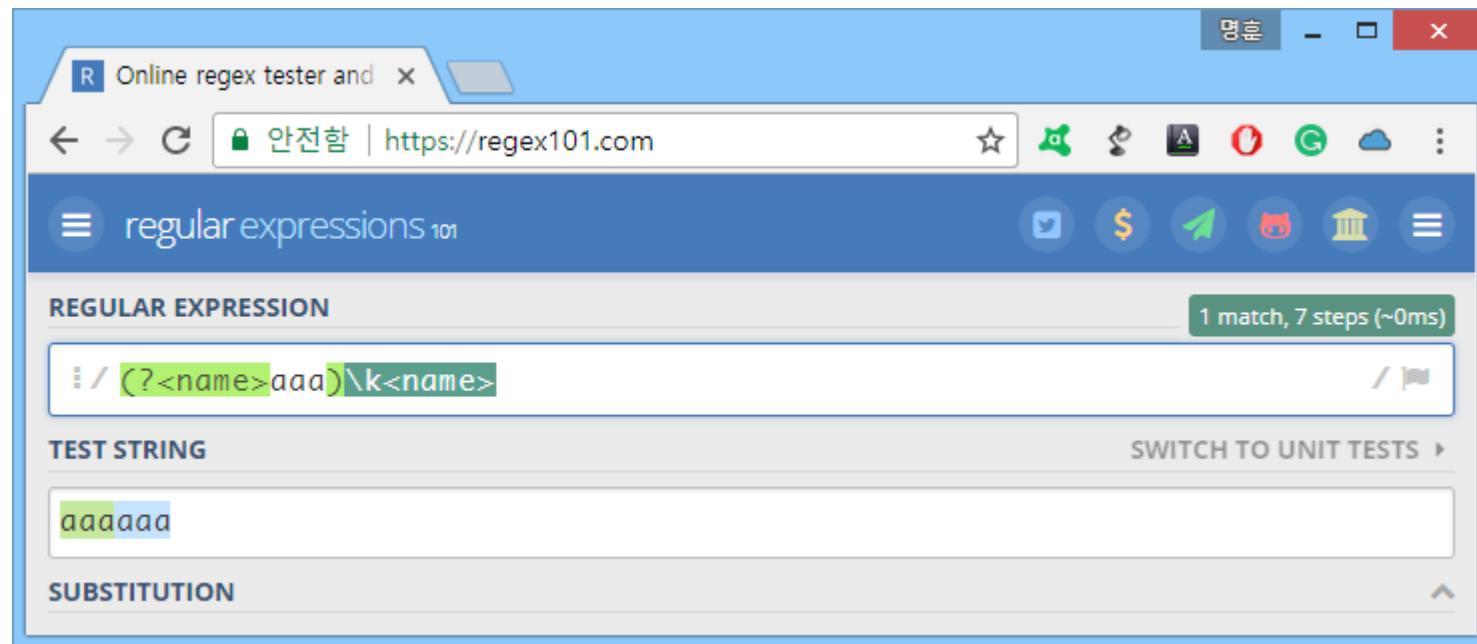
- 가변적 문자열 범위를 검사하는 메타문자
- Not 캡쳐 그룹



# 정규표현식

## ▶ 메타문자

- 가변적 문자열 범위를 검사하는 메타문자
- Named 캡쳐 그룹



# 정규표현식



## ▶ 메타문자

- 가변적 문자열 범위를 검사하는 메타문자
- OR 연산자

The screenshot displays the [regex101.com](https://regex101.com) interface, which is an online tool for testing regular expressions. It features two separate test environments.

**Top Test:**

- REGULAR EXPRESSION:** `/a|bc`
- TEST STRING:** `abc`
- SUBSTITUTION:** (empty)
- Result:** 2 matches, 8 steps (~48ms)

**Bottom Test:**

- REGULAR EXPRESSION:** `/^(a|b)c`
- TEST STRING:** `abc`
- SUBSTITUTION:** (empty)
- Result:** 1 match, 11 steps (~0ms)

# 정규표현식

## ▶ 메타문자

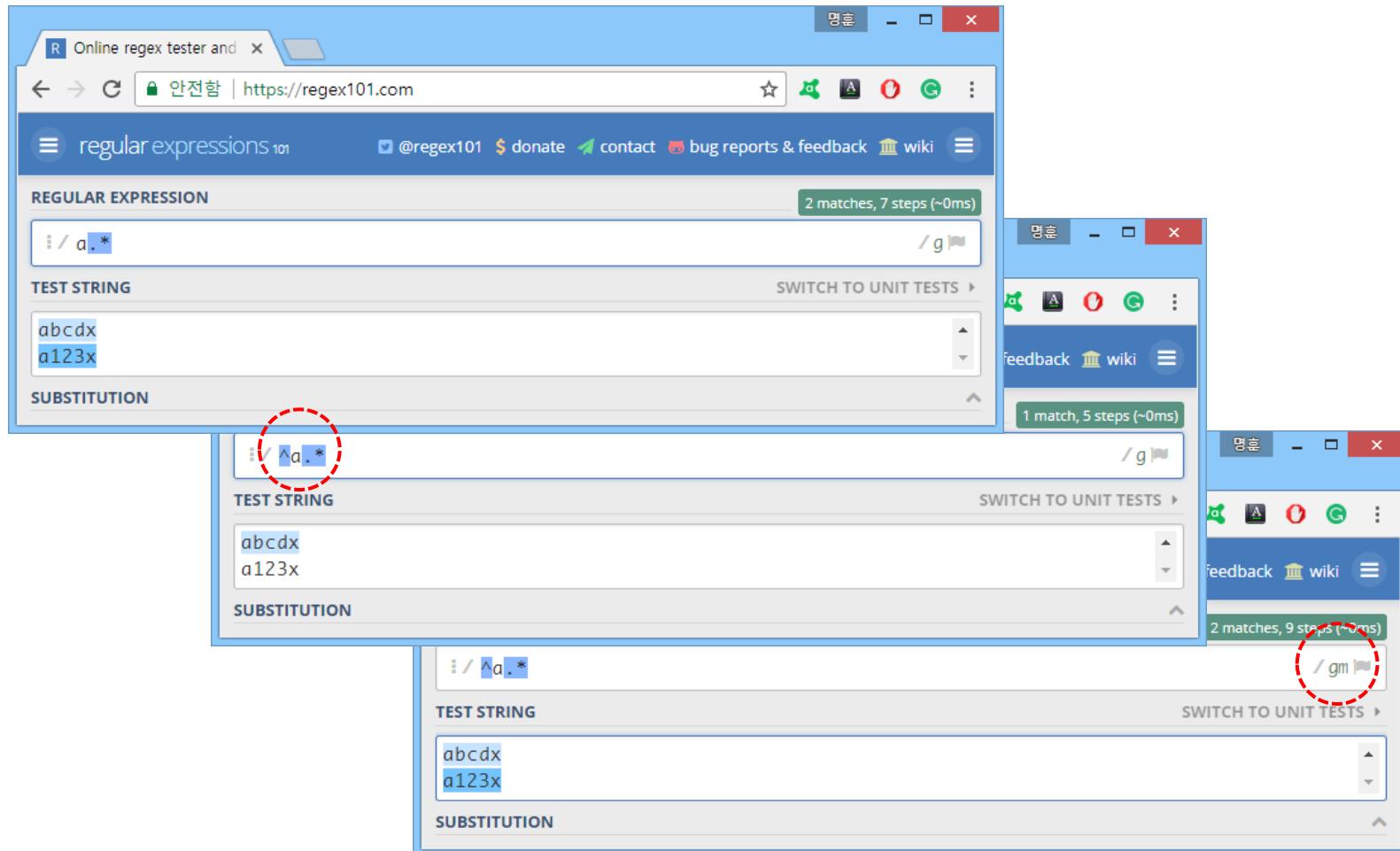
- 검사 위치를 결정짓는 메타문자

PCRE	설명	명칭
^	첫 번째 문자부터 검사(첫 번째 줄에서만 동작)	Anchor 문자
\$	마지막 문자까지 검사(마지막 줄에서만 동작)	
(?=)	검사 종료 위치 지정	긍정형 전방탐색
(?!)		부정형 전방탐색
(?<=)	검사 시작 위치 지정	긍정형 후방탐색
(?<!)		부정형 후방탐색

# 정규표현식

## ▶ 메타문자

- 앵커 문자



# 정규표현식

## ▶ 메타문자

- 앵커 문자

The screenshot shows the regex101.com website interface. At the top, there's a navigation bar with links like 'regular expressions 101', '@regex101', '\$ donate', 'contact', 'bug reports & feedback', 'wiki', and a search bar. Below the navigation, there are two main sections for testing regular expressions.

**Top Section:** The regular expression input field contains `// a.*x$`. The 'TEST STRING' input field contains `abcdx` and `a123x`. The output shows **1 match, 14 steps (~0ms)**. The matched part `a.*x$` is highlighted in blue in both the test string and the output.

**Bottom Section:** The regular expression input field also contains `// a.*x$`. The 'TEST STRING' input field contains the same strings `abcdx` and `a123x`. The output shows **2 matches, 12 steps (~0ms)**. The matched parts are highlighted in blue. In this section, the `/gm` flag at the end of the regex is circled in red.

# 정규표현식



## ▶ 메타문자

- 긍정형 전방탐색 구문      **패턴1(?=패턴2)**
- 패턴2로 끝나는 패턴1을 검사

The screenshot shows the regex101.com interface. The regular expression input field contains `/z(?={})/ig`. The test string input field contains:  
!"#\$%& ' ( )\*+, - ./  
0123456789 : ; <=> ?  
@ABCDEFGHIJKLMNPQRSTUVWXYZ[ \]^\_  
' abcdefghijklmnopqrstuvwxyz z{|}~

The results panel indicates "1 MATCH - 9 STEPS".

# 정규표현식



## ▶ 메타문자

- 부정형 전방탐색 구문      **패턴1(?!패턴2)**
- 패턴2로 끝나지 않는 패턴1을 검사

The screenshot shows the regex101.com interface. The regular expression input field contains `/z(?!{})`. The test string input field contains `!#$%&' ( )*+, -./  
0123456789: ;<=>?  
@ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^_  
' abcdefghijklmnopqrstuvwxyz z{|}~`. The results panel shows "1 MATCH - 9 STEPS" with a green highlight under the character 'z' in the test string.

# 정규표현식



## ▶ 메타문자

- 긍정형 후방탐색 구문 **(?<=패턴1)패턴2**
- 패턴1로 시작하는 패턴2를 검사

The screenshot shows the regex101.com interface. The regular expression input field contains `/(?<=@)a/` with options `ig`. The test string input field contains:  
!"#\$%&'(\*+,-./0123456789:;<=>?  
@ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^\_  
`abcdefghijklmnopqrstuvwxyz{|}~

# 정규표현식



## ▶ 메타문자

- 부정형 후방탐색 구문 **(?<!패턴1)패턴2**
- 패턴1로 시작하지 않는 패턴2를 검사

The screenshot shows the 'Online regex tester and debugger' interface. In the 'REGULAR EXPRESSION' field, the pattern `/(?<!@)a/` is entered with flags `ig`. The 'TEST STRING' field contains the following multi-line text:  
! #\$%& ' ( ) \* + , - . /  
0123456789 : ; < = > ?  
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[ \ ] ^\_  
` abcdefghijklmnopqrstuvwxyz z { | } ~

# 정규표현식



## ▶ 메타문자

- 검사 수량을 결정하는 수량자(Quantifier)



PCRE	설명	비고
*	0개 이상	고정된 검사 수량 (수량을 모를 때 적합)
+	1개 이상	
?	0 또는 1개	
{n}	n개	가변적 검사 수량 (수량을 알 때 적합)
{n,}	n개 이상	
{m,n}	m개 이상, n개 이하	

- 검사 가능한 범위를 최대한 검사하려는 특징이 있음

# 정규표현식



## ▶ 메타문자

- 검사량은 수량자의 보조를 받는 메타문자의 검사 범위에 달려 있음

The screenshot shows the 'Online regex tester and debugger' interface at <https://regex101.com>. The regular expression input field contains the pattern '/.\*'. The test string input field contains the following multi-line text:  
! #\$%& '(. )\*+, -./  
0123456789: ;<=>?  
@ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^\_  
~abcdefghijklmnoprstuvwxyz z{ |}~

- 줄바꿈 문자를 검사하지 못하는 메타문자 .은 줄바꿈 문자 앞에서 검사를 종료한다

# 정규표현식



## ▶ 메타문자

- s(싱글라인) 수정자를 이용해서 메타문자 .의 검사 범위에 줄바꿈 문자를 포함한 결과

The screenshot shows the 'Online regex tester and' interface from regex101.com. The regular expression input field contains `/.*`. The test string input field contains `0123456789; ;<=>?`. The result summary indicates `1 match, 2 steps (~0ms)`. The match is highlighted in the test string at position 10, spanning from the semicolon to the question mark.

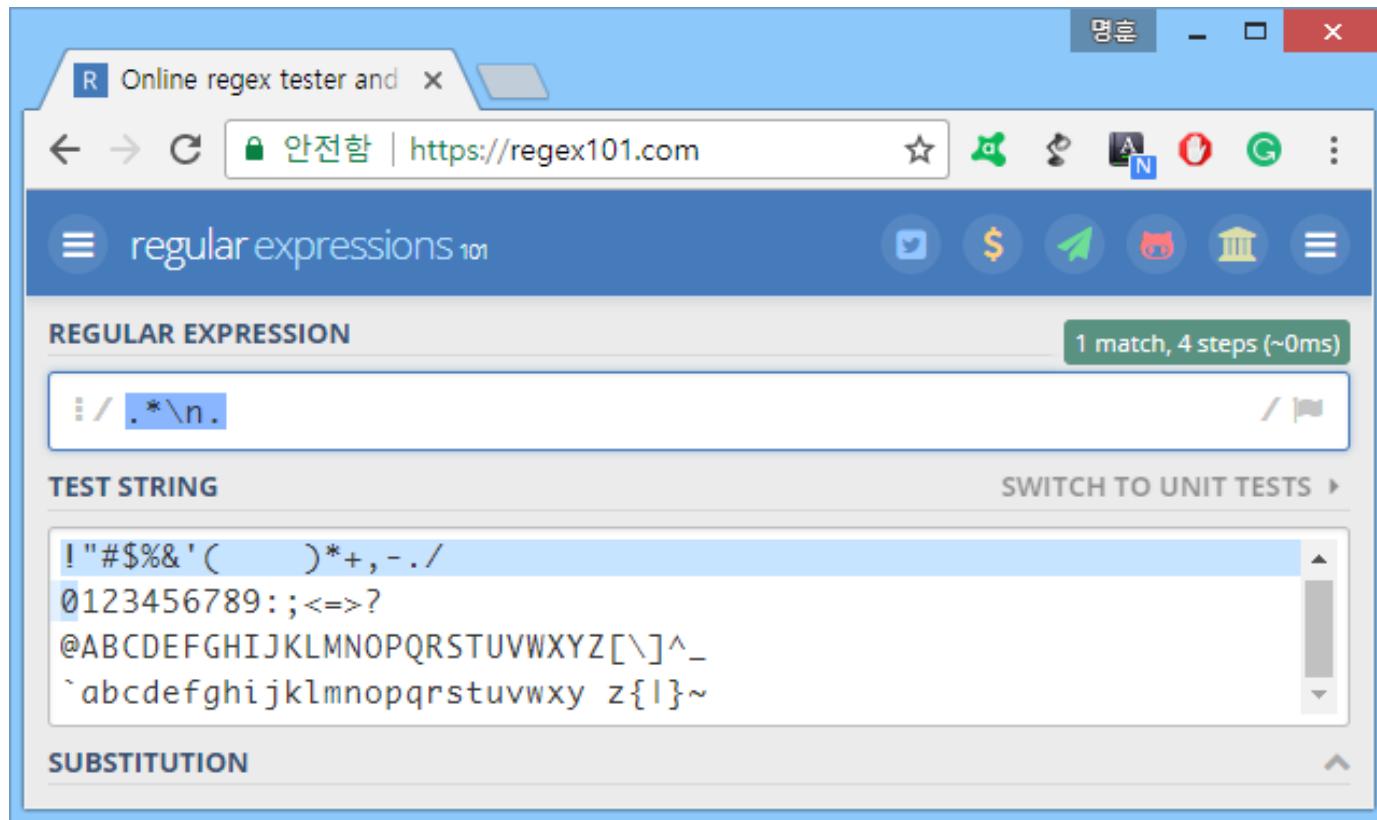
REGULAR EXPRESSION	TEST STRING	SUBSTITUTION
<code>/.*</code>	<code>0123456789; ;&lt;=&gt;?</code>	

# 정규표현식



## ▶ 메타문자

- s(싱글라인) 수정자를 이용하지 않고 새 줄(\n)로 검사를 이어가는 방법



The screenshot shows the 'Online regex tester and' interface at <https://regex101.com>. The regular expression input field contains `.*\n.`. The test string input field contains the following multi-line text:  
! "#\$%&' ( )\*+, - ./  
0123456789: ;<=>?  
@ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^\_  
`abcdefghijklmnoprstuvwxyz z{|}~

# 정규표현식



## ▶ 메타문자

- 최대 수량 검사가 기본이지만, 검사 범위를 최소로 제한할 수 있음

PCRE	설명
*?	최소 0개
+?	최소 1개
??	최소 0개
{n,}?	최소 n개
{m,n}?	최소 m개

# 정규표현식



## ▶ 메타문자

- 최대 수량 검사가 기본이지만, 검사 범위를 최소로 제한할 수 있음

The screenshot shows four separate browser windows or tabs, all titled "Online regex" or "Online regex tester and", demonstrating different regular expression patterns:

- Pattern:** `.*?`  
**Test String:** "#\$%&('0123456789:@ABCDEFGHIJ`abcdefg hij  
The pattern matches the entire string, including the quotes and punctuation.
- Pattern:** `.+?`  
**Test String:** "#\$%&('0123456789:@ABCDEFGHIJ`abcdefg hij  
The pattern matches from the first character to the last, skipping the quotes and punctuation.
- Pattern:** `.??`  
**Test String:** "#\$%&('0123456789:@ABCDEFGHIJ`abcdefg hij  
The pattern matches the first two characters of the string.
- Pattern:** `.{2,5}?`  
**Test String:** "#\$%&('0123456789:@ABCDEFGHIJ`abcdefg hijKLMNOPQRSTU VWXYZ[\]^\_`abcdefg hijklmnopqrstuvwxyz z{|}~  
The pattern matches a range of 2 to 5 characters from the start of the string, including the quotes and punctuation.

# 정규표현식

## ▶ 메타문자

- 최대 수량 검사는 전체 검사 후 하나씩 후퇴

The image shows two side-by-side screenshots of the 'Online regex tester and' tool from regex101.com. Both screenshots have a blue header bar with the URL '안전함 | https://regex101.com'.

**Left Tester ( .\* )**

- REGULAR EXPRESSION:** `/.*`
- TEST STRING:** `!"#$%&'( )*+, -./  
0123456789: ;<=>?  
@ABCDEFGHIJKLMNPQRSTUVWXYZ  
`abcdefghijklmnoprstuvwxyz`
- SUBSTITUTION:** (empty)

**Right Tester ( .\*\\+ )**

- REGULAR EXPRESSION:** `/.*\+`
- TEST STRING:** `!"#$%&'( )*+, -./  
0123456789: ;<=>?  
@ABCDEFGHIJKLMNPQRSTUVWXYZ[\\]^_  
`abcdefghijklmnoprstuvwxyz z{|}~`
- SUBSTITUTION:** (empty)

In the left tester's test string, the character '0' is highlighted in red, and a red arrow points to it from the explanatory text below. In the right tester's test string, the character '0' is also highlighted in red, but a red arrow points to it from the explanatory text above.

**Explanatory Text (Left Tester):**

0은 최대 수량 검사를 통해 전체 문자열을 검사한 후 하나씩 후퇴하는 방식으로 처리되는 특수 문자입니다. 예를 들어, 문자열에 있는 모든 '0'은 모두 선택되는 대상이 됩니다.

**Explanatory Text (Right Tester):**

0은 최대 수량 검사를 통해 전체 문자열을 검사한 후 하나씩 후퇴하는 방식으로 처리되는 특수 문자입니다. 예를 들어, 문자열에 있는 모든 '0'은 모두 선택되는 대상이 됩니다.

# 정규표현식



## ▶ 메타문자

- 최소 수량 검사는 하나씩 검사

The image shows two side-by-side screenshots of the 'Online regex tester and' tool from regex101.com.

**Left Tester:**

- REGULAR EXPRESSION:** `/.*?`
- TEST STRING:** `!"#$%&'( )*+, -./  
0123456789: ;<=>?  
@ABCDEFGHIJKLMNPQRSTUVWXYZ  
~abcdefghijklmnpqrstuvwxyz`
- SUBSTITUTION:** (empty)

**Right Tester:**

- REGULAR EXPRESSION:** `/.*?\+ /`
- TEST STRING:** `!"#$%&'( )*+, -./  
0123456789: ;<=>?  
@ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^_  
~abcdefghijklmnpqrstuvwxyz z{|}~`
- SUBSTITUTION:** (empty)

In the right tester's test string, the portion `0123456789: ;<=>?` is highlighted in red, indicating it is the current target for matching. The result bar at the top right of the right tester says `1 match, 14 steps (~0ms)`.

# 정규표현식

## ▶ 메타문자

- 최소 수량 검사는 하나씩 검사

The screenshot shows a window titled "Regex Debugger" from the website "https://regex101.com/debugger". The window has tabs for "MATCH" and "DEBUG DATA". The "MATCH" tab shows a green checkmark and the message "match 1 found in 15 step(s)". The "DEBUG DATA" tab displays five numbered steps (8, 9, 10, 11, 12) showing the regex engine's state at each step. Each step shows a partial string being processed, with specific characters highlighted in blue and green, indicating the current position of the regex engine.

MATCH

match 1 found in 15 step(s)

DEBUG DATA

8. /.\*?\+ / !"#\$%&'( )\*+, -./
9. /.\*?\+ / !"#\$%&'( )\*+, -./
10. /.\*?\+ / !"#\$%&'( )\*+, -./
11. /.\*?\+ / !"#\$%&'( )\*+, -./
12. /.\*?\+ / !"#\$%&'( )\*+, -./

# VIM 정규표현식





## ▶ VIM 정규표현식

- VIM은 텍스트 우선, PCRE는 정규표현식 우선
- 매직모드(\v)를 이용하면 PCRE와 유사한 정규표현식 사용 가능

```
 SERVER-WEBAPP remote in...#Snort#rule_log - GVIM
파일(E) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
 1 GET /filemanager/filemanager_forms.php?lib_path=http://cirt.net/rfiinc.txt? HTTP/1.1
 2 Host: 192.168.56.100
 3 User-Agent: Mozilla/5.00 (Nikto/2.1.5) (Evasions:None
 ) (Test:000081)
 4 Connection: Keep-Alive
 5
 6 GET /phpBB2/includes/db.php?phpbb_root_path=http://ci
 rt.net/rfiinc.txt? HTTP/1.1
 7 Host: 192.168.56.100
 8 Connection: Keep-Alive
 /\vpath\=(https?|ftps?|php)
```

# 정규표현식



## ▶ VIM 정규표현식

- 가변적 문자열 범위를 검사하는 메타문자
- 문자열 범위 지정 가능

일반모드	매직모드	명칭	설명
[ ]		문자 클래스	[ ] 안에 나열된 문자 중 하나 검사
[^]			[ ] 안에 나열된 문자를 제외한 모든 문자 검사
\(\)	( )	캡쳐 그룹	( ) 안에 나열된 문자 그룹 검사
\%(\)	%()		
\		Alternation	OR 연산자

# 정규표현식



## ▶ VIM 정규표현식

- 검사 위치를 지정해주는 메타문자

일반모드	매직모드	설명	명칭
^		검사 위치를 첫 번째 줄의 시작하는 문자로 고정	Anchor 문자
\$		검사 위치를 마지막 줄의 끝나는 문자로 고정	
\<, \>	<, >	\w의 경계는 \W	경계 문자
\(\)\@=	()@=	검사 종료 위치 지정	긍정형 전방탐색
\(\)\@!	()@!		부정형 전방탐색
\(\)\@<=	()@<=	검사 시작 위치 지정	긍정형 후방탐색
\(\)\@<!	()@<!		부정형 후방탐색

# 정규표현식



## ▶ VIM 정규표현식

- 긍정형 전방탐색 구문      **패턴1(패턴2)@=**
- 패턴2로 끝나는 패턴1을 검사

The screenshot shows the GVIM interface with the title bar "[이름 없음] + - GVIM". The menu bar includes "파일(F)", "편집(E)", "도구(I)", "문법(S)", "버퍼(B)", "창(W)", and "도움말(H)". The toolbar below the menu contains various icons for file operations. The main editor window displays the following text:

```
1 !"#$%&'(          )*+,,-./
2 0123456789:;<=>?
3 @ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
4 `abcdefghijklmnoprstuvwxyz z{|}~
5
~
```

The status bar at the bottom indicates the search term `/\vz(\{)@=`, the current line number `4,28`, and the search mode `모 두`.

# 정규표현식



## ▶ VIM 정규표현식

- 부정형 전방탐색 구문 **패턴1(패턴2)@!**
- 패턴2로 끝나지 않는 패턴1을 검사

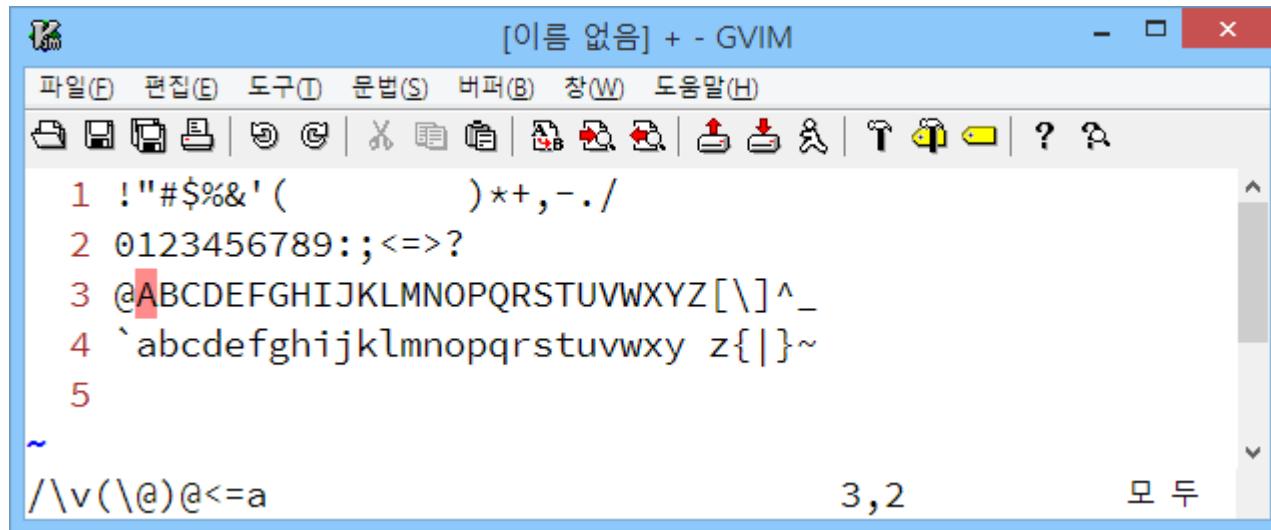
```
[이름 없음] + - GVIM
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
1 !"#$%&' ( )*+,.-./
2 0123456789:;=>?
3 @ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^_
4 `abcdefghijklmnoprstuvwxyz z{|}~
5
~
/\vz(\{\}@!
```

3,27      모 두

# 정규표현식

## ▶ VIM 정규표현식

- 긍정형 후방탐색 구문 **(패턴1)@<=패턴2**
- 패턴1로 시작하는 패턴2를 검사



The screenshot shows a GVIM window with the title '[이름 없음] + - GVIM'. The menu bar includes '파일(F)', '편집(E)', '도구(I)', '문법(S)', '버퍼(B)', '창(W)', and '도움말(H)'. The toolbar below the menu has various icons. The main text area contains five numbered lines of text:

- 1 !"#%&'( )\*+,-. /
- 2 0123456789:;<=>?
- 3 @ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^\_
- 4 `abcdefghijklmnoprstuvwxyz z{|}~
- 5

The status bar at the bottom shows the search command `/\v(\@\@<=a`, the current position `3,2`, and the selection mode `모 두`.

# 정규표현식



## ▶ VIM 정규표현식

- 부정형 후방탐색 구문 **(패턴1)@<!패턴2**
- 패턴1로 시작하지 않는 패턴2를 검사

The screenshot shows a GVIM window with the title "[이름 없음] + - GVIM". The menu bar includes 파일(F), 편집(E), 도구(I), 문법(S), 버퍼(B), 창(W), and 도움말(H). The toolbar below the menu has various icons for file operations. The main text area displays the following content:

```
1 !"#$%&'(          )*+,,-./
2 0123456789:;=>?
3 @ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
4 `abcdefghijklmnoprstuvwxyz z{|}~
5
~
```

The status bar at the bottom shows the search command `/\\v(\\@)@<!a`, the current line number `4,2`, and the selection mode `모 두`.

# 정규표현식



## ▶ VIM 정규표현식

- 수량을 결정하는 메타문자, 수량자(Quantifier)
- 최대 검사 모드

일반모드	매직모드	설명	비고
*		0개 이상	
\+	+	1개 이상	
\?, \=	? , =	0 또는 1개	고정된 검사 수량 (수량을 모를 때 적합)
\{n\}	{n}	n개	
\{n,}	{n,}	n개 이상	가변적 검사 수량 (수량을 알 때 적합)
\{m,n\}	{m,n}	m개 이상, n개 이하	

- 최소 검사 모드

일반모드	매직모드	설명	PCRE
\{-}	{-}, {-0,}	최소 0개	*?
\{-1,}	{-1,}	최소 1개	+?
\{n,}	{-n,}	최소 n개	{n,}?

# 정규표현식



## ▶ VIM 정규표현식

- 동작 방식을 수정하는 메타문자, 수정자(Modifier)

VIM	설명
:set ignore case	대소문자 구분 해제
multiline, global 옵션 기본 사용	일치하는 모든 문자열 검사(global) 앵커(Anchors) 문자 '^', '\$'의 줄 구분 해제(multi line)
\_(single line)	메타 문자의 검사 범위에 줄바꿈 문자 포함

# 정규표현식



## ▶▶ Why VIM?

- 강력한 성능

The screenshot shows a Windows desktop environment. In the foreground, there is a GVIM window titled "secure.log (D:\) - GVIM1". The window displays a log file with several entries. The entries are highlighted with red boxes around the timestamp and process ID, demonstrating Vim's search and highlighting capabilities. The log entries include:

- 217569 Dec 31 06:55:47 Sensor sshd[25218]: Accepted password for ykm^ kelho from ::ffff:192.168.182.99 port 10455 ssh2
- 217570 Dec 31 06:56:05 Sensor sshd[25220]: Received disconnect from ::ffff:192.168.182.99: 11: Disconnect requested by Windows SS H Client.
- 217571 Dec 31 14:43:30 Sensor sshd[25643]: Accepted password for nkm kanhg from ::ffff:192.168.7.225 port 33005 ssh2
- 217572 Dec 31 14:43:47 Sensor sshd[25647]: Accepted password for nkm kanhg from ::ffff:192.168.7.225 port 4724 ssh2
- 217573 Dec 31 14:43:50 Sensor sshd[25649]: Received disconnect from ::ffff:192.168.7.225: 11:
- 217574 Dec 31 14:44:55 Sensor sshd[25653]: Accepted password for nkm kanhg from ::ffff:192.168.7.225 port 46250 ssh2

In the background, there is another window titled "regular expressions 101" which is an online regex tester. It has a search bar containing the regular expression "\d+", a "TEST STRING" input field with the log file content, and a "SUBSTITUTION" input field.

# 정규표현식



## ▶▶ Why VIM?

- 데이터 구조 파악 및 변환을 지원하는 다양한 기능

The screenshot shows two windows side-by-side. On the left is GVIM displaying a log file named 'secure.log'. The log contains several entries, such as:

```
217559 Dec 30 07:41:27 Sensor sshd[23386]: Accepted password for vkmk^
217560 Dec 30 07:42:16 Sensor sshd[23388]: Rec...
217561 Dec 30 08:35:43 Sensor sshd[23439]: Rec...
217562 Dec 30 08:37:56 Sensor sshd[23441]: Rec...
217563 Dec 30 08:38:04 Sensor sshd[23443]: Rec...
217564 Dec 30 08:39:55 Sensor sshd[23445]: Rec...
217565 Dec 30 08:40:08 Sensor sshd[23452]: Acc...
217566 Dec 30 08:43:09 Sensor sshd[23454]: Rec...
217567 Dec 31 06:41:40 Sensor sshd[25183]: Acc...
217568 Dec 31 06:42:44 Sensor sshd[25185]: Rec...
217569 Dec 31 06:55:47 Sensor sshd[25218]: Acc...
217570 Dec 31 06:56:05 Sensor sshd[25220]: Rec...
217571 Dec 31 14:43:30 Sensor sshd[25643]: Acc...
```

A large red arrow points from the bottom of the GVIM window towards the Microsoft Excel window on the right. The Excel window displays a CSV file with the following data:

	Date	Time	Host	Proc	Keyword	Msg
1						
2	02-Jan	8:06:36	Sensor	sshd	Accepted password	Accepted password for islee from ::ffff:192...
3	02-Jan	9:08:29	Sensor	sshd	Received disconnect	Received disconnect from ::ffff:192.168.19...
4	02-Jan	9:12:28	Sensor	sshd	Accepted password	Accepted password for islee from ::ffff:192...
5	02-Jan	9:14:29	Sensor	sshd	Received disconnect	Received disconnect from ::ffff:192.168.19...
6	02-Jan	18:47:35	Sensor	sshd	Accepted password	Accepted password for islee from ::ffff:192...
7	02-Jan	18:47:53	Sensor	sshd	Received disconnect	Received disconnect from ::ffff:192.168.19...
8	03-Jan	11:09:41	Sensor	sshd	Accepted password	Accepted password for kye from ::ffff:192...
9	03-Jan	11:29:58	Sensor	sshd	Received disconnect	Received disconnect from ::ffff:192.168.19...
10	03-Jan	16:01:38	Sensor	sshd	Accepted password	Accepted password for byun from ::ffff:19...
11	03-Jan	16:38:05	Sensor	sshd	Received disconnect	Received disconnect from ::ffff:192.168.28...
12	04-Jan	8:28:44	Sensor	sshd	Accepted password	Accepted password for kye from ::ffff:192...
13	04-Jan	8:48:00	Sensor	sshd	Received disconnect	Received disconnect from ::ffff:192.168.27...
14	04-Jan	9:14:57	Sensor	sshd	Accepted password	Accepted password for kye from ::ffff:192...

# 정규표현식

## ▶ VIM 심화

- 치환 명령어 작업 순서
  - ① 치환을 원하는 문자열 검사
  - ② 검사 성공
  - ③ 치환 명령어 작성 및 실행

:%s/**치환 전 문자열**/**치환 후 문자열**/

The image shows two GVIM windows side-by-side. The left window is titled 'secure.log (D:\)' and the right window is titled 'secure.log + (D:\)'. Both windows display log entries from a file. The left window has a blue selection box around its status bar, which contains the command ':%s/\v\\[\d+]/'. The right window also has a blue selection box around its status bar, which contains the result of the command: ':%s/\v\\[\d+]///'. Red boxes highlight the search pattern '\v\\[\d+' in the left status bar and the replaced text '\\[\d+' in the right status bar.

```
secure.log (D:\) - GVIM1
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
217569 Dec 31 06:55:47 Sensor sshd[25218]: kelho from ::ffff:192.168.182.99 port 55555 ssh2
217570 Dec 31 06:56:05 Sensor sshd[25220]: ::ffff:192.168.182.99: 11: Disconnected H Client.
217571 Dec 31 14:43:30 Sensor sshd[25643]: kanhg from ::ffff:192.168.7.225 port 46250 ssh2
217572 Dec 31 14:43:47 Sensor sshd[25647]: kanhg from ::ffff:192.168.7.225 port 46250 ssh2
217573 Dec 31 14:43:50 Sensor sshd[25649]: ::ffff:192.168.7.225: 11:
217574 Dec 31 14:44:55 Sensor sshd: Accepted password for nkmkanhg from ::ffff:192.168.7.225 port 4724 ssh2
217575 Dec 31 14:44:58 Sensor sshd: Received disconnect from ::ffff:192.168.7.225: 11:
217576 Dec 31 20:02:57 Sensor sshd: Accepted password for ykmkelho from ::ffff:192.168.76.175 port 59281 ssh2
217577 Dec 31 20:03:19 Sensor sshd: Received disconnect from ::ffff:192.168.76.175: 11: Disconnect requested by Windows SSH Client.

secure.log + (D:\) - GVIM1
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
217572 Dec 31 14:43:47 Sensor sshd: Accepted password for nkmkanhg from ::ffff:192.168.7.225 port 4724 ssh2
217573 Dec 31 14:43:50 Sensor sshd: Received disconnect from ::ffff:192.168.7.225: 11:
217574 Dec 31 14:44:55 Sensor sshd: Accepted password for nkmkanhg from ::ffff:192.168.7.225 port 46250 ssh2
217575 Dec 31 14:44:58 Sensor sshd: Received disconnect from ::ffff:192.168.7.225: 11:
217576 Dec 31 20:02:57 Sensor sshd: Accepted password for ykmkelho from ::ffff:192.168.76.175 port 59281 ssh2
217577 Dec 31 20:03:19 Sensor sshd: Received disconnect from ::ffff:192.168.76.175: 11: Disconnect requested by Windows SSH Client.

~ ~
:%s/\v\\[\d+]///
```

# 정규표현식

## ▶ VIM 심화

- VIM은 마지막 검색어를 기억한다

```
secure.log + (D:W) - GVIM1
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
217571 Dec 31 14:43:30 Sensor sshd: Accepted password for nkmkanhg from ::ffff:192.168.7.225 port 33005 ssh2
217572 Dec 31 14:43:47 Sensor sshd: Accepted password for nkmkanhg from ::ffff:192.168.7.225 port 4724 ssh2
217573 Dec 31 14:43:50 Sensor sshd: Received disconnect from ::ffff:192.168.7.225: 11:
217574 Dec 31 14:44:55 Sensor sshd: Accepted password for nkmkanhg from ::ffff:192.168.7.225 port 46250 ssh2
217575 Dec 31 14:44:58 Sensor sshd: Received disconnect from ::ffff:192.168.7.225: 11:
217576 Dec 31 20:02:57 Sensor sshd: Accepted password for ykmkelho from ::ffff:192.168.76.175 port 59281 ssh2
217577 Dec 31 20:03:19 Sensor sshd: Received disconnect from ::ffff:192.168.76.175: 11: Disconnect requested by Windows SSH Client.
```

직전 사용했던 검색어  
자동 반영

삭제 목적으로  
비워둔 공란

:%s/공란/공란/

# 정규표현식



## ▶ VIM 심화

- VIM은 마지막 검색어를 기억한다

```
secure.log (D:\) - GVIM1
217572 Dec 31 14:43:47 Sensor sshd[25647]: Accepted password for nkmkan^
      hg from ::ffff:192.168.7.225 port 4724 ssh2
217573 Dec 31 14:43:50 Sensor sshd[25649]: Received disconnect from ::f
      fff:192.168.7.22
217574 Dec 31 14:44:55
      hg from ::ffff:1
217575 Dec 31 14:44:58
      fff:192.168.7.22
217576 Dec 31 20:02:57
      ho from ::ffff:1
217577 Dec 31 20:03:19
      fff:192.168.76.1
      nt.

/gssh

secure.log + (D:\) - GVIM1
174 Dec 16 16:20:04 Sensor login: pam_unix(remote:session): session clo^
sed for user user07
175 Dec 16 16:20:35 Sensor login: pam_unix(remote:auth): authentication
failure; logname=LOGIN uid=0 euid=0 tty=/dev/pts/2 ruser= rhost=
user=user11
176 Dec 16 16:20:43 Sensor login: pam_unix(remote:auth): check pass; us
er unknown
177 Dec 16 16:20:43 Sensor login: pam_unix(remote:auth): authentication
failure; logname=LOGIN uid=0 euid=0 tty=/dev/pts/2 ruser= rhost=
178 Dec 16 16:20:43 Sensor login: pam_succeed_if(remote:auth): error re
trieving information about user itl
179 Dec 16 16:21:11 Sensor login: pam_unix(remote:session): session ope
ned for user user07 by LOGIN(uid=0)

:g//d
```

# 정규표현식



## ▶ VIM 심화

- 원본을 유지하면서 구분자를 추가하고 싶다면?

```
secure.log (D:\) - GVIM1
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
217572 Dec 31 14:43:47 Sensor sshd[25647]: Accepted password for nkmkan^
hg from ::ffff:192.168.7.225 port 4724 ssh2
217573 Dec 31 14:43:50 Sensor s
fff:192.168.7.225: 11:
217574 Dec 31 14:44:55 Sensor s
hg from ::ffff:192.168.7
217575 Dec 31 14:44:58 Sensor s
fff:192.168.7.225: 11:
217576 Dec 31 20:02:57 Sensor s
ho from ::ffff:192.168.7
217577 Dec 31 20:03:19 Sensor s
fff:192.168.76.175: 11:
nt.

~
```

```
/\v^.{15}
```

```
secure.log + (D:\) - GVIM1
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
217572 Dec 31 14:43:47 Sensor sshd[25647]: Accepted password for nkmk^
anhg from ::ffff:192.168.7.225 port 4724 ssh2
217573 Dec 31 14:43:50 Sensor sshd[25649]: Received disconnect from :
:ffff:192.168.7.225: 11:
217574 Dec 31 14:44:55 Sensor sshd[25653]: Accepted password for nkmk
anhg from ::ffff:192.168.7.225 port 46250 ssh2
217575 Dec 31 14:44:58 Sensor sshd[25655]: Received disconnect from :
:ffff:192.168.7.225: 11:
217576 Dec 31 20:02:57 Sensor sshd[25951]: Accepted password for ykmk
elho from ::ffff:192.168.76.175 port 59281 ssh2
217577 Dec 31 20:03:19 Sensor sshd[25953]: Received disconnect from :
:ffff:192.168.76.175: 11: Disconnect requested by Windows SSH cl
ient.

~
```

```
217577,16 바 닥
```

# 정규표현식

## ▶ VIM 심화

- 원본 유지 실패!

The image shows two GVIM windows side-by-side. The left window is titled "secure.log (D:\)" and the right window is titled "secure.log + (D:\)". Both windows display log entries from a sensor's SSHD service. A red U-shaped arrow points from the bottom of the left window towards the top of the right window, indicating a search operation.

Left Window Content:

```
217572 Dec 31 14:43:47 Sensor sshd[25647]: Accepted password for nkmkan^  
hg from ::ffff:192.168.7.225 port 4724 ssh2  
217573 Dec 31 14:43:50 Sensor sshd[28969]: Received disconnect from ::ffff:  
192.168.7.225: 11: Disconnect requested by Windows SSH Client.  
217574 Dec 31 14:44:55 Sensor sshd[28971]: Accepted password for ikmslehe fro^  
m ::ffff:192.168.191.174 port 60589 ssh2  
217575 Dec 31 14:44:58 Sensor sshd[29032]: Received disconnect from ::ffff:  
192.168.7.225: 11: Disconnect requested by Windows SSH Client.  
217576 Dec 31 20:02:57 Sensor sshd[29034]: Accepted password for ikmslehe fro^  
m ::ffff:192.168.191.174 port 56083 ssh2  
217577 Dec 31 20:03:19 Sensor sshd[29547]: Received disconnect from ::ffff:  
192.168.76.175: 11: Disconnect requested by Windows SSH Client.  
217578 Dec 31 20:03:19 Sensor sshd[29549]: Accepted password for ikmslehe fro^  
m ::ffff:192.168.191.174 port 60691 ssh2  
217579 Dec 31 20:03:19 Sensor sshd[1258]: Received disconnect from ::ffff:  
192.168.182.99 port 1207 ssh2  
217580 Dec 31 20:03:19 Sensor sshd[1258]: Accepted password for ykmkelho from  
::ffff:192.168.182.99 port 1207 ssh2
```

Right Window Content:

```
1 v^.{15}= Sensor sshd[28969]: Accepted password for ikmslehe fro^  
m ::ffff:192.168.191.174 port 60589 ssh2  
2 v^.{15}= Sensor sshd[28971]: Received disconnect from ::ffff:  
192.168.191.174: 11: Disconnect requested by Windows SSH Client.  
3 v^.{15}= Sensor sshd[29032]: Accepted password for ikmslehe fro^  
m ::ffff:192.168.191.174 port 56083 ssh2  
4 v^.{15}= Sensor sshd[29034]: Received disconnect from ::ffff:  
192.168.191.174: 11: Disconnect requested by Windows SSH Client.  
5 v^.{15}= Sensor sshd[29547]: Accepted password for ikmslehe fro^  
m ::ffff:192.168.191.174 port 60691 ssh2  
6 v^.{15}= Sensor sshd[29549]: Received disconnect from ::ffff:  
192.168.191.174: 11: Disconnect requested by Windows SSH Client.  
7 v^.{15}= Sensor sshd[1258]: Accepted password for ykmkelho from  
::ffff:192.168.182.99 port 1207 ssh2
```

Bottom Command Line:

```
:%s//\v^.{15}= /
```

# 정규표현식

## ▶ VIM 심화

- VIM은 마지막 검사 결과를 저장할 수 있다

secure.log (D:\) - GVIM1

파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)

217572 Dec 31 14:43:47 Sensor sshd[25647]: Accepted password for nkmk^  
ahg from ::ffff:192.168.7.225 port 4724 ssh2  
217573 Dec 31 14:43:50 Sensor sshd[25649]: Received disconnect from ::ffff:192.168.7.225: 11:  
217574 Dec 31 14:44:55 Sensor sshd[25653]: Accepted password for nkmk^  
ahg from ::ffff:192.168.7.225 port 46250 ssh2  
217575 Dec 31 14:44:58 Sensor sshd[25655]: Received disconnect from ::ffff:192.168.7.225: 11:  
217576 Dec 31 20:02:57 Sensor sshd[25951]: Accepted password for ykmk^  
elho from ::ffff:192.168.76.175 port 59281 ssh2  
217577 Dec 31 20:03:19 Sensor sshd[25953]: Received disconnect from ::ffff:192.168.76.175: 11: Disconnect requested by client.

secure.log + (D:\) - GVIM1

파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)

:%s//&= /

직전 검사 결과를 저장해주는 메타문자

:%s//&=/

# 룰 최적화 #1



## 를 최적화



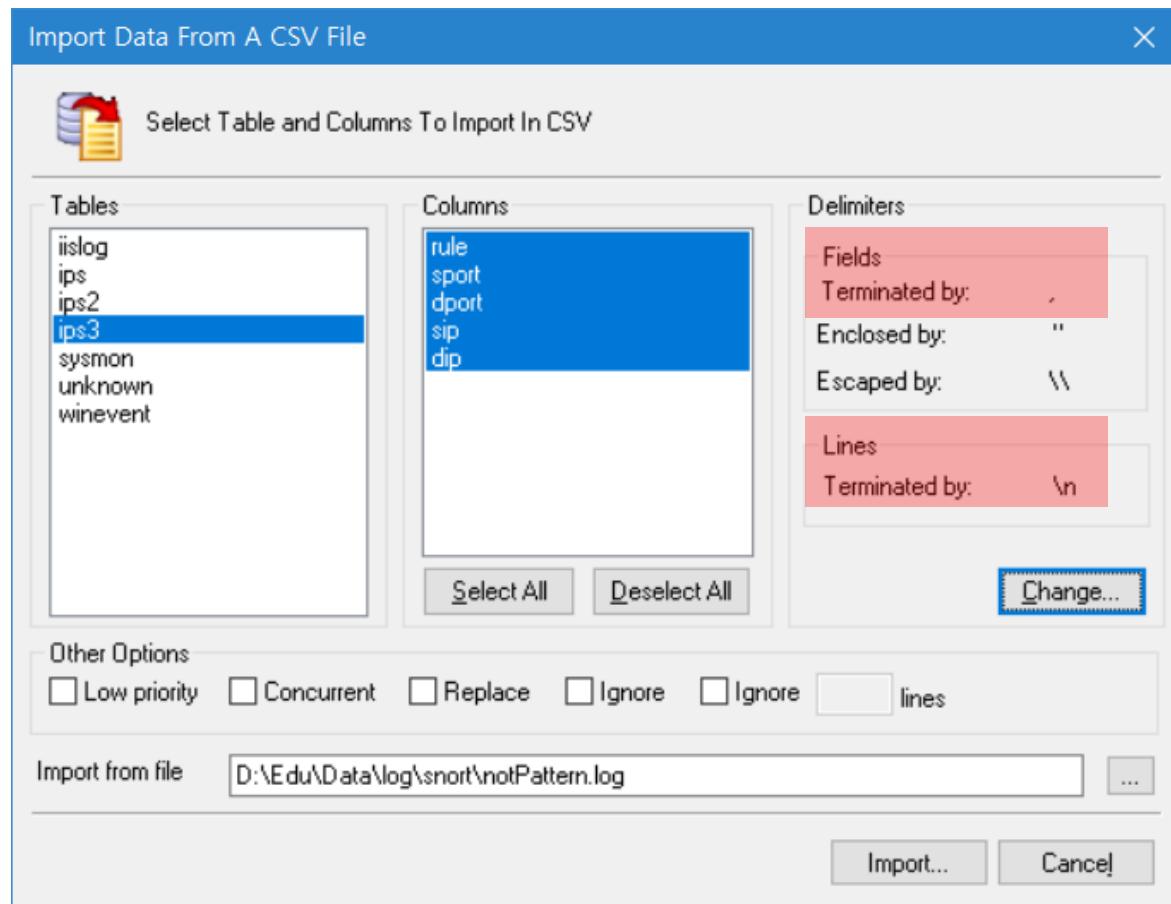
## ► Create Table

- ## ▪ 필드 구조

# 룰 최적화

## ▶ Import Data From CSV

- 레코드/필드 구분



# 룰 최적화

## ▶ 계산 함수

- count: 개수를 센다

The screenshot shows the SQLyog Community Edition interface. The left sidebar displays the database schema with databases like information\_schema, mysql, performance\_schema, snort, sys, and test. The main area has a 'Query' tab containing the following SQL code:

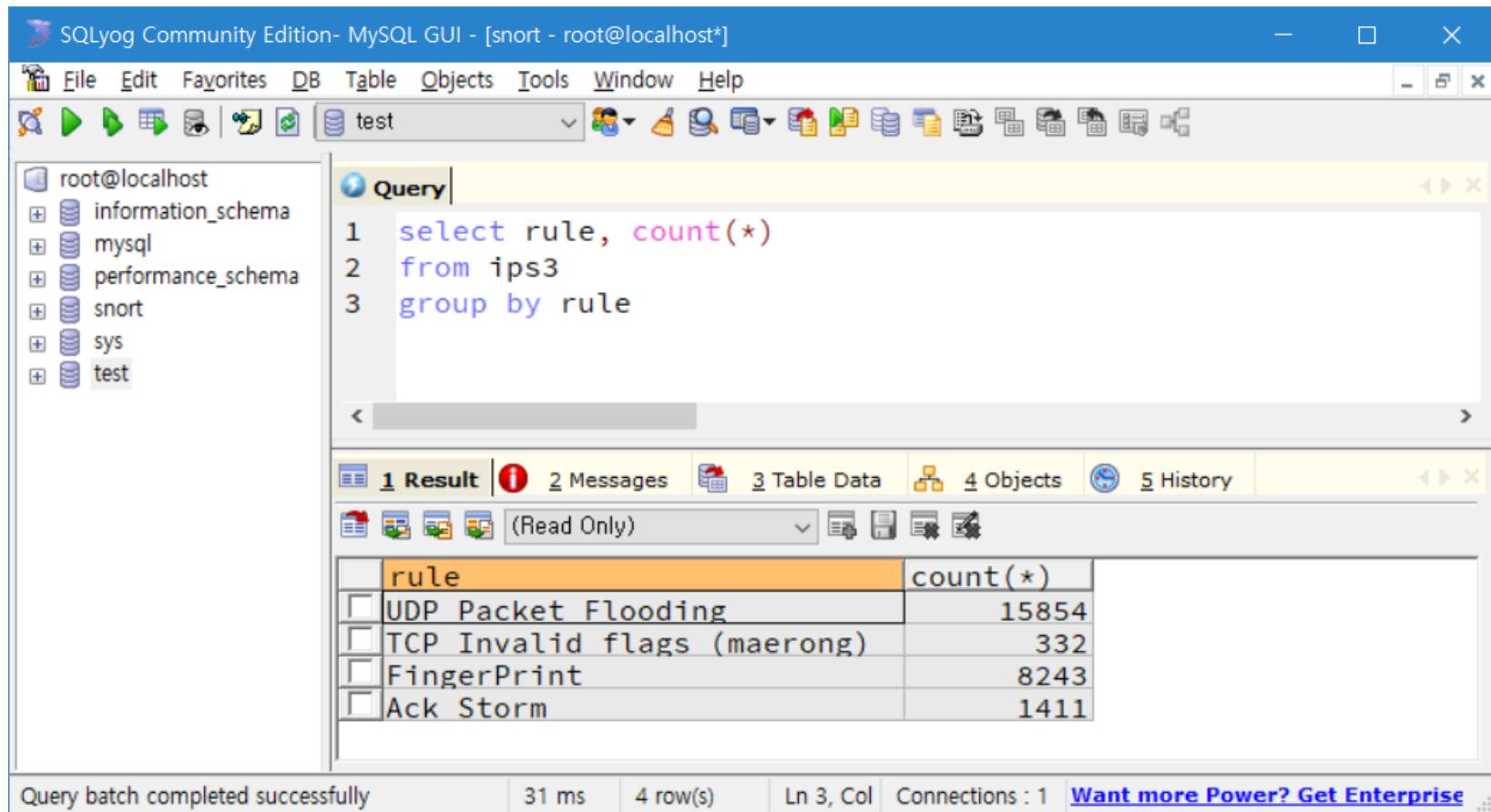
```
1 select count(*)
2 from ips3
```

The results pane below shows a single row with the column 'count(\*)' containing the value '25840'. The status bar at the bottom indicates 'Query batch completed successfully'.

# 룰 최적화

## ▶ 계산 함수

- group by: 필드가 같으면 계산 결과를 합친다



The screenshot shows the SQLyog Community Edition MySQL GUI interface. The left sidebar displays database schemas: root@localhost, information\_schema, mysql, performance\_schema, snort, sys, and test. The main window has a toolbar at the top with various icons. Below the toolbar is a dropdown menu showing 'test'. The central area contains a 'Query' editor with the following SQL code:

```
1 select rule, count(*)
2 from ips3
3 group by rule
```

Below the query editor is a results pane titled 'Result' with tabs for 'Messages', 'Table Data', 'Objects', and 'History'. The 'Table Data' tab is selected, showing a table with two columns: 'rule' and 'count(\*)'. The data is as follows:

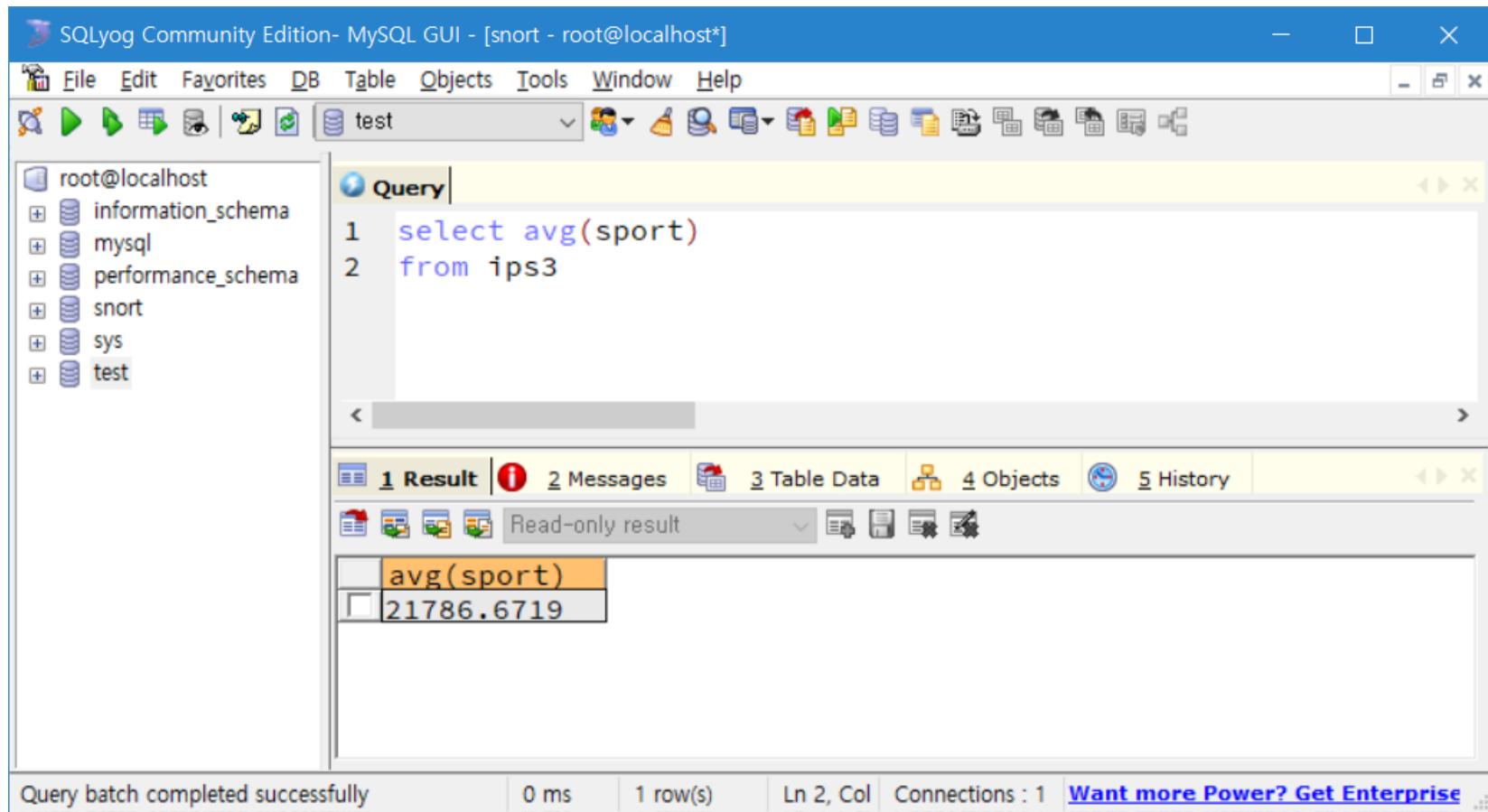
rule	count(*)
UDP Packet Flooding	15854
TCP Invalid flags (maerong)	332
FingerPrint	8243
Ack Storm	1411

At the bottom of the results pane, there are buttons for 'Read Only', 'Edit', 'Insert', 'Delete', and 'Update'. The status bar at the bottom of the application window shows 'Query batch completed successfully', '31 ms', '4 row(s)', 'Ln 3, Col', 'Connections : 1', and a link 'Want more Power? Get Enterprise'.

# 룰 최적화

## ▶ 계산 함수

- avg: 평균



The screenshot shows the SQLyog Community Edition interface. The left sidebar displays database schemas: root@localhost, information\_schema, mysql, performance\_schema, snort, sys, and test. The current schema is set to 'test'. The main area contains a 'Query' editor with the following SQL code:

```
1 select avg(sport)
2 from ips3
```

Below the query editor, the results pane shows one row of data:

avg(sport)
21786.6719

At the bottom of the interface, a status bar indicates: 'Query batch completed successfully'.

# 룰 최적화

## ▶ 계산 함수

- `distinct`: 중복 제거

The screenshot shows the SQLyog Community Edition MySQL GUI interface. The left sidebar displays the database structure with databases like information\_schema, mysql, performance\_schema, snort, sys, and test. The main window has a toolbar at the top with various icons. The central area is divided into sections: 'Query' (containing the SQL code), 'Result' (showing the execution status), 'Messages' (empty), 'Table Data' (disabled), 'Objects' (disabled), and 'History' (disabled). The 'Result' tab is selected, showing a table with one column named 'sip'. The data in the table is:

sip
7.7.7.17
7.7.7.21
8.8.8.8
7.7.7.2
168.126.63.1

At the bottom, the status bar indicates "Query batch completed successfully" and "Want more Power? Get Enterprise".

# 룰 최적화

## ▶ 계산 함수

- `distinct`: 중복 제거

The screenshot shows the SQLyog Community Edition interface. The left sidebar displays the database structure with databases like information\_schema, mysql, performance\_schema, snort, sys, and test. The main window has a toolbar at the top with various icons. The central area is divided into sections: 'Query' (containing the SQL code), 'Result' (showing the execution output), and 'Messages'. The 'Query' section contains the following SQL:

```
1 select count(distinct sip)
2 from ips3
```

The 'Result' section shows the output of the query:

count(distinct sip)
6578

At the bottom, a status bar indicates "Query batch completed successfully" and "Want more Power? Get Enterprise".

# 룰 최적화

## ▶ 계산 함수

- `distinct`: 중복 제거

The screenshot shows the SQLyog Community Edition interface. The left sidebar displays database schemas: root@localhost, information\_schema, mysql, performance\_schema, snort, sys, and test. The main area has a toolbar at the top with various icons. Below the toolbar is a 'Query' editor containing the following SQL code:

```
1 select rule, count(distinct sip)
2 from ips3
3 group by rule
```

Below the query editor is a results pane with tabs for 'Result', 'Messages', 'Table Data', 'Objects', and 'History'. The 'Result' tab is selected, showing a table with four rows. The table has two columns: 'rule' and 'count(distinct sip)'. The data is as follows:

rule	count(distinct sip)
Ack Storm	59
FingerPrint	6346
TCP Invalid flags (maerong)	164
UDP Packet Flooding	11

At the bottom of the results pane, it says 'Query batch completed successfully'.

# 룰 최적화

## ▶ 발생량 기반 룰

- 출발/목적지간 발생 현황 분석

The screenshot shows the SQLyog Community Edition MySQL GUI interface. The left sidebar displays the database schema with databases like root@localhost, information\_schema, mysql, performance\_schema, snort, sys, and test. The main window has a 'Query' tab containing the following SQL code:

```
1 select sip, dip, sport, dport
2 from ips3
3 where rule = 'udp packet flooding'
```

The results tab shows a table with the following data:

sip	dip	sport	dport
7.7.7.17	172.17.30.240	161	53429
7.7.7.17	172.17.30.240	161	53429
7.7.7.17	172.17.30.240	161	53429
7.7.7.17	172.17.30.240	161	53429
7.7.7.17	172.17.30.240	161	53429
7.7.7.17	172.17.30.240	161	53429
7.7.7.17	172.17.30.240	161	53429

At the bottom, the status bar shows the URL <http://www.webyog.com>, execution time 0 ms, 15854 row(s), Ln 3, Col, Connections : 2, and a link to 'Want more Power? Get Enterprise'.

# 룰 최적화

## ▶ 발생량 기반 룰

- 출발/목적지간 발생 현황 분석

The screenshot shows the SQLyog Community Edition MySQL GUI interface. The database connection is set to [snort - root@localhost\*]. The left sidebar displays the database schema with databases like information\_schema, mysql, performance\_schema, snort, sys, and test. The main window has a 'Query' tab where the following SQL code is entered:

```
1 select sip, dip, sport, dport, count(*)
2 from ips3
3 where rule = 'udp packet flooding'
4 group by sip, dip, sport, dport
```

The results are displayed in a 'Result' tab, showing a table with columns: sip, dip, sport, dport, and count(\*). The data is as follows:

sip	dip	sport	dport	count(*)
7.7.7.17	172.17.30.240	161	53429	17
7.7.7.21	172.17.30.240	48182	161	1
7.7.7.17	172.17.30.240	161	40398	48
7.7.7.21	172.17.30.240	44836	161	1
7.7.7.17	172.17.30.240	161	47901	335
7.7.7.21	172.17.30.240	40409	161	1
7.7.7.21	172.17.30.240	50053	161	1

At the bottom, the status bar indicates "Query batch completed successfully" and "32 ms 658 row(s) Ln 4, Col". There is also a link "Want more Power? Get Enterprise".

# 룰 최적화

## ▶ 발생량 기반 룰

- 출발/목적지간 발생 현황 분석

The screenshot shows the SQLyog Community Edition interface. The left sidebar displays the database structure with databases like information\_schema, mysql, performance\_schema, snort, sys, and test. The main window has a 'Query' tab where a MySQL query is written:

```
1 select sip, dip, case
2           when sport >= 1024
3               then 'gte1024'
4           else sport
5       end as srcport, case
6           when dport >= 1024
7               then 'gte1024'
8           else dport
9       end as dstport, count(*) 
10  from ips3
11  where rule = 'udp packet flooding'
12  group by sip, dip, srcport, dstport
```

Below the query, the 'Result' tab shows the execution output:

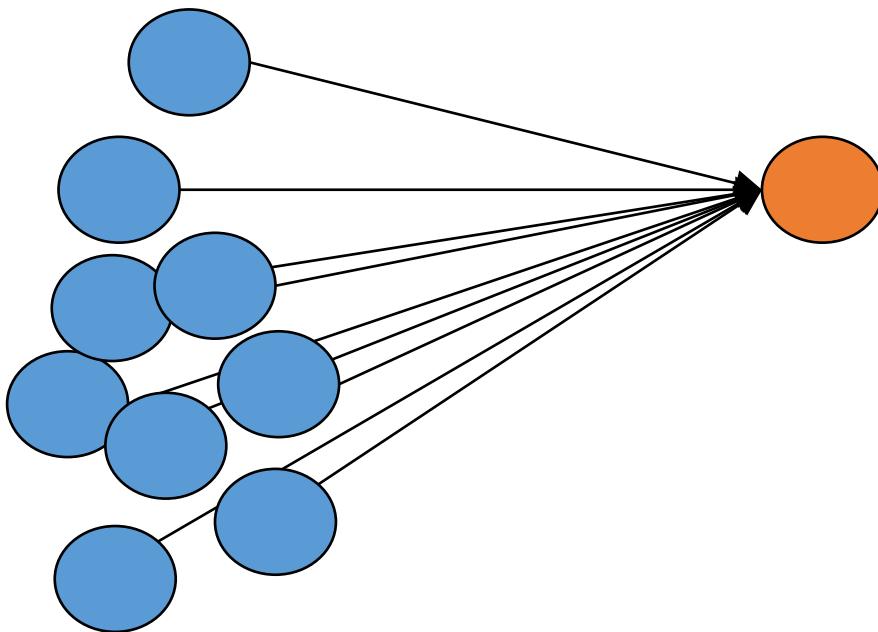
sip	dip	srcport	dstport	count(*)
10.0.0.5	172.17.20.178	gte1024	53	7
168.126.63.1	172.17.20.178	53	gte1024	15
200.200.2.2	172.17.20.177	gte1024	53	2
200.200.2.4	172.17.20.177	gte1024	53	2
200.200.2.5	172.17.20.177	gte1024	53	1
200.200.2.7	172.17.20.177	gte1024	53	6

At the bottom, it says "Query batch completed successfully" and "Want more Power? Get Enterprise".

# 룰 최적화

## ▶ 발생량 기반 룰

- 서버/클라이언트 구분



# 룰 최적화

## ▶ 발생량 기반 룰

- 출발/목적지간 발생 현황 분석

The screenshot shows the SQLyog Community Edition interface. The left sidebar displays database schemas: root@localhost, information\_schema, mysql, performance\_schema, snort, sys, and test. The main window has a 'Query' tab open with the following SQL code:

```
4           else sport
5           end as srcport, case
6                           when dport >= 1024
7                               then 'gte1024'
8                           else dport
9           end as dstport, count(*)
10      from ips3
11      where rule = 'fingerprint'
12      group by sip, dip, srcport, dstport
```

The 'Result' tab below shows the query results:

sip	dip	srcport	dstport	count(*)
118.235.26.15	172.17.10.70	get1024	443	
118.32.50.166	172.17.10.70	get1024	443	
1.218.215.49	172.17.10.70	get1024	443	

At the bottom, a message indicates the query completed successfully.

# 룰 최적화

## ▶ 발생량 기반 룰

- 출발/목적지간 발생 현황 분석

The screenshot shows the SQLyog Community Edition interface. The left sidebar displays a database tree with databases like information\_schema, mysql, performance\_schema, snort, sys, and test. The main area has a 'Query' tab containing the following SQL code:

```
1 select sip, count(distinct dip)
2 from ips3
3 where rule = 'fingerprint'
4 group by sip
```

The results are displayed in a 'Result' tab, showing the following data:

sip	count(distinct dip)
1.177.147.152	1
1.208.110.151	1
1.209.231.130	1
1.209.9.20	1
1.210.120.4	1
1.210.123.55	1

At the bottom, a message says "Query batch completed successfully" and shows statistics: 32 ms, 6346 row(s), Ln 4, Col 1, Connections : 1.

# 룰 최적화

## ▶ 발생량 기반 룰

- 출발/목적지간 발생 현황 분석



SQLyog Community Edition- MySQL GUI - [snort - root@localhost\*]

File Edit Favorites DB Table Objects Tools Window Help

test

Query

```
1 select dip, count(distinct sip)
2 from ips3
3 where rule = 'fingerprint'
4 group by dip
```

Result | Messages | Table Data | Objects | History

(Read Only)

dip	count(distinct sip)
172.17.10.130	77
172.17.10.134	1
172.17.10.135	21
172.17.10.136	2
172.17.10.142	2
172.17.10.50	144

Query batch completed successfully | 47 ms | 10 row(s) | Ln 4, Col | Connections : 1 | Want more Power? Get Enterprise

# 룰 최적화

## ▶ 발생량 기반 룰

- 출발/목적지간 발생 현황 분석

The screenshot shows the SQLyog Community Edition MySQL GUI interface. The title bar reads "SQLyog Community Edition- MySQL GUI - [snort - root@localhost\*]". The menu bar includes File, Edit, Favorites, DB, Table, Objects, Tools, Window, and Help. The toolbar contains various icons for database management. The left sidebar shows a tree view of databases: root@localhost, information\_schema, mysql, performance\_schema, snort, sys, and test. The main area has two tabs: "Query" and "Result". The "Query" tab contains the following SQL code:

```
1 select sport, count(distinct dport)
2 from ips3
3 where rule = 'fingerprint'
4 group by sport
```

The "Result" tab displays the output of the query in a table format:

sport	count(distinct dport)
1028	1
1029	1
1037	1
1055	1
1056	1
1092	2

At the bottom, a message says "Query batch completed successfully" and provides performance metrics: 15 ms, 6743 row(s), Ln 4, Col, Connections : 1. A blue link at the bottom right says "Want more Power? Get Enterprise".

# 룰 최적화

## ▶ 발생량 기반 룰

- 출발/목적지간 발생 현황 분석

The screenshot shows the SQLyog Community Edition interface. The left sidebar displays database schemas: root@localhost, information\_schema, mysql, performance\_schema, snort, sys, and test. The current database is set to 'test'. The main window has a 'Query' tab where the following SQL code is entered:

```
1 select dport, count(distinct sport)
2 from ips3
3 where rule = 'fingerprint'
4 group by dport
```

Below the query, there are tabs for 'Result', 'Messages', 'Table Data', 'Objects', and 'History'. The 'Result' tab is selected, showing a table with the following data:

dport	count(distinct sport)
80	22
443	6628
8080	7
8180	18
9443	1
14001	5

At the bottom of the interface, a message states 'Query batch completed successfully'.

# 룰 최적화

## ▶ 발생량 기반 룰

- 출발/목적지간 발생 현황 분석



SQLyog Community Edition- MySQL GUI - [snort - root@localhost\*]

File Edit Favorites DB Table Objects Tools Window Help

test

Query

```
1 select count(distinct sip), count(distinct sport),
2 count(distinct dip), count(distinct dport)
3 from ips3
4 where rule = 'fingerprint'
```

1 Result 2 Messages 3 Table Data 4 Objects 5 History

Read-only result

count(distinct sip)	count(distinct sport)	count(distinct dip)	count(distinct dport)
6346	6743	10	8

Query batch completed successfully | 31 ms | 1 row(s) | Ln 4, Col 26 | Connections : 1 | Want more Power? Get Enterprise

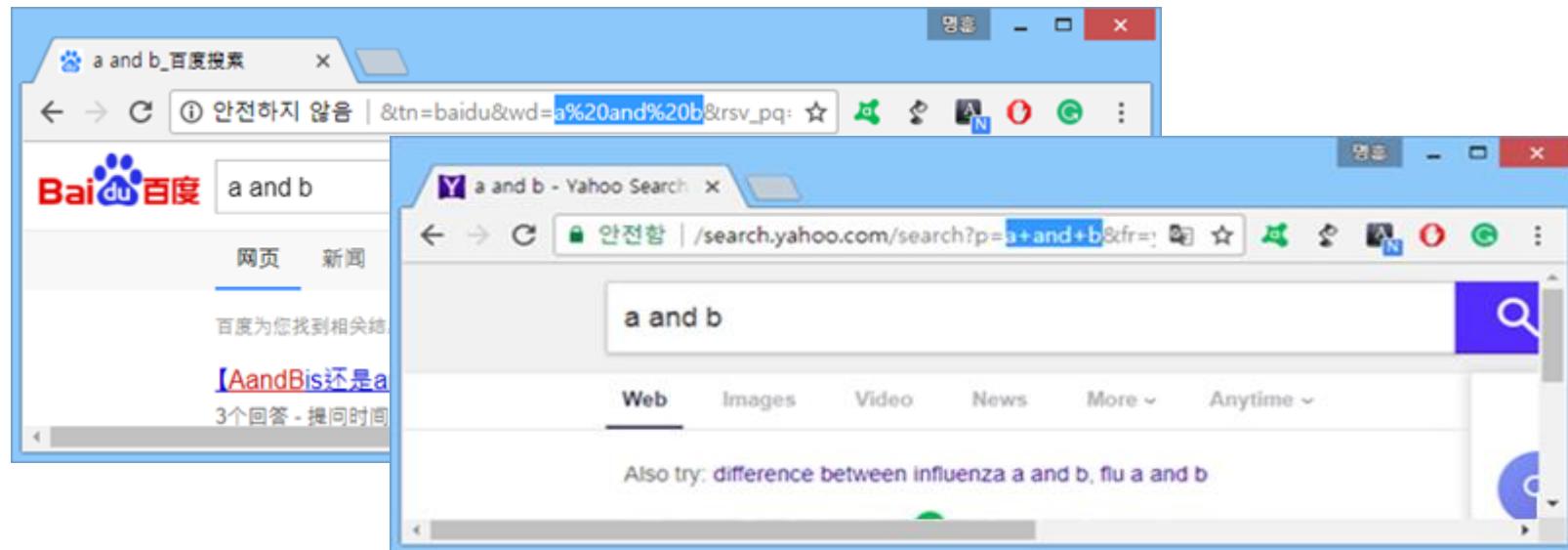
# 룰 최적화 #2



# 를 최적화

## ▶▶ SQL Injection 탐지를

- alert tcp any any -> any 80 (msg:"sql-injection-and"; content:"%20and%20"; nocase;)



- 공백(%20)으로 구분되는 웹요청 헤더 구조  
GET /www.baidu.com/s?wd=a%20and%20b HTTP/1.1  
GET /www.baidu.com/s?wd=a|and|b HTTP/1.1

# 를 최적화



## ▶▶ SQL Injection 탐지를

- 탐지 패턴 ‘ and ‘ 사용 근거는?
- 테스트용 웹서버 및 DB 생성

The screenshot shows the SQLyog Community Edition MySQL GUI interface. The left sidebar displays the database structure with databases like information\_schema, mysql, performance\_schema, snort, sys, and test, and tables like users, Views, Stored Procs, and Functions under the test database. The main window has a 'Query' tab where the following SQL command is entered:

```
1 insert into users(id, pw) values('root', 'n0lids')
```

The 'Messages' tab at the bottom shows the results of the query execution:

1 Messages | 2 Table Data | 3 Objects | 4 History

(1 row(s) affected)  
(0 ms taken)

At the bottom of the interface, it says "Query batch completed successfully".

# 룰 최적화

## ▶ PCRE 를 작성

- 탐지 데이터 조회

The screenshot shows the SQLyog Community Edition interface. The left sidebar displays the database structure with the 'snort' database selected. The main window has a 'Query' tab open, containing the following SQL code:

```
1 select unhex(replace(c.data_payload,'0D0A','20'))
2 from event a, signature b, data c
3 where b.sig_name = 'sql-injection-and'
4 and a.signature = b.sig_id
5 and a.sid = c.sid
6 and a.cid = c.cid
7
```

The results pane shows a single row of data:

unhex(replace(c.data_payload,'0D0A','20'))	8Length
GET /s?ie=utf-8&f=8&	81%3D1
GET /s?ie=utf-8&f=8&	81%3D1
GET /s?ie=utf-8&f=8&	1 HTTP
GET /s?ie=utf-8&f=8&	0b HTTP
GET /s?ie=utf-8&f=8&	0b HTTP
GET /s?ie=utf-8&f=8&	0b HTTP
GET /s?ie=utf-8&f=8&	0b HTTP

A context menu is open over the first row of the results table, with the 'Copy All Rows To Clipboard...' option highlighted.

At the bottom of the interface, it says "Query batch completed successfully" and "0 ms".

# 룰 최적화

## ▶ PCRE 를 작성

- URI 추출



```
[이름 없음] + - GVIM1
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
76 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b HTTP/1.1 User-Agent: curl/7.21.3 (i386-pc-win32) libcurl/7.21.3 OpenSSL/0.9.8q zlib/1.2.5 Host: www.baidu.com Accept: */*
77 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b HTTP/1.1 User-Agent: curl/7.21.3 (i386-pc-win32) libcurl/7.21.3 OpenSSL/0.9.8q zlib/1.2.5 Host: www.baidu.com Accept: */*
78 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b HTTP/1.1 User-Agent: curl/7.21.3 (i386-pc-win32) libcurl/7.21.3 OpenSSL/0.9.8q zlib/1.2.5 Host: www.baidu.com Accept: */*
79 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b HTTP/1.1 User-Agent: curl/7.21.3 (i386-pc-win32) libcurl/7.21.3 OpenSSL/0.9.8q zlib/1.2.5 Host: www.baidu.com Accept: */*
80 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b HTTP/1.1 User-Agent: curl/7.21.3 (i386-pc-win32) libcurl/7.21.3 OpenSSL/0.9.8q zlib/1.2.5 Host: www.baidu.com Accept: */*
/http\/.*
```

```
[이름 없음] + - GVIM1
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
68 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
69 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
70 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
71 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
72 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
73 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
74 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
75 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
76 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
77 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
78 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
79 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
80 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
~
~  
:%s///
```

# 룰 최적화



## ▶ PCRE 를 작성

- 구분기호 추가

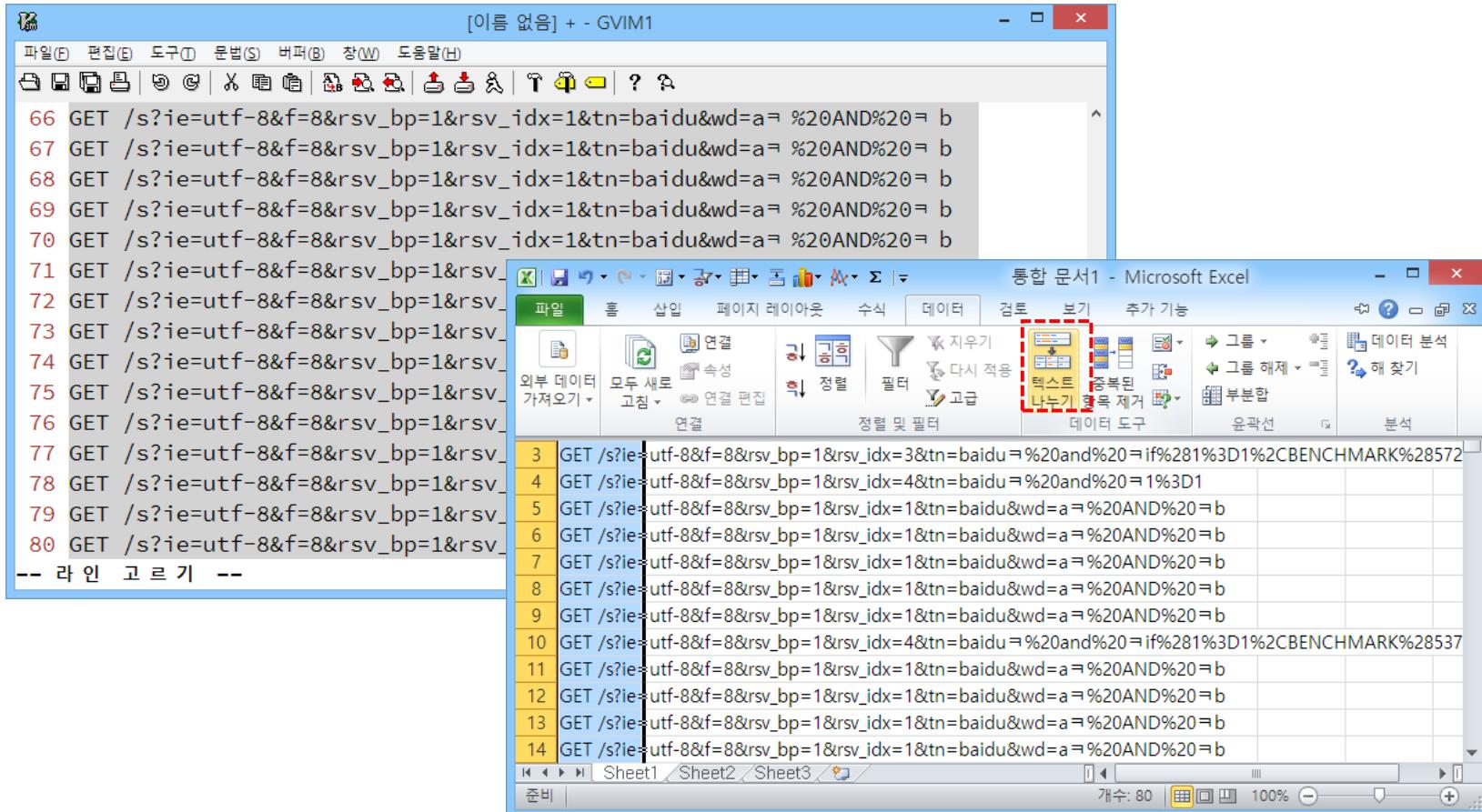
```
[이름 없음] - GVIM1
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
68 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
69 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
70 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a%20AND%20b
71 GET /s?ie=utf-8&f=8&rsv_bp=
72 GET /s?ie=utf-8&f=8&rsv_bp=
73 GET /s?ie=utf-8&f=8&rsv_bp=
74 GET /s?ie=utf-8&f=8&rsv_bp=
75 GET /s?ie=utf-8&f=8&rsv_bp=
76 GET /s?ie=utf-8&f=8&rsv_bp=
77 GET /s?ie=utf-8&f=8&rsv_bp=
78 GET /s?ie=utf-8&f=8&rsv_bp=
79 GET /s?ie=utf-8&f=8&rsv_bp=
80 GET /s?ie=utf-8&f=8&rsv_bp=
~  
~  
/%20and%20
```

```
[이름 없음] - GVIM1
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
68 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a=: %20AND%20=: b
69 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a=: %20AND%20=: b
70 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a=: %20AND%20=: b
71 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a=: %20AND%20=: b
72 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a=: %20AND%20=: b
73 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a=: %20AND%20=: b
74 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a=: %20AND%20=: b
75 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a=: %20AND%20=: b
76 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a=: %20AND%20=: b
77 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a=: %20AND%20=: b
78 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a=: %20AND%20=: b
79 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a=: %20AND%20=: b
80 GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=a=: %20AND%20=: b
~  
~  
:  
:s / = & = /
```

# 룰 최적화

## ▶ PCRE 를 작성

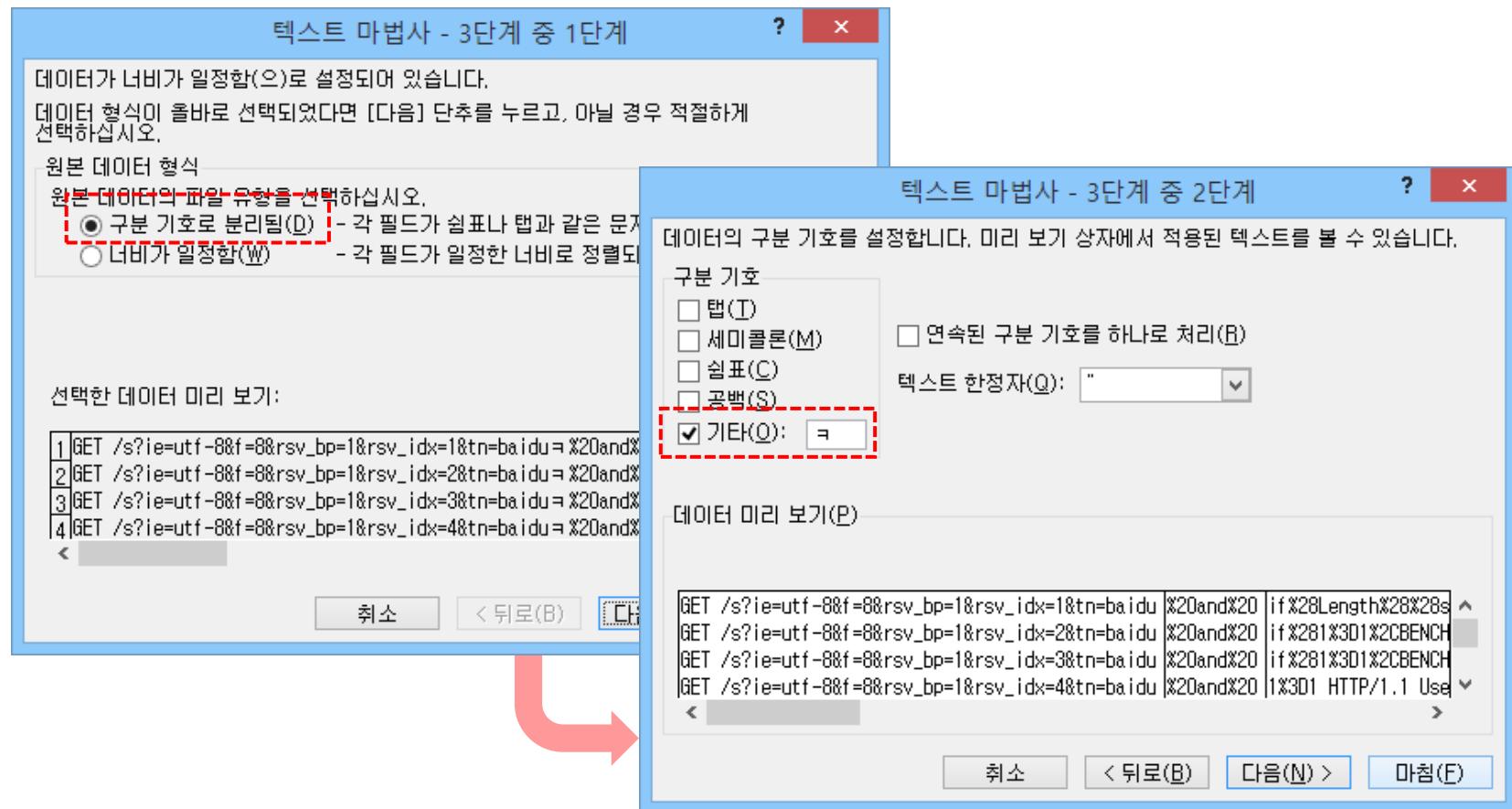
- 엑셀 붙여넣기 후 '텍스트 나누기' 실행



# 룰 최적화

## ▶ PCRE 를 작성

### ▪ 텍스트 정규화 과정



# 룰 최적화



## ▶ PCRE 를 작성

- 텍스트 구분 작업 후 '필드명' 행 추가

통합 문서1 - Microsoft Excel

	C	D	E	F	G	H	I	
1	GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu	if%28Length%28%28select%20distinct%20schema_name%20from%20%	620					
2		if%281%3D1%2CBENCHMARK%28353600%2CMD5%280x41%29%29%2C0	620					
3		if%281%3D1%2CBENCHMARK%2857200%2CMD5%280x41%29%29%2C0	620					
4		1%3D1	620					
5		b	620					
6		b	620					
7		b	620					
8		b	620					
9		b	620					
10		if%281%3D1%2CBENCHMARK%2853733%2CMD5%280x41%29%29%2C0	620					
11		0%29%29%2C0	620					
12		b	620					
13		b	620					
14		b	620					

파일 툴 색인 페이지 레이아웃 수식 데이터 검토 보기 추가 기능

11 가 가 가 가 가 가 가 가

GET /s?ie=utf-8&f=8&rsv\_bp=1&rsv\_idx=1&tn=baidu

1. 잘라내기(I)  
2. 복사(C)  
3. 붙여넣기 옵션:  
4. 선택하여 붙여넣기(S)...  
5. 삽입(I)  
6. 삭제(D)  
7. 내용 지우기(N)  
8. 셀 서식(F)...  
9. 행 높이(R)...  
10. 숨기기(H)  
11. 숨기기 취소(U)

Sheet1 Sheet2 Sheet3

준비 | 개수: 3 | 100%

# 룰 최적화



## ▶ PCRE 를 작성

- '필드명' 행 추가 결과

통합 문서1 - Microsoft Excel

	A	B	C	D	E	F	G	H	I
1	패턴 전	패턴	패턴 후						
2	GET /s?ie=utf-8%20and%20		if%28Length%28%28select%20distinct%20schema_name%20from%20%						
3	GET /s?ie=utf-8%20and%20		if%281%3D1%2CBENCHMARK%28353600%2CMD5%280x41%29%29%2C0						
4	GET /s?ie=utf-8%20and%20		if%281%3D1%2CBENCHMARK%2857200%2CMD5%280x41%29%29%2C0						
5	GET /s?ie=utf-8%20and%20	1%3D1							
6	GET /s?ie=utf-8%20AND%20	b							
7	GET /s?ie=utf-8%20AND%20	b							
8	GET /s?ie=utf-8%20AND%20	b							
9	GET /s?ie=utf-8%20AND%20	b							
10	GET /s?ie=utf-8%20AND%20	b							
11	GET /s?ie=utf-8%20and%20		if%281%3D1%2CBENCHMARK%2853733%2CMD5%280x41%29%29%2C0						
12	GET /s?ie=utf-8%20AND%20	b							
13	GET /s?ie=utf-8%20AND%20	b							
14	GET /s?ie=utf-8%20AND%20	b							

# 룰 최적화



## ▶▶ PCRE 를 작성

- '필드명' 행 고정

통합 문서1 - Microsoft Excel

파일 흘 삽입 페이지 레이아웃 수식 데이터 검토 보기 추가 기능

기본 표시 확대/축소 100% 선택 영역 확대/축소

페이지 나누기 미리 보기 사용자 지정 보기 전체 화면 통합 문서 보기

새 창 모두 정렬 작업 영역 저장 창 전환 매크로

틀 고정(F)

현재 선택 영역을 기준으로 워크시트의 나머지 부분을 스크롤하는 도와 해과 열이 표시되도록 합니다.

첫 행 시트 창 고정(C)

워크시트의 나머지 부분을 스크롤할 때 첫 행이 표시되도록 합니다.

첫 열 고정(C)

워크시트의 나머지 부분을 스크롤할 때 첫 열이 표시되도록 합니다.

3	GET /s?ie=utf-8%20and%20	if%281%3D1%2CBENCHMARK%
4	GET /s?ie=utf-8%20and%20	if%281%3D1%2CBENCHMARK%
5	GET /s?ie=utf-8%20and%20	1%3D1
6	GET /s?ie=utf-8%20AND%20	b
7	GET /s?ie=utf-8%20AND%20	b
8	GET /s?ie=utf-8%20AND%20	b
9	GET /s?ie=utf-8%20AND%20	b
10	GET /s?ie=utf-8%20AND%20	b
11	GET /s?ie=utf-8%20and%20	if%281%3D1%2CBENCHMARK%2853733%2CMD5%280x41%29%29%2C0
12	GET /s?ie=utf-8%20AND%20	b
13	GET /s?ie=utf-8%20AND%20	b
14	GET /s?ie=utf-8%20AND%20	b

Sheet1 Sheet2 Sheet3

준비 개수: 3 100%

# 룰 최적화

## ▶ PCRE 를 작성

- 필드명에 '필터(CTRL + SHIFT + L)' 적용

	A	B	C	D	E	F	G	H	I
1	패턴 전	패턴	패턴 후						
2	GET /s?ie=utf-8%20and%20		if%28Length%28%28select%20distinct%20schema_name%20from%20%						
3	GET /s?ie=utf-8%20and%20		if%281%3D1%2CBENCHMARK%28353600%2CMD5%280x41%29%29%2C						
4	GET /s?ie=utf-8%20and%20		if%281%3D1%2CBENCHMARK%2857200%2CMD5%280x41%29%29%2C						
5	GET /s?ie=utf-8%20and%20		1%3D1						
6	GET /s?ie=UTF-8%20and%20								
7	GET /s?ie=UTF-8%20and%20								

# 룰 최적화



## ▶▶ PCRE 를 작성

### ▪ 탐지패턴 확인

The screenshot shows a Microsoft Excel window titled "통합 문서1 - Microsoft Excel". The formula bar contains the text: "if%28Length%28%28select%20distinct%20schema\_name%20from%20". The search dialog is open, with the following details:

- Pattern:** if%28Length%28%28select%20distinct%20schema\_name%20from%20
- Search Type:** Pattern
- Search Results:** The results list shows several matches:
  - 1. GET /s
  - 2. GET /s
  - 3. GET /s
  - 4. GET /s
  - 5. GET /s
  - 6. GET /s
  - 7. GET /s
  - 8. GET /s
  - 9. GET /s
  - 10. GET /s
  - 11. GET /s
  - 12. GET /s
  - 13. GET /s
  - 14. GET /s
- Filter Options:** The filter dropdown menu includes:
  - 텍스트 오름차순 정렬(S)
  - 텍스트 내림차순 정렬(O)
  - 색 기준 정렬(I)
  - "패턴 후"에서 필터 해제(C)
  - 색 기준 필터(I)
  - 텍스트 필터(E)
- Search Results View:** The main area displays the filtered results, showing multiple occurrences of the pattern across the rows.

# 룰 최적화

## ▶ PCRE 를 작성

- URL 디코딩

The screenshot shows a Microsoft Excel spreadsheet titled "통합 문서1 - Microsoft Excel". The spreadsheet has three rows of data:

	A	B	C	D	E	F	G	H	I
1	패턴 전	패턴	패턴	패턴					
2	GET /s?ie=utf-8%20and%20		if%28Length%28%28select%20distinct%20schema_name%20fr						

A search and replace dialog box is open over the spreadsheet, titled "찾기 및 바꾸기" (Search and Replace). The "찾기(D)" (Search) tab is selected. The "찾을 내용(N)" (Search for) field contains "%3d". The "바꿀 내용(E)" (Replace with) field is empty. The "모두 바꾸기(A)" (Replace All) button is highlighted.

# 룰 최적화



## ▶▶ PCRE 를 작성

- 자주 사용되는 URL 인코딩 패턴

URL 인코딩	URL 디코딩	URL 인코딩	URL 디코딩
%3D	=	%20	공백
%27	'	%3A	:
%22	"	%3C	<
%28	(	%3E	)
%29	)	%2B	+
%2C	,	%60	`
%2E	.	%2F	/

# 룰 최적화

## ▶ PCRE 를 작성

- URL 디코딩 결과

The screenshot shows a Microsoft Excel spreadsheet titled "통합 문서1 - Microsoft Excel". The spreadsheet contains a table with four columns: A, B, C, and D. Column A lists URLs, column B lists patterns, column C lists "Pattern After" (the result of applying the pattern), and column D lists "Pattern After" (another result). A context menu is open over the first row of the table, specifically over the cell containing "패턴 전". The menu includes options like "텍스트 오름차순 정렬(S)", "텍스트 내림차순 정렬(O)", "색 기준 정렬(I)", "색 기준 필터(C)", and "텍스트 필터(F)". A search bar is also present in the menu. The main content area shows several rows of data:

A	B	C	D
1 패턴 전	패턴	패턴 후	패턴 후
2 GET /s?ie=utf-8%	텍스트 오름차순 정렬(S)		select distinct schema_name from %60information_
3 GET /s?ie=utf-8%	텍스트 내림차순 정렬(O)		ICHMARK(353600,MD5(0x41)),0) and 'x'='x
4 GET /s?ie=utf-8%	색 기준 정렬(I)		ICHMARK(57200,MD5(0x41)),0) and 'x'='x
5 GET /s?ie=utf-8%	"패턴 후"에서 필터 해제(C)		
6 GET /s?ie=utf-8%	색 기준 필터(I)		
7 GET /s?ie=utf-8%	텍스트 필터(F)		
8 GET /s?ie=utf-8%			
9 GET /s?ie=utf-8%			
10 GET /s?ie=utf-8%			
11 GET /s?ie=utf-8%			
12 GET /s?ie=utf-8%			
13 GET /s?ie=utf-8%			
14 GET /s?ie=utf-8%			

# 룰 최적화



## ▶▶ PCRE 를 작성

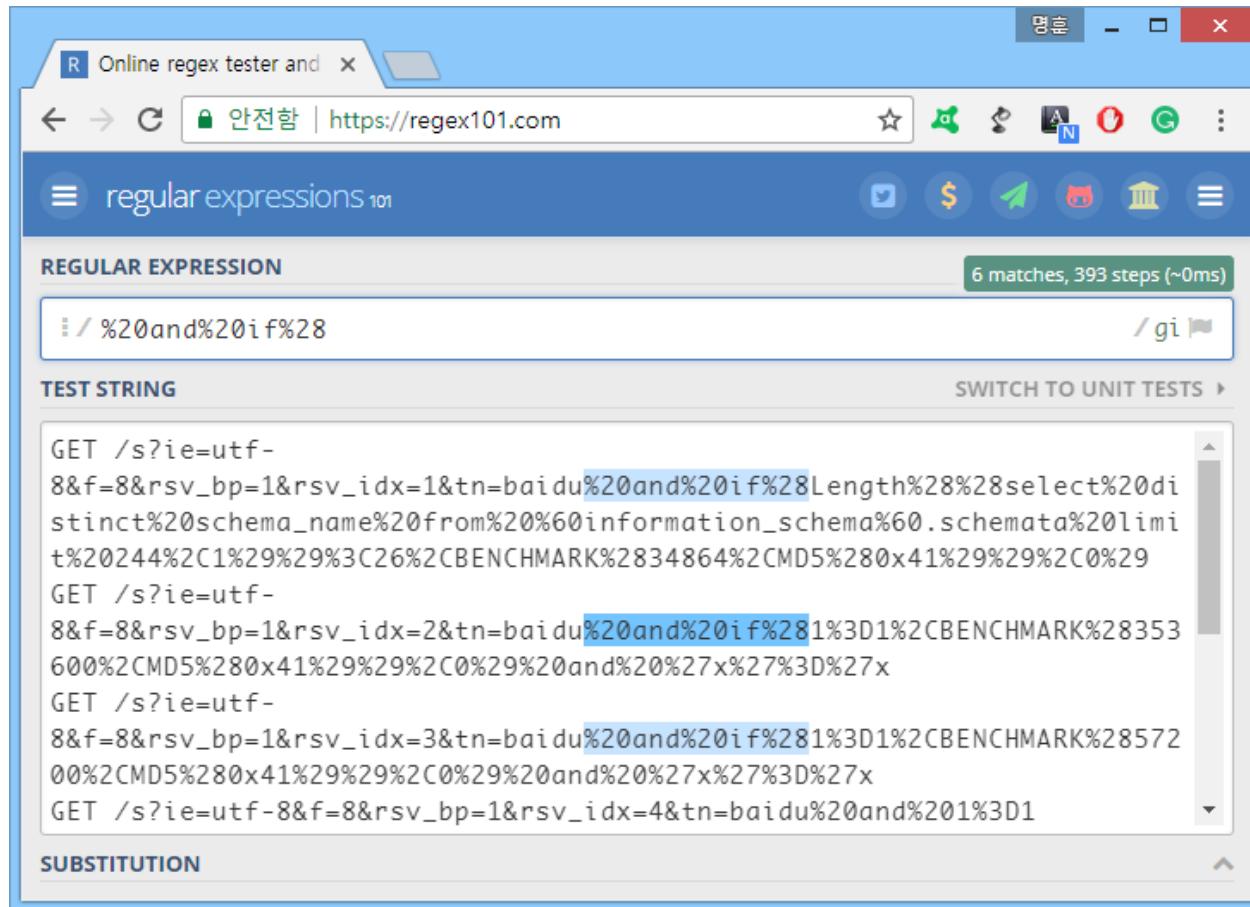
- 룰 정확도는 10% 미만

	A	B	C	D	E	F	G	H	I
1	패턴 전	패턴	패턴 후	패턴 후	판정				
2	GET /s?ie=utf-8%20and%20		if%28Length(if(Length(i	공격					
3	GET /s?ie=utf-8%20and%20		if%281%31if(1=1,BEN	공격					
4	GET /s?ie=utf-8%20and%20		if%281%31if(1=1,BEN	공격					
5	GET /s?ie=utf-8%20and%20		1%3D11=1	공격					
11	GET /s?ie=utf-8%20and%20		if%281%31if(1=1,BEN	공격					
18	GET /s?ie=utf-8%20and%20		if%281%31if(1=1,BEN	공격					
33	GET /s?ie=utf-8%20and%20		if%281%31if(1=1,BEN	공격					
82									
83									
84									
85									
86									
87									

# 룰 최적화

## ▶ PCRE 를 작성

- 룰 정확도를 높여보자



The screenshot shows the 'Online regex tester and visualizer' interface at <https://regex101.com>. The regular expression input field contains the pattern `/%20and%20if%28`. The test string area displays several lines of a GET request, with the portion containing the regex pattern highlighted in blue. The results panel indicates 6 matches found in 393 steps (~0ms). The substitution panel is currently empty.

```
GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu%20and%20if%28Length%28%28select%20distinct%20schema_name%20from%20%60information_schema%60.schemata%20limit%20244%2C1%29%29%3C26%2CBENCHMARK%2834864%2CMD5%280x41%29%29%2C0%29
GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=2&tn=baidu%20and%20if%281%3D1%2CBENCHMARK%28353600%2CMD5%280x41%29%29%2C0%29%20and%20%27x%27%3D%27x
GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=3&tn=baidu%20and%20if%281%3D1%2CBENCHMARK%2857200%2CMD5%280x41%29%29%2C0%29%20and%20%27x%27%3D%27x
GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=4&tn=baidu%20and%201%3D1
```

# 룰 최적화

## ▶ PCRE 를 작성

- 룰 정확도를 높여보자

The screenshot shows the regex101.com interface. In the 'REGULAR EXPRESSION' field, the pattern `\%20and\%28if\%27)?\w(\%27)?%3D(\%27)?\w)` is entered with flags `/gi`. Below it, the 'TEST STRING' field contains several GET requests from a web browser's history, with the regex highlighting specific parts of the URLs. The highlighted areas include portions of the URL parameters, such as 'and', 'if', '27)', '\w', '%3D', and '27)'.

```
REGULAR EXPRESSION: \%20and\%28if\%27)?\w(\%27)?%3D(\%27)?\w)
TEST STRING: GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu%20and%20if%28Length%28%28select%20distinct%20schema_name%20from%20%60information_schema%60.schemata%20limit%2044%2C1%29%29%3C26%2CBENCHMARK%2834864%2CMD5%280x41%29%29%2C0%29
GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=2&tn=baidu%20and%20if%281%3D1%2CBENCHMARK%28353600%2CMD5%280x41%29%29%2C0%29%20and%20%27x%27%3D%27x
GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=3&tn=baidu%20and%20if%281%3D1%2CBENCHMARK%2857200%2CMD5%280x41%29%29%2C0%29%20and%20%27x%27%3D%27x
GET /s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=4&tn=baidu%20and%201%3D1
```

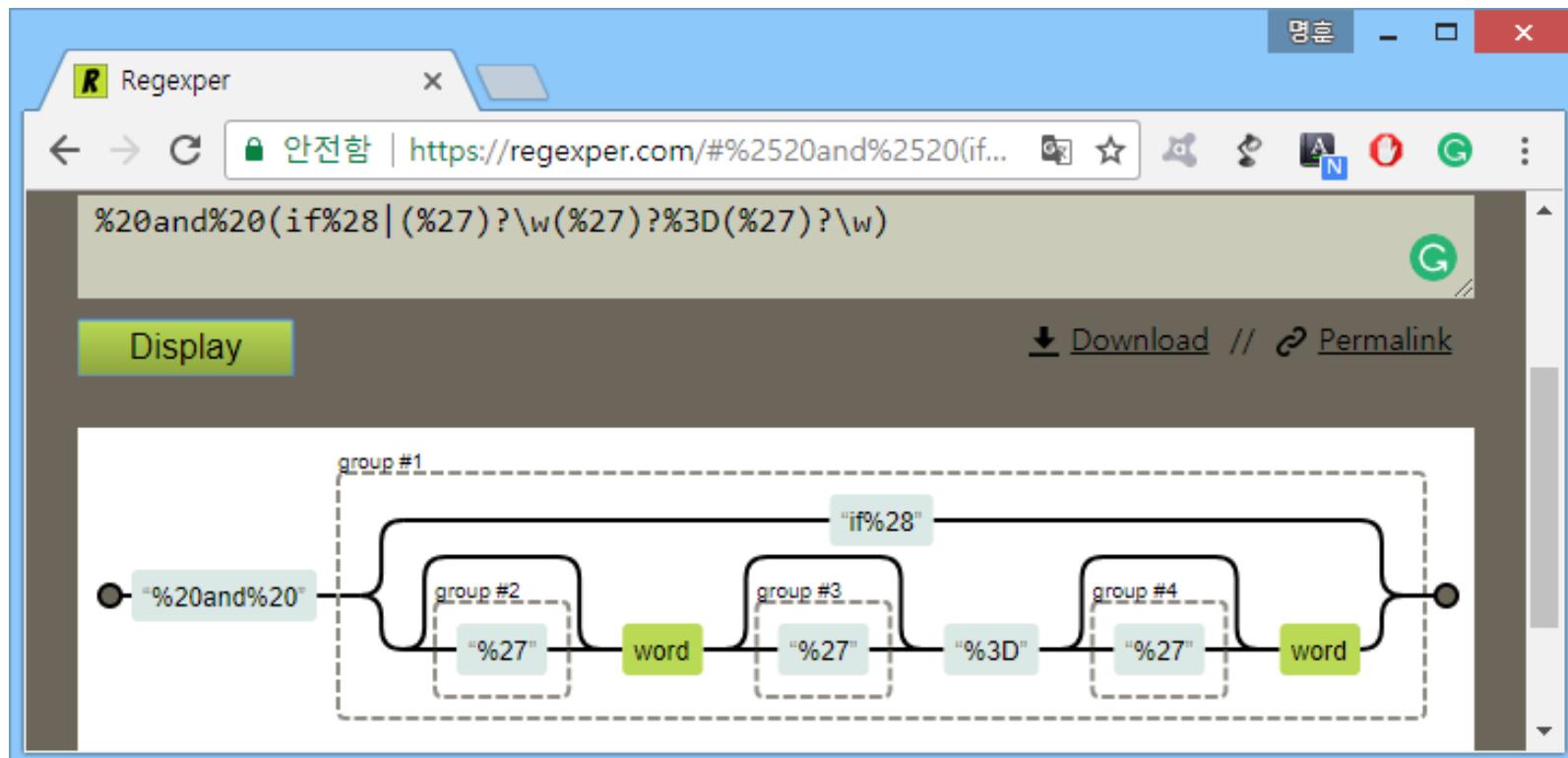
SUBSTITUTION:

# 룰 최적화



## ▶ PCRE 를 작성

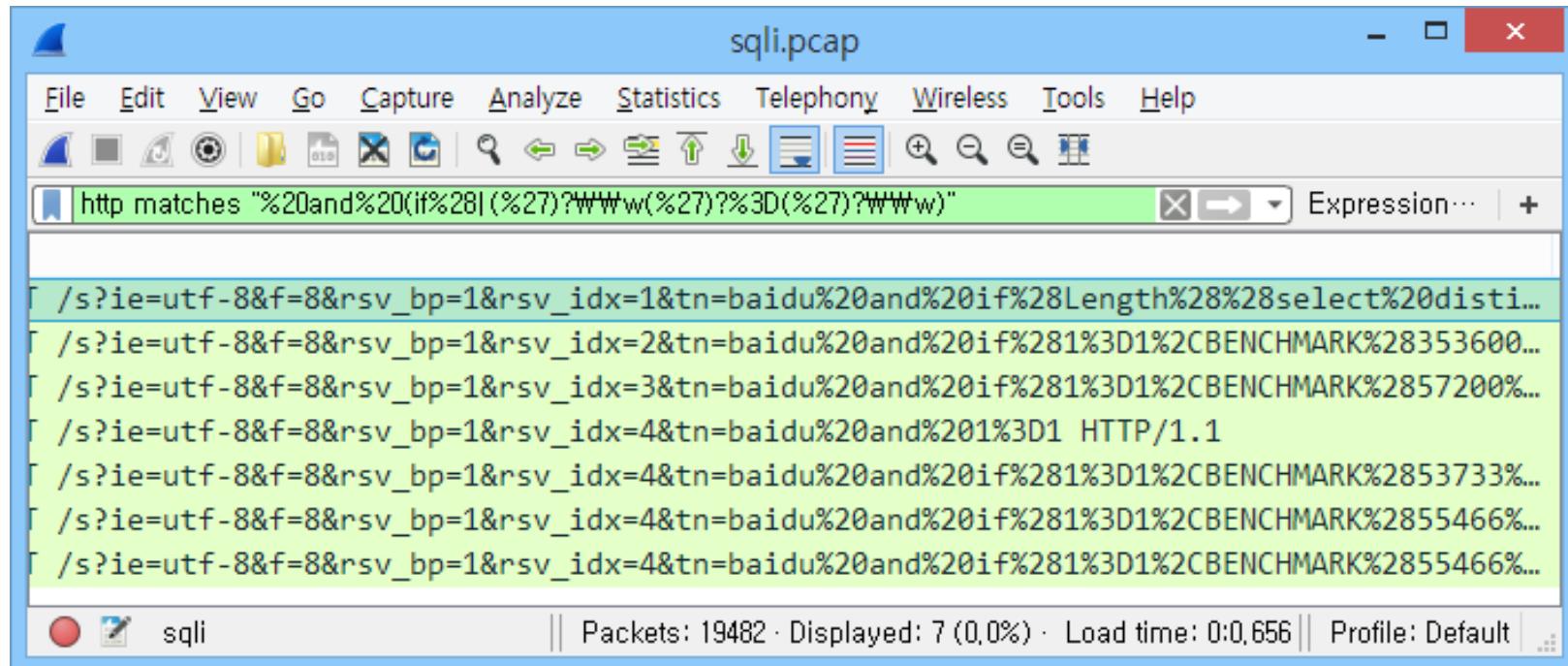
- 룰 정확도를 높여보자



# 룰 최적화

## ▶ PCRE 를 작성

- 룰 정확도를 높여보자
- Wireshark 정규표현식 테스트



# 룰 최적화



## ▶ PCRE 를 작성

- content:"+and+"; nocase;

sql2log (D:\교육\멀티캠) - GVIM

파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)

2008 GET /info/desk?wt=json&indent=true&json.wrf=jsonp1327428850561&q=%22Fi\*%22+  
AND+task%3A4+AND+grow\_type%3A2&start=0&rows=10 HTTP/1.1 Accept: application  
/javascript, \*/\*;q=0.8 Referer: http://dfnexli.com/?G0=news|cinfo& Accept-L  
anguage: ko-KR User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.  
1; Trident/5.0) Accept-Encoding: gzip, deflate Host: dfightli.com Connectio  
n: Keep-Alive k?90?????????吳 죽 ?삶 ? @??

2009 GET /info/desk?wt=json&indent=true&json.wrf=jsonp1327833452954&q=%22%EB%8C%  
80%ED%95%99%EC%83%9D%ED%99%9CSA\*%22+AND+node%3A4+AND+task%3A6&start=0&rows=  
10 HTTP/1.1 Accept: \*/\* Referer: http://dfnexli.com/?G0=news|cinfo& Accept-  
Language: ko-KR User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6  
.1; Trident/4.0; GTB7.2; SLCC2; .NET CLR 2.12.2327; .NET CLR 3.12.2329; .NE  
T CLR 3.12.2329; Media Center PC 6.0; MASM; InfoPath.2; .NET4.0C) Accept-En  
coding: gzip, deflate Host: dfightli.com Connection: Keep-Alive 孔^Y?\_翼 꼈 ? @??

/http\1.\* 2009,154 바 닥

~  
~  
:%s///

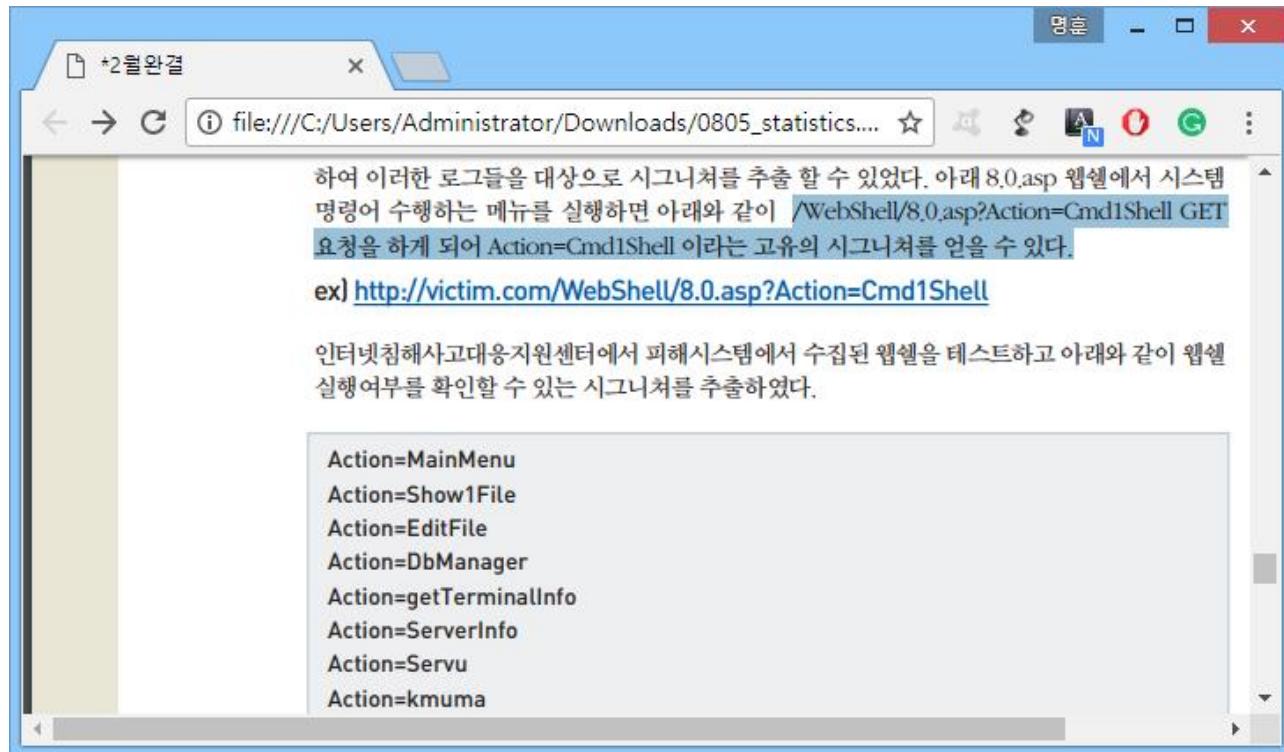
# 룰 최적화 #3



# 룰 최적화

## ▶ PCRE 를 작성

- ASP 웹쉘



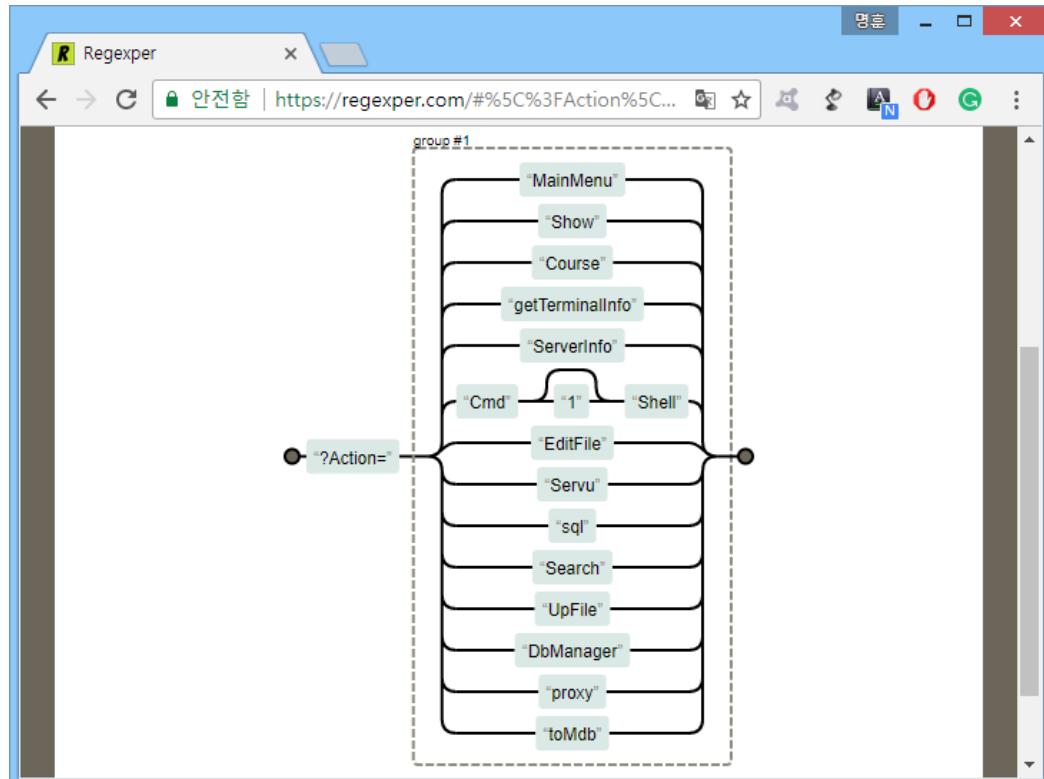
- 해당 변수값을 사용하면 웹쉘인가?

# 룰 최적화



## ▶ PCRE 를 작성

- alert tcp any any -> any 80 (content:"?Action="; nocase;  
pcre:"/\?Action\=(MainMenu>Show|Course|getTerminalInfo|ServerInfo|Cmd1?Shell|  
EditFile|Servu|sql|Search|UpFile|DbManager|proxy|toMdb)/i";)



# 룰 최적화

## ▶ PCRE 를 작성

- ASP 웹쉘 탐지로그 분석

The screenshot shows two windows. The top window is titled "asp\_webshell.log (D:\교육\멀티캠) - GVIM" and displays a log entry for a GET request to "/member/skin/lib/text\_css.css". The entire content of this entry is highlighted with a red rectangle. The bottom window is partially visible and contains log entries for requests to "/Library/member.js", "/member/skin/lib/Default.js", and another entry for "/member/member.asp?action=logout&group\_name=lib". The bottom window also has a command-line interface at the bottom with the text ":%s///".

```
5090 GET /member/skin/lib/text_css.css HTTP/1.1 Accept: */* Referer: http://www.libli.com/member/member.asp?action=logout&group_name=lib Accept-Language: ko Accept-Encoding: gzip, deflate If-Modified-Since: Sun, 23 Apr 2006 04:35:30 GMT If-None-Match: "a271515a8f66c61:25a" User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; GTB6.6; Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) ; EmbeddedWB 14.52 from: http://www.bsalli.com/ EmbeddedWB 14.52; .NET CLR 2.12.2327) Host: www.libli.com Connection: Keep-Alive Cookie: ASPSESSIONIDCCATBQBD=NIKLIIBBCHKHBMJEHBJGK0GP; 150=done; 151=done; 132=done; mbb%5Fwriter=%B9%DA%BB%F3%B9%CC; mbb%5Fmail=podo%5F05%40naver%2Ecom; mbb%5Fhome=; mbb%5Fnick=; blog%5Flogin%5Fname=; blog%5Flogin%5Fid=
```

```
5087 GET /Library/member.js
5088 GET /member/skin/lib/Default.js
5089 GET /member/member.asp?action=logout&group_name=lib
5090 GET /member/skin/lib/text_css.css
~ :%s///
```

# 룰 최적화



## ▶▶ PCRE 를 작성

- ASP 검사 실패, URI 영역 검사 실패

```
asp_webshell.log + (D:\교육\멀티캠) - GVIM
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
파일 편집 도구 문법 버퍼 창 도움말
5077 GET /YangJu_lib/imgreas/quick_img_03.gif
5078 GET /index.php?action=logout
5079 GET /YangJu_lib/imgreas/quick_img_04.gif
5080 POST /YangJu_lib/
5081 GET /member/skin/lib/Default.js
5082 GET /Member/member.asp?action=logout&group_name=LandMaca
5083 POST /open_content/web/login.php?action=logout
5084 GET /login/login_ok.php?action=logout
5085 GET /member/skin/lib/Default.js
5086 GET /YangJu_lib/imgreas/quick_img_06.gif
5087 GET /Library/member.js
5088 GET /member/skin/lib/Default.js
5089 GET /member/member.asp?action=logout&group_name=lib
5090 GET /member/skin/lib/text_css.css
~
/?action=
```

5082,23

바 닥

# 룰 최적화

## ▶ PCRE 를 작성

- ASP 검사 실패, URI 영역 검사 실패



```
asp_webshell.log (D:\교육\멀티캠) - GVIM
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
  1 GET /inc/css/common.css HTTP/1.1 Accept: */* Referer: http://krc.korali.com^
    /BoardNew/member.asp?Action=logout Accept-Language: ko Accept-Encoding: gzi
    p, deflate If-Modified-Since: Tue, 09 Feb 2010 03:50:27 GMT If-None-Match:
    "3e89143ba9ca1:6a0" User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows
    NT 5.1; SV1) Host: krc.korali.com Connection: Keep-Alive Cookie: MPLUS%5FLO
    GIN%5FPWD=; MPLUS%5FLOGIN%5FID=; MPLUS%5FWRITE%5FHOME PAGE=; MPLUS%5FWRITE%5
    /\.asp?action=
                                                1,92      꼭 대기
```

순서	URL
1	http://krc.korali.com/BoardNew/member.asp
2	http://krc.korali.com/inc/css/common.css

# 룰 최적화

## ▶ PCRE 를 작성

- ASP 파일을 포함하지 않은 로그 삭제



```
asp_webshell.log (D:\Edu\Multicam\wexec\snort) - GVIM
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
파일(F) 편집(E) 도구(I) 문법(S) 버퍼(B) 창(W) 도움말(H)
5089 GET /member/member.asp?action=logout&group_name=lib HTTP/1.1 Accept: image/^
gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flas
h, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msw
ord, */* Referer: http://www.libli.com/member/member.asp?action=logout&group_name=lib
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; G
.1; SV1); EmbeddedWB 14.52; .NET CLR 2.12.2327) Host: www.libli.com Connection: Keep-Al
ive Cookie: ASPSESSIONIDCCATBQBD=NIKLIIBBCHKHBMJEHBKGKOGP; 150=done; 151=d
one; 132=done; mbb%5Fwriter=%B9%DA%BB%F3%BB%CC; mbb%5Fmail=podo%5F05%40nave
r%2Ecom; mbb%5Fhome=; mbb%5Fnick=; blog%5Flogin%5Fname=; blog%5Flogin%5Fid=
@.asp?
: v//d
```

# 룰 최적화

## ▶ PCRE 를 작성

- ASP 파일만 포함된 변수 영역 추출

The image shows two windows of the GVIM text editor. Both windows have the title "asp\_webshell.log + (D:\#교육\#멀티캠) - GVIM".  
The left window displays a log entry:  
2578 GET /member/skin/lib/text\_css.css HTTP/1.1 Accept: \*/\* Referer: http://www.^ libli.com/member/ member.asp?action=logout&group\_name=lib Accept-Language: ko Accept-Encoding: gzip, deflate If-Modified-Since: Sun, 23 Apr 2006 04:35:30 GMT If-None-Match: "a271515a8f66c61:25a" User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) ; EmbeddedWB 14.52; .NET CLR 2.12.2327) live Cookie: ASPSESSIONIDCCATBQBD=NIKLIIBBCHKHBMJEHBKGOKGP; 150=done; 151=done; 132=done; mbb%5Fwriter=%B9%DA%BB%F3%B9%CC; mbb%5Fmail=podo%5F05%40naver%2Ecom; mbb%5Fhome=; mbb%5Fnick=; blog%5Flogin%5Fname=; blog%5Flogin%5Fid=  
The right window shows the same log entry with a search pattern overlaid:  
2578 GET /member/skin/lib/text\_css.css HTTP/1.1 Accept: \*/\* Referer: http://www.^ libli.com/member/ member.asp?action=logout&group\_name=lib Accept-Language: ko Accept-Encoding: gzip, deflate If-Modified-Since: Sun, 23 Apr 2006 04:35:30 GMT If-None-Match: "a271515a8f66c61:25a" User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) ; EmbeddedWB 14.52; .NET CLR 2.12.2327) Host: www.libli.com Connection: Keep-Alive Cookie: ASPSESSIONIDCCATBQBD=NIKLIIBBCHKHBMJEHBKGOKGP; 150=done; 151=done; 132=done; mbb%5Fwriter=%B9%DA%BB%F3%B9%CC; mbb%5Fmail=podo%5F05%40naver%2Ecom; mbb%5Fhome=; mbb%5Fnick=; blog%5Flogin%5Fname=; blog%5Flogin%5Fid=  
A blue box highlights the URL portion of the log entry. A blue line at the bottom of the left window highlights the search pattern: "/v[^/]+\.asp\?action\=\w+".  
The status bar at the bottom of the right window shows the command ":%s//&& /".

# 룰 최적화



## ▶ PCRE 를 작성

- 엑셀 복사 후 '텍스트 나누기' 실행

The screenshot shows two windows side-by-side. On the left is a GVIM window titled "asp\_webshell.log + (D:\교육\멀티캠) - GVIM" displaying a log entry. On the right is a Microsoft Excel window titled "통합 문서1 - Microsoft Excel" containing a list of log entries. A red box highlights the "Text to Columns" button in the Excel ribbon's "Data" tab.

GVIM Window Content:

```
2578 [GET /member/skin/lib/text_css.css HTTP/1.1 Accept: */* Referer: http://www.jjli.com/member/] member.asp?action=logout&group_name=lib Accept-Language: ko Accept-Encoding: gzip, deflate If-Modified-Since: Sun, 23 Apr 2006 01:35:30 GMT If-None-Match: mpatile; MSIE 6.0; Windows SIE 6.0; Windows NT 5.1; Sm/ EmbeddedWB 14.52; .NET ep-Alive Cookie: ASPSESSIONID=done; 132=done; mbb%5Fnaver%2Ecom; mbb%5Fhome=Fid=
```

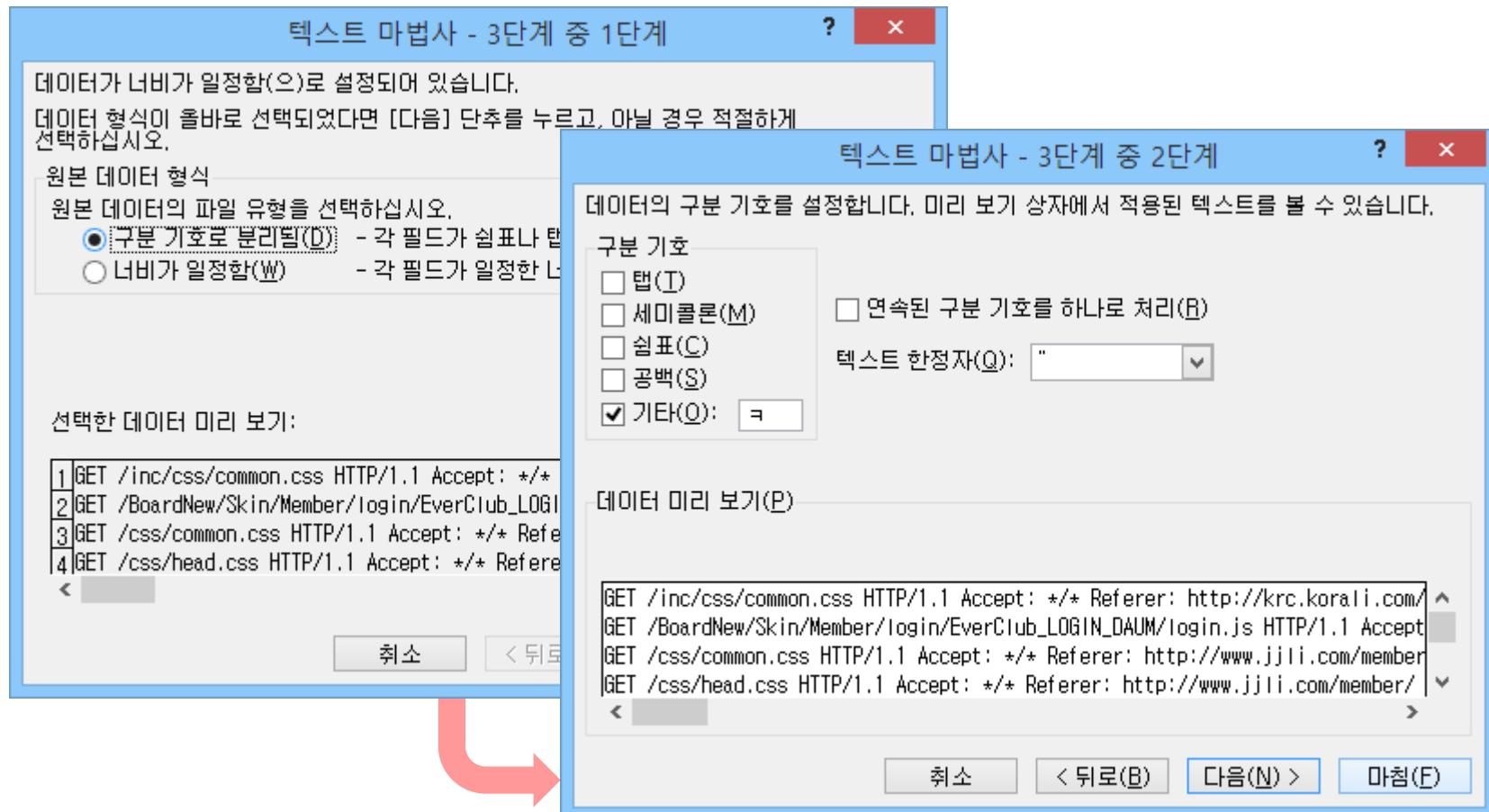
Excel Window Content (Sheet1):

번호	로그 내용
3	GET /css/common.css HTTP/1.1 Accept: */* Referer: http://www.jjli.com/member/ default.asp?action=LOGOUT
4	GET /css/head.css HTTP/1.1 Accept: */* Referer: http://www.jjli.com/member/ default.asp?action=LOGOUT
5	GET /fileserver/attach/ shell.asp?Action>Show1File HTTP/1.1 Accept: image/gif, image/x-bitmap, image/j
6	GET /fileserver/attach/ shell.asp?Action=MainMenu HTTP/1.1 Accept: image/gif, image/x-bitmap, image/
7	GET /member/skin/default/style.css HTTP/1.1 Accept: */* Referer: http://www.anyangjeli.com/member/ mem
8	GET /Skin/Member/login/VER20_LOGIN_INVERS/style.css HTTP/1.1 Accept: */* Referer: http://www.gyeongsan
9	GET /member/skin/child/text_css.css HTTP/1.1 Accept: */* Referer: http://www.libli.com/member/ member.as
10	GET /inc/css/style.css HTTP/1.1 Accept: */* Referer: http://www.funli.com/ member.asp?Action=logout
11	GET /Skin/Member/login/VER20_LOGIN_SP_GREEN/style.css HTTP/1.1 Accept: */* Referer: http://cyberseodali.c
12	GET /inc/css/style.css HTTP/1.1 Accept: */* Referer: http://www.funli.com/ member.asp?Action=logout
13	GET /inc/css/style.css HTTP/1.1 Accept: */* Referer: http://www.funli.com/ member.asp?Action=logout
14	GET /inc/css/style.css HTTP/1.1 Accept: */* Referer: http://www.funli.com/ member.asp?Action=logout

# 룰 최적화

## ▶ PCRE 를 작성

- 텍스트 정규화 과정



# 룰 최적화



## ▶▶ PCRE 를 작성

- 파일과 변수 영역 분리

The screenshot shows a Microsoft Excel window titled "통합 문서1 - Microsoft Excel". In the background, there is a list of URLs in column A:

A
1 member.asp?Action=logout
2 member.asp?Action=logout
3 default.asp?action=LOGOUT
4 default.asp?action=LOGOUT
5 shell.asp>Action=Show1File
6 shell.asp>Action=MainMenu
7 member.asp?action=logout
8 member.asp?Action=logout
9 member.asp?action=logout
10 member.asp?Action=logout
11 Member.asp?Action=logout
12 member.asp?Action=logout
13 member.asp?Action=logout
14 member.asp?Action=logout

In the foreground, a "Text Wizard" dialog box is open, titled "텍스트 마법사 - 3단계 중 2단계". It contains the following settings:

- 구분 기호:
  - 탭(I)
  - 세미콜론(M)
  - 쉼표(C)
  - 공백(S)
  - 기타(Q)
- 연속된 구분 기호를 하나로 처리(B)
- 텍스트 한정자(Q):

The "데이터 미리 보기(P)" section shows the following preview:

```
member.asp|Action=logout  
member.asp|Action=logout  
default.asp|action=LOGOUT  
default.asp|action=LOGOUT
```

At the bottom of the dialog box are buttons: "취소", "< 뒤로(B)", "다음(N) >", and "마침(E)".

# 룰 최적화



## ▶▶ PCRE 를 작성

- 변수와 변수값 영역 분리

The screenshot shows a Microsoft Excel spreadsheet titled "통합 문서1 - Microsoft Excel". The spreadsheet contains a table with two columns: "A" and "B". Column A lists URLs, and column B lists actions. Rows 1 through 14 show entries like "member.asp" and "Action=logout". The "Text Wizard" dialog box is open, titled "텍스트 마법사 - 3단계 중 2단계". It displays the following configuration:

- 구분 기호:
  - 탭(T)
  - 세미콜론(M)
  - 쉼표(O)
  - 공백(S)
  - 기타(Q):
- 연속된 구분 기호를 하나로 처리(B):
- 텍스트 한정자(Q):

The "데이터 미리 보기(P)" section shows a preview of the data with the selected delimiter (equal sign). The preview window contains the following text:

Action	logout

At the bottom of the dialog, there are buttons for "취소", "< 뒤로(B)", "다음(N) >", and "마침(F)".

# 룰 최적화



## ▶ PCRE 를 작성

### ▪ 필드명 추가

통합 문서1 - Microsoft Excel

파일 툴 삽입 페이지 레이아웃 수식 데이터 검토 보기 추가 기능

member.asp

1 member.asp Action  
2 잘라내기(I) Action  
3 복사(C) action  
4 붙여넣기 옵션: action  
5 선택하여 붙여넣기(S)... Action  
6 삽입(I) Action  
7 삭제(D) Action  
8 내용 지우기(N) action  
9 셀 서식(F)... Action  
10 행 높이(R)... Action  
11 승기기(H) Action  
12 승기기 취소(U) Action

통합 문서1 - Microsoft Excel

파일 툴 삽입 페이지 레이아웃 수식 데이터 검토 보기 추가 기능

member.asp

A2

A	B	C
file	var	value
member.asp	Action	logout
member.asp	Action	logout
default.asp	action	LOGOUT
default.asp	action	LOGOUT
shell.asp	Action	Show1File
shell.asp	Action	MainMenu
member.asp	action	logout
member.asp	Action	logout
member.asp	action	logout

클 고정(F)  
현재 선택 영역을 기준으로 워크시트의 나머지 부분을 스크롤하는 동안 행과 열이 표시되도록 합니다.

첫 행 고정(R)  
워크시트의 나머지 부분을 스크롤할 때 첫 행이 표시되도록 합니다.

첫 열 고정(C)  
워크시트의 나머지 부분을 스크롤할 때 첫 열이 표시되도록 합니다.

# 룰 최적화

## ▶▶ PCRE 를 작성

- 피벗 테이블 : 다양한 조합의 데이터 분석 가능

The screenshot shows a Microsoft Excel window titled "통합 문서1 - Microsoft Excel". In the top-left corner of the ribbon, the "파일" tab is selected. Below it, the "삽입" tab is active, indicated by a red dashed box. On the far left of the ribbon, there is a dropdown menu for "피벗 테이블" which is also highlighted with a red dashed box. A "피벗 테이블" dialog box is open in the foreground, overlaid on the Excel interface. This dialog box is titled "피벗 테이블 만들기" and contains the following text:  
분석할 데이터를 선택하십시오.  
 표 또는 범위 선택(S)  
표/범위(I): Sheet1!\$A\$1:\$C\$2579  
 외부 데이터 원본 사용(U)  
연결 선택(C)...  
연결 이름:  
피벗 테이블 보고서를 넣을 위치를 선택하십시오.  
 새 워크시트(N)  
 기존 워크시트(E)  
위치(L):

A	B
1 file	var
2 member.asp	Action
3 member.asp	Action
4 default.asp	action
5 default.asp	action
6 shell.asp	Action
7 shell.asp	Action
8 member.asp	action
9 member.asp	Action
10 member.asp	action

At the bottom right of the dialog box are two buttons: "확인" (Confirm) and "취소" (Cancel). The status bar at the bottom of the Excel window shows "준비" and "100%".

# 룰 최적화



## ▶▶ PCRE 를 작성

- 파일과 변수 발생 관계

The screenshot shows a Microsoft Excel window titled "통합 문서1 - Microsoft Excel". The ribbon tabs include 파일, 흰, 삽입, 페이지 레이아웃, 수식, 데이터, 검토, 보기, 추가 기능, 옵션, 디자인. The formula bar shows "F4 shell.asp". The main table has columns A through G. The "피벗 테이블 도구" ribbon tab is selected. The "피벗 테이블 필드 목록" pane on the right lists fields: file (checked), var, value. The "보고서에 추가할 필드 선택:" section shows "file" checked. The "아래 영역 사이에 필드를 끌어 놓으십시오." section contains "보고서 필터" and "열 레이블". The "행 레이블" and "값" sections show "value" and "개수 : v...". The status bar at the bottom shows "평균: 42.88888889 개수: 10 합계: 386".

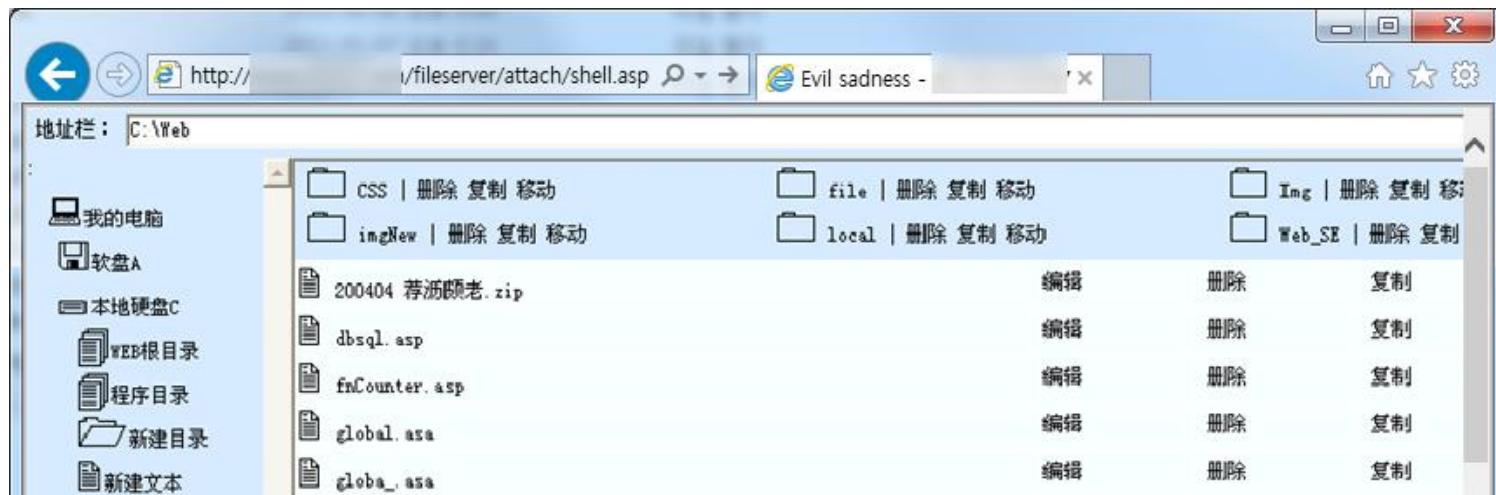
	A	B	C	D	E	F	G
4	행 레이블	default.asp	logout.asp	mboard.asp	member.asp	shell.asp	upload.asp
5	adminab					1	
6	Cmd1Shell					20	
7	course		30			1	
8	getTerminalInfo					11	
9	login					70	
10	logout	69		1	5	2246	
11	MainMenu					1	
12	MMD					1	
13	ScanPort					82	
14	Show1File					17	
15	sql					23	
16	upfile						
17	총합계	99	1	5	2257	193	23

# 를 최적화



## ▶▶ PCRE 를 작성

- shell.asp 접속



- alert tcp any any -> any 80 (content: "asp?Action="; nocase; http\_uri;  
pcre:"/asp\?Action\=(MainMenu|Show|Course|getTerminalInfo|ServerInfo|Cmd1?Shell|EditFile  
|Servu|sql|Search|UpFile|DbManager|proxy|toMdb)/Ui";)

# 룰 최적화



## ▶ 룰을 만드는 두 가지 방법

- 위험해 보이는 문자열을 찾거나(블랙리스트)

장점	알려진 공격은 확실히 방어한다
단점	알려지지 않은 공격은 방어하지 못한다

- 위험하지 않은 문자열을 회피하거나(화이트리스트)

장점	알려지지 않은 공격도 방어할 수 있다
단점	위험하지 않은 패턴, 즉 정상 패턴 정의가 어렵다

## ▶ 패턴매칭은 완벽하지 않다

- 패턴 일치 = 공격?

# 이상징후 분석



# 네트워크 보안



## ▶ 이상징후 분석?

The screenshot shows a Google search results page for the query "anomaly detection". The results are displayed under the "이미지" (Images) tab. The search bar at the top contains the query "anomaly detection". Below the search bar, there are several filters: machine learning, google analytics, numenata anomaly, iot, clustering, statistical, series, time series, data, python, and unsupervised. The main area displays a grid of approximately 40 image thumbnails, each showing various plots, graphs, and diagrams related to anomaly detection in different fields like machine learning, IoT, and data science.

# 네트워크 보안

## ▶▶ 이상징후 분석

- 데이터의 변화를 가장 쉽게 알 수 있는 방법



# 네트워크 보안

## ▶ 통계 분석



[단원 : 통계 (統計) ]

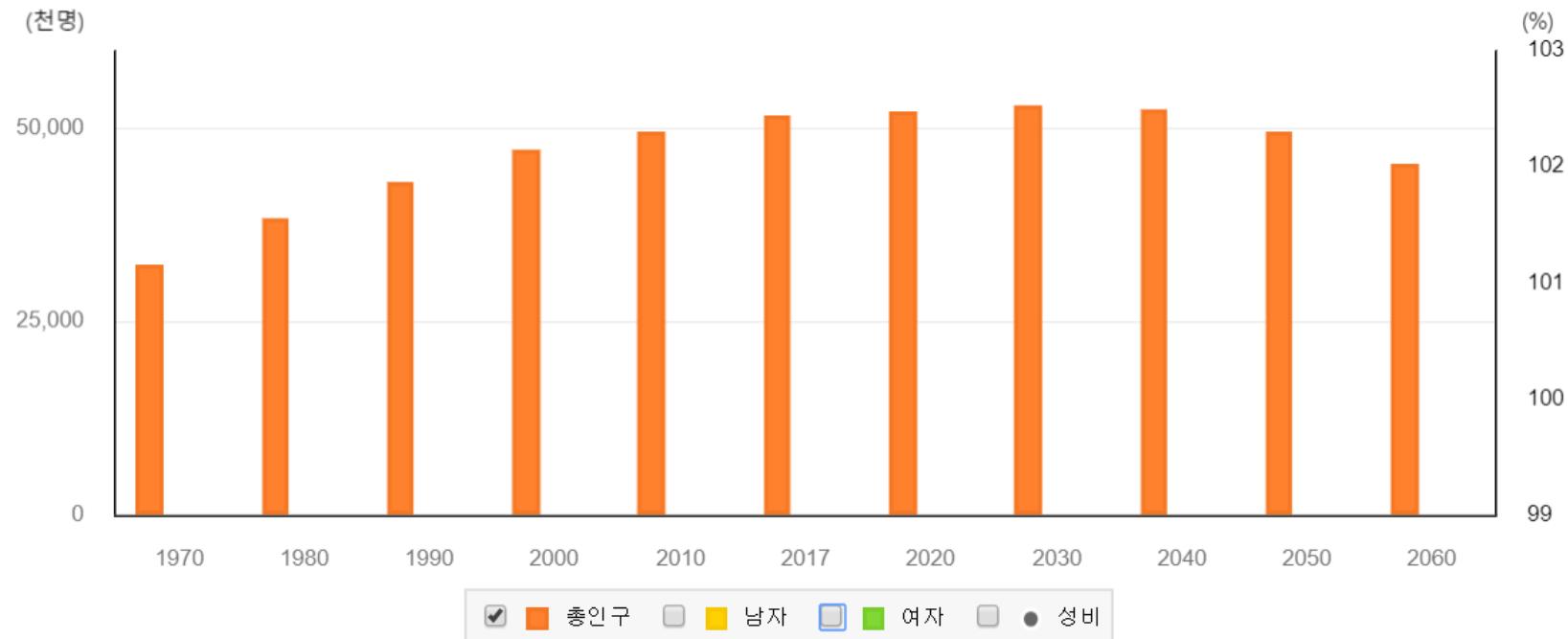
- 어떤 현상을 종합적으로 한 눈에 알아보기 쉽게 일정한 체계에 따라 숫자로 나타냄.



# 네트워크 보안

## ▶ 통계 분석

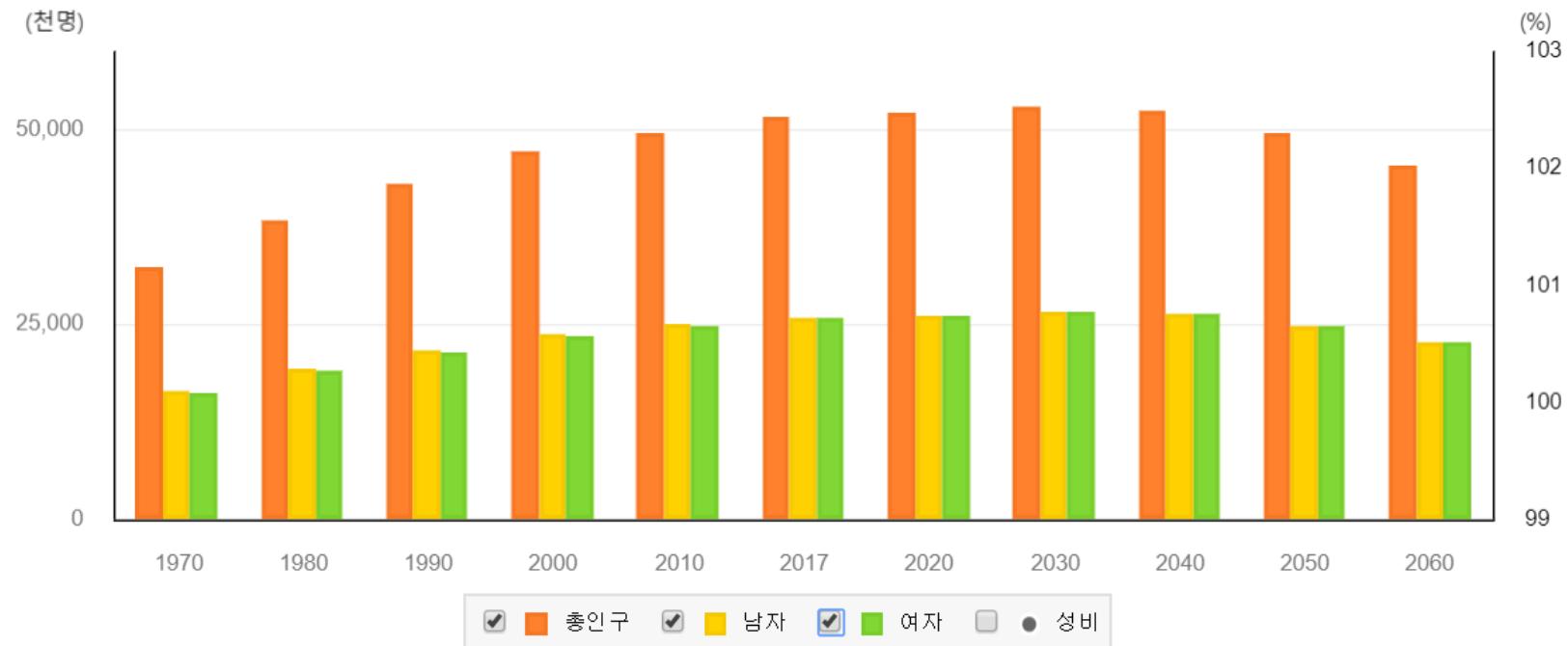
### ▪ 인구조사



# 네트워크 보안

## ▶ 통계 분석

### ▪ 성별 인구조사

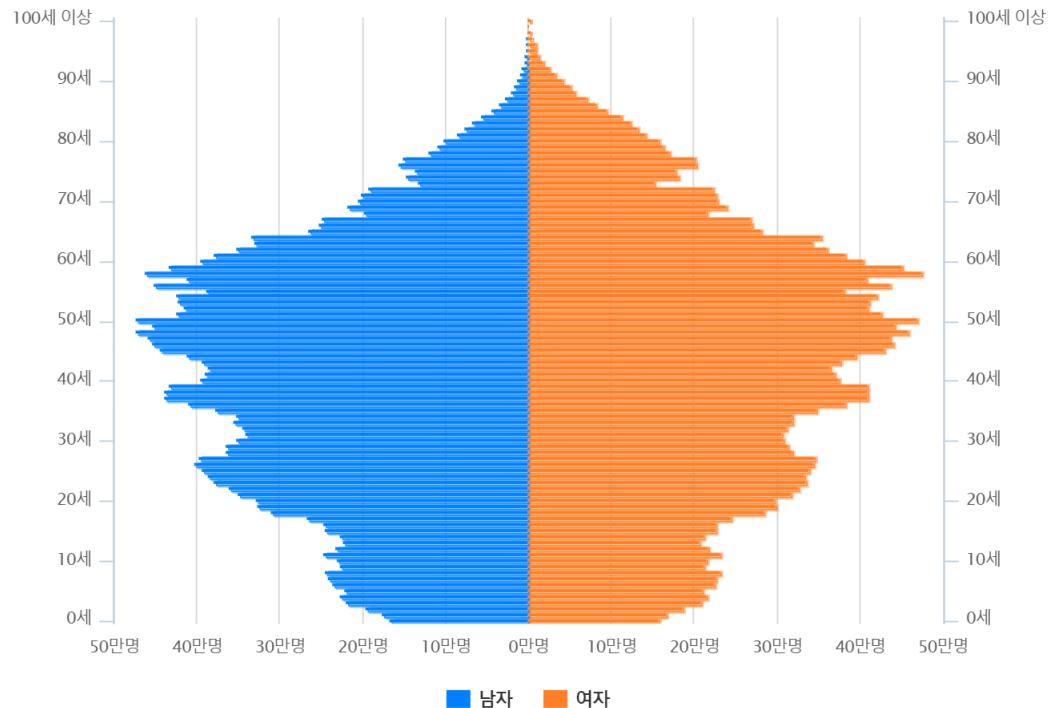


# 네트워크 보안



## ▶ 통계 분석

### ▪ 연령별 인구조사



- 특징 : 상태를 세분화할수록 더 잘 알게 된다.
- 상태 분류를 통한 데이터 분석 구조?

# 네트워크 보안



## ▶ 통계 분석

- 상태(필드)가 분리된 데이터베이스

	필드1	필드2	필드3	필드4	
레코드	성별	이름	나이	지역	테이블
	여자	신사임당	17	강릉	
	남자	홍길동	20	한양	
	남자	임꺽정	22	황해도	

## ▶ 데이터베이스 구조

- 필드 : 동일 성격의 데이터 집합
- 레코드 : 여러 필드로 구성된 데이터 집합
- 테이블 : 필드와 레코드의 집합

# 네트워크 보안



## ▶ 데이터 분석

### ▪ SQL: 데이터 계산 언어

업무	
	요구사항
데이터 엔지니어	<ul style="list-style-type: none"><li>• 비즈니스를 이해하고 대량의 데이터셋을 가공</li><li>• 데이터 파이프라인 개발 및 관리</li><li>• 사내 데이터 애널리스트와 데이터 사이언티스</li><li>• 제품을 최적화하기 위한 분석 도구 개발</li><li>• AWS, 애저(Azure), GCP 등 클라우드 환경에서 대량의 데이터 관리 백엔드 시스템 개발</li><li>• Hadoop(Hadoop), 스파크(Spark) 등을 이용해 대용량 데이터 분산 처리 시스템 개발</li></ul>
데이터 애널리스트	<ul style="list-style-type: none"><li>• 최적의 의사결정을 내리는 데 도움을 주는 비즈니스 인사이트 제공</li><li>• 데이터의 경향, 패턴, 이상치 등을 인식하기 위한 시각화 진행, 보고서 작성</li><li>• 비즈니스 팀과 연계해 각 팀의 전략을 수립하거나 업무 효율화에 필요한 데이터를 수집 및 분석</li></ul>
데이터 사이언티스트	<ul style="list-style-type: none"><li>• 머신러닝 모델을 사용해 정형, 비정형 데이터에서 인사이트 창출</li><li>• 사내 데이터를 이용해 고객 행동 패턴 모델링 진행, 패턴을 찾아내거나 이상치 탐지</li><li>• 예측 모델링, 추천 시스템 등을 개발해 비즈니스 의사결정에 필요한 인사이트 제공</li></ul>

# 네트워크 보안



## ▶ 데이터 분석?

- 데이터 계산

A large red arrow points from the bottom window to the top window.

secure.log (D:\) - GVIM

```
217564 Dec 30 08:39:55 Sensor sshd[23445]: Received disconnect from ::ffff:192.1^
217565 Dec 30 08:40:08 Sensor sshd[23452]: Accepted password for ykmkelho from :
217566 Dec 30 08:43:09 Sensor sshd[23454]: Received disconnect from ::ffff:192.1
217567 Dec 31 06:41:40 Sensor sshd[25183]: Accepted password for ykmkelho from :
217568 Dec 31 06:42:44 Sensor sshd[25185]: Received disconnect from ::ffff:192.1
217569 Dec 31 06:55:47 Sensor sshd[25218]: Accepted password for ykmkelho from :
217570 Dec 31 06:56:05 Sensor sshd[25220]: Received disconnect from ::ffff:192.1
```

A large red arrow points from the bottom window to the top window.

secure.csv - Microsoft Excel

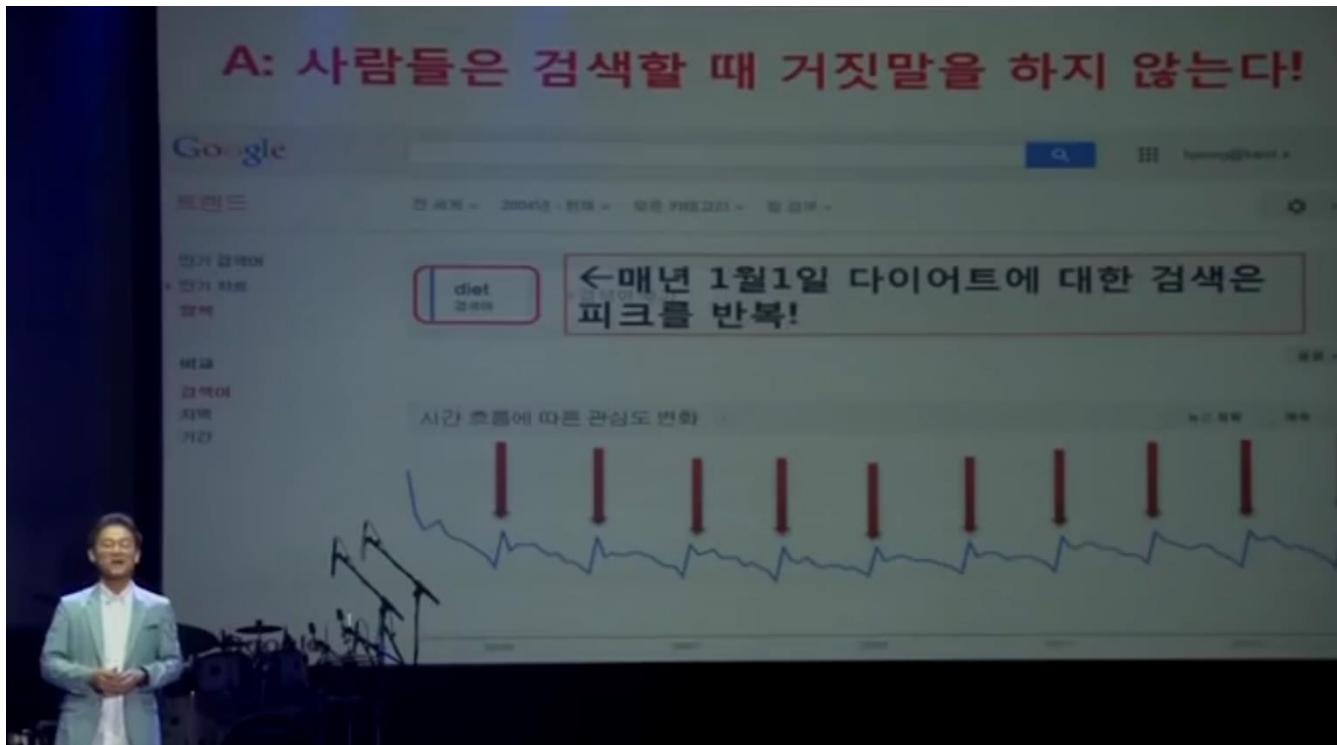
	A	B	C	D	E	F	G	H	I	J	K
1	date	time	host	proc	proc_nu	msg	keyword	ip	id		
159	18-Jan	17:31:21	Sensor	sshd	1854	pam_unix(pam_unix(sshd:session) root					
162	18-Jan	17:41:40	Sensor	sshd	1909	pam_unix(pam_unix(sshd:session) root					
165	18-Jan	17:44:00	Sensor	sshd	1947	pam_unix(pam_unix(sshd:session) root					
2572	13-Mar	20:34:38	Sensor	sshd	1853	pam_unix(pam_unix(sshd:session) root					
3586	09-May	4:05:01	Sensor	sshd	1834	pam_unix(pam_unix(sshd:session) root					
3607	09-May	18:10:59	Sensor	sshd	1894	pam_unix(pam_unix(sshd:session) root					

# 네트워크 보안



## ▶ 데이터 분석

- <https://www.youtube.com/watch?v=4ymo2hdRuQg>
- 제일 중요한 조건: 정확한 데이터





## ▶ 데이터 분석

- 보안 분야에서도 정확한 데이터의 숫자를 세어야 한다

무의미한  
통계

- 전통적 보안관제 분야
- 핵심 네트워크 보안 로그(IDS, IPS 등)
- 필터링(패턴매칭)된 기록
- 를 정확도 개선을 통해 오탐 문제 해결

VS

유의미한  
통계

- 보안관제 소외 분야
- 방화벽, 서버 로그(OS, 웹, DNS 등)
- 사실 관계를 그대로 기록
- 통계분석을 통해 미탐 문제 해결

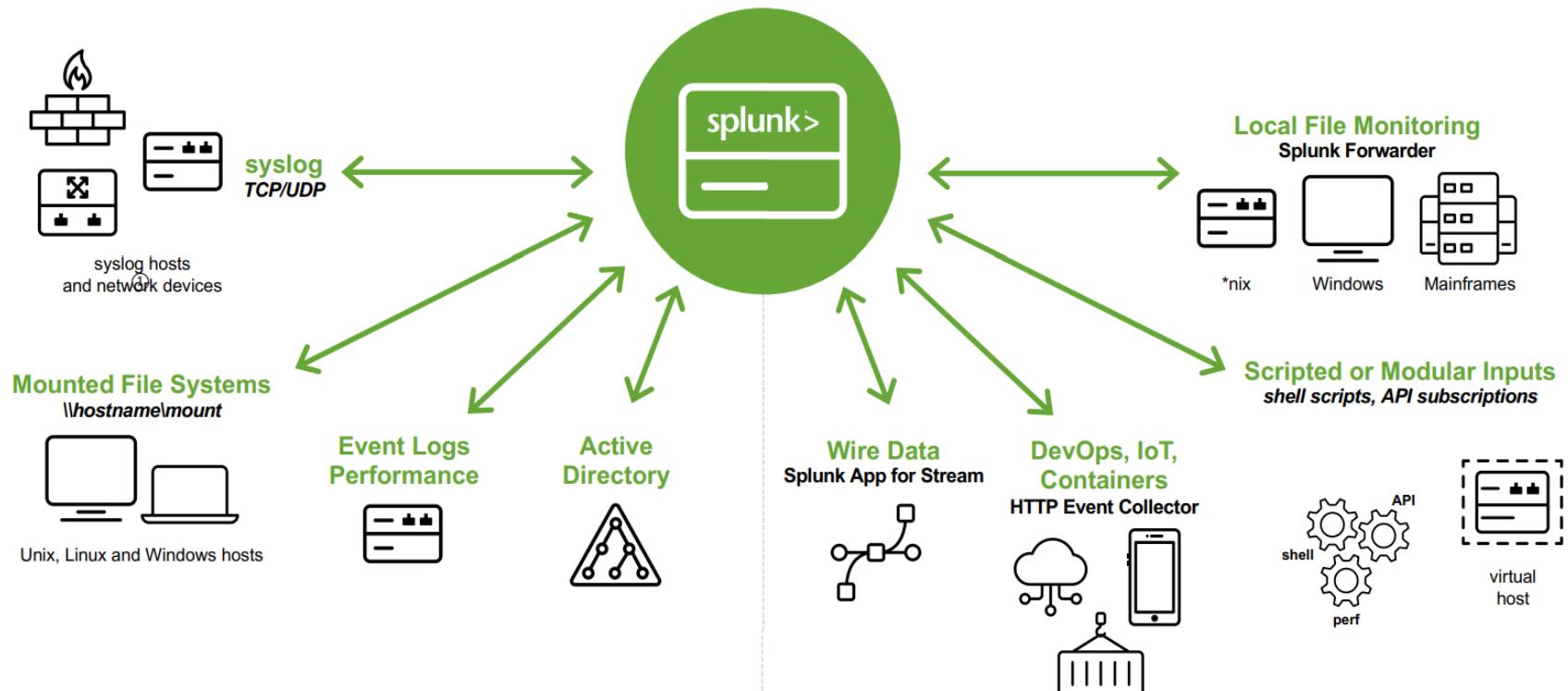
# 스플렁크



# 스플렁크

## ▶ 이기종 데이터 분석 플랫폼

- 다양한 로컬/원격 데이터 연동 지원



# 스플렁크



## ▶▶ 인덱서

- 데이터 저장(인덱싱)
- 포워더/검색 헤드/마스터 기능 포함(헤비 포워더)

인덱서 구성 요소	역할
인덱서(Indexer)	데이터 저장(인덱싱) 및 가공
포워더(Forwarder)	데이터 수집, 전송 및 가공
검색 헤드(Search Head)	검색 관리
마스터(Master)	클러스터 관리

## ▶▶ 포워더

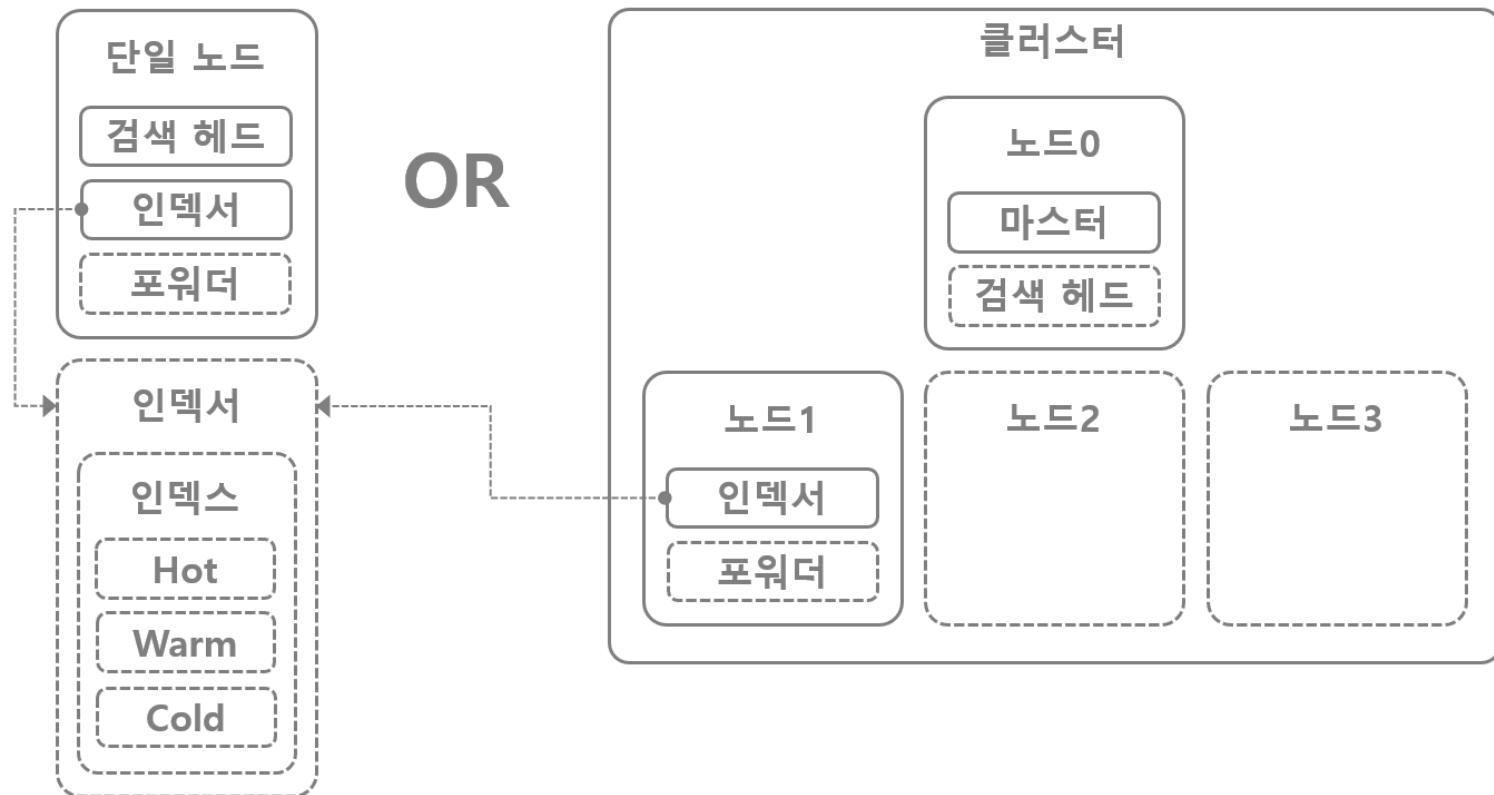
- 데이터 수집/가공/전송

# 스플렁크



## ▶ 분산 데이터베이스 환경 지원

- 단독/클러스터 운영 지원





## ▶ 인덱스

- 데이터 저장 단위(DB 테이블 역할)

인덱서		버킷	특징	유지 관리(indexes.conf)
인덱스	Hot	Hot	데이터 읽기, 쓰기 가능	다음 조건 중 하나라도 일치하면 Warm 버킷으로 이동 ① 서비스 재시작 ② maxDataSize(기본값: 750MB) ③ maxHotBuckets(기본값: 3개) ④ maxHotSpanSecs(기본값: 90일)
	Warm		읽기만 가능	다음 조건 중 하나라도 일치하면 Cold 버킷으로 이동 ① maxHotSpanSecs(기본값: 90일) ② maxWarmDBCount(기본값: 300개)
	Cold			다음 조건 중 하나라도 일치하면 <u>Frozen</u> 버킷으로 이동(삭제) ① maxTotalDataSizeMB(기본값: 500GB) ② frozenTimePeriodInSecs(기본값: 6년)



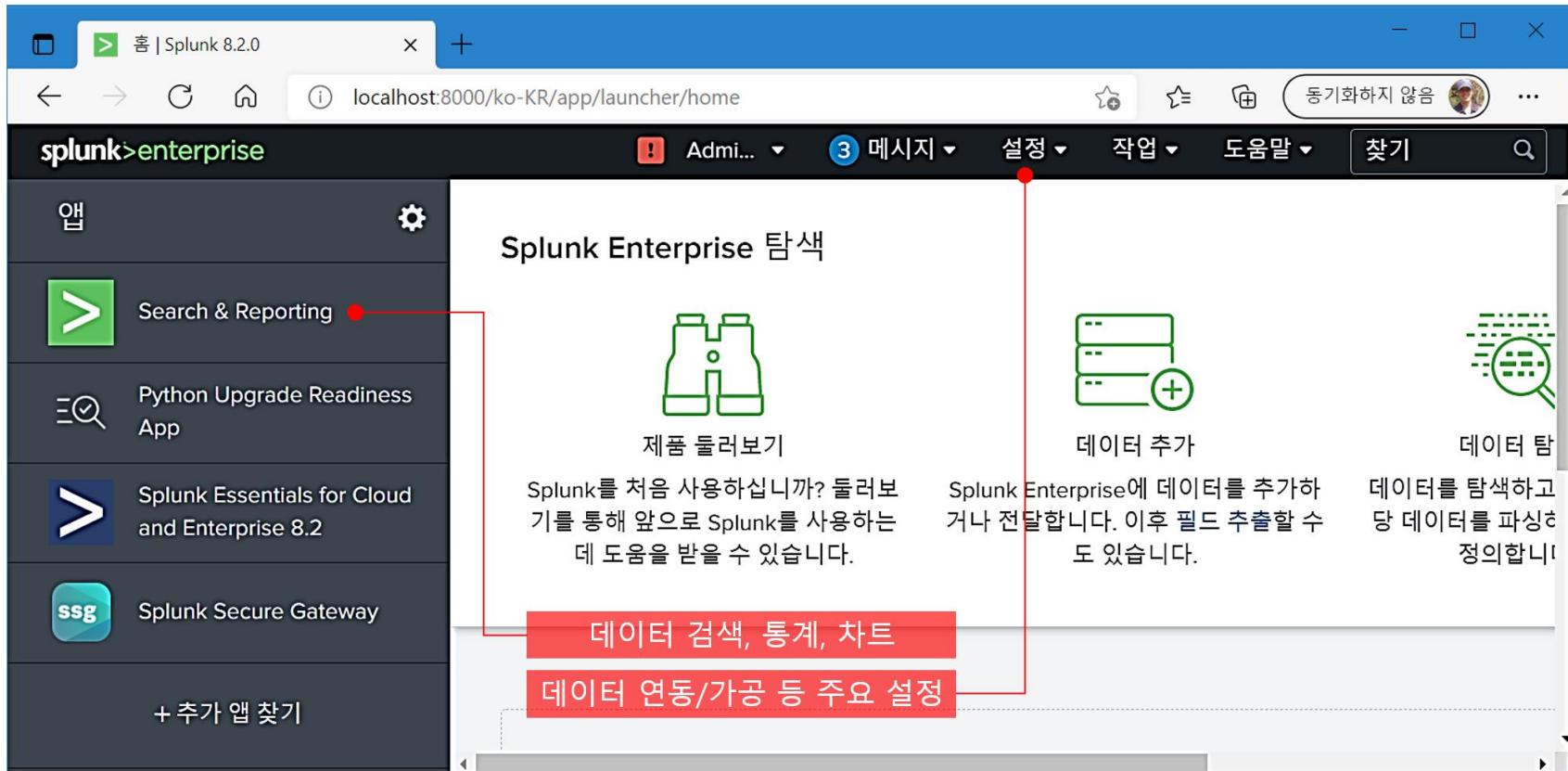
## ▶ 라이선스 구분

- [splunk.com](http://splunk.com) 회원 가입 후 Trial 라이선스 다운로드 가능

구분	설명
Enterprise 라이선스	모든 기능 지원
Trial 라이선스	<ul style="list-style-type: none"><li>① 저장 용량(1일 500MB)을 제외한 모든 기능 지원</li><li>② 60일 후 Enterprise 또는 Free 라이선스로 전환 필요</li></ul>
Free 라이선스	<ul style="list-style-type: none"><li>① 단일 노드로만 운영 가능</li><li>② 인증 등 보안 기능 사용 불가</li><li>③ 1일 저장 용량 500MB 제한</li></ul>

# 스플렁크

## ▶ 스플렁크 주요 메뉴



# 스플렁크

## ▶ 데이터 연동

- 파일 업로드/실시간 모니터/포워더 연동

The screenshot shows the Splunk 8.2.0 interface. At the top, there's a browser header with the URL `192.168.56.100:8000/ko-KR/app/launcher/home`. Below it is the Splunk logo and the word "enterprise". The navigation bar includes icons for "설정" (Settings), "작업" (Jobs), "도움말" (Help), and a search bar. A red arrow points to the "설정" icon. The main content area has two columns. The left column, titled "Spl", contains icons for "Search & Reporting", "Python Upgrade Readiness App", "Splunk Essentials for Cloud and Enterprise 8.2" (with a "업데이트" button), and "Splunk Secure Gateway". The right column, titled "Data", lists various configuration options: "데이터 추가" (highlighted with a blue box and red arrow), "데이터 탐색", "사용자 인터페이스", "경고 작업", "고급 검색", "모든 설정", "시스템", and "서버 설정". To the right of these, there are Korean descriptions for each item.

데이터 추가	데이터
데이터 탐색	데이터 입력
사용자 인터페이스	전달 및 수신
경고 작업	인덱스
고급 검색	보고서 가속 요약
모든 설정	가상 인덱스
시스템	Source Type
서버 설정	분산 환경
	인덱서 클러스터링
	포워더 관리
	데이터 패브릭
	Federated search
	분산 검색

# 스플렁크

## ▶ 데이터 연동

- 인덱서 자체 수집: 파일 업로드/실시간 모니터
- 에이전트를 이용한 수집: 포워더

The screenshot shows the Splunk 9.0.1 Data Manager interface with the title '데이터 추가 | Splunk 9.0.1' in the browser bar. The URL is 'localhost:8000/ko-KR/manager/search/adddata'. The page displays three main data input methods:

- 업로드** (Upload): Represented by a green upward arrow icon. Sub-options include "내 컴퓨터의 파일" (Files on my computer), "로컬 로그 파일" (Local log files), and "로컬에 있는 정형 파일(예: CSV)" (Structured files in local storage (e.g., CSV)). A link "데이터 추가에 관한 튜토리얼" (Tutorial on adding data) is also present.
- 모니터** (Monitor): Represented by a green monitor icon showing a graph. Sub-options include "이 Splunk 플랫폼 인스턴스의 파일 및 포트" (Files and ports of this Splunk platform instance), "파일 - HTTP - WMI - TCP/UDP - 스크립트" (File - HTTP - WMI - TCP/UDP - Script), and "외부 데이터 원본에 대한 모듈식 입력" (Modular input for external data sources).
- 전달** (Forward): Represented by a green icon of a stack of documents with an arrow pointing right. Sub-options include "Splunk 포워더의 데이터" (Data from Splunk forwarder) and "파일 - TCP/UDP - 스크립트" (File - TCP/UDP - Script).

# 스플렁크



## ▶ 데이터 연동

- 실시간 모니터/포워더의 데이터 연동 설정 지정

```
splunk add monitor -source 파일경로 -index 인덱스명 -sourcetype 소스타입명
```

- 포워더의 데이터 수신 서버 설정 지정

```
splunk add forward-server 서버IP:9997
```

- 데이터 수신 서버의 수신 포트 설정 지정

```
splunk enable listen 9997
```

# 스플렁크

## ▶ 데이터 연동

- 포워더 연동을 위한 인덱서 설정(설정 > 전달 및 수신)

The screenshot shows the Splunk Manager interface at `localhost:8000/ko-KR/manager/launcher/forwardreceive`. The top navigation bar includes links for Admin, Message, Setting, Job, Help, and Search.

**전달 및 수신**

**데이터 전달**

다음의 Splunk 인스턴스 간 전달을 설정합니다.

전달 기본값      데이터 전달 시 사본 저장 여부  
전달 설정      데이터 수신 노드(host:port) 지정

데이터 송신자(forwarder) 역할

+ 새로 추가

**데이터 수신**

다른 인스턴스에서 전달된 데이터를 수신하도록 이 인스턴스를 설정합니다.

수신 설정      데이터 수신 포트 지정      데이터 수신자(indexer) 역할

+ 새로 추가



## ▶▶ SPL(Search Process Language)

- UNIX + SQL 컨셉

The screenshot shows the Splunk 9.0.1 search interface. The search bar contains the following SPL query:

```
1 index=apache_sample  
2 | rename uri_path as url
```

Annotations with red circles and numbers highlight parts of the query:

- ① 검색 (Search): Points to the first line of the query.
- ② 가공 (Processing): Points to the second line of the query.

The search results summary indicates 300,000 events found. The main pane displays a histogram of event times. Below the histogram, the event list table shows the first few entries:

i	시간	이벤트
>	22/06/07 22:00:00.000	183.60.215.50 -- [07/Jun/2022:13:00:00 +0000] "GET /server HTTP/1.1" 200 182 "-" "Mozilla/5.0 (compatible; p://www.easou.com/search/spider.html)" method = GET url = /scripts/netcat-webserver

On the left sidebar, under '선택한 필드' (Selected Fields), the fields 'method' and 'url' are listed. The page navigation shows page 1 of 7.



## ▶▶ SPL(Search Process Language)

- 검색(search)

- ① 검색창에 최초 입력되는 구문은 (search 명령 없이) search 구문으로 인식
- ② 필드="값" 구조의 검색어를 AND, OR, NOT, IN 등의 연산자와 함께 사용
- ③ 필드="값" 구조에서 인용부호 생략해도 필드="값" 구조로 인식
- ④ 타 명령어는 인용부호가 있을 때만 필드 ="값" 구조로 인식
- ⑤ 필드를 지정하지 않으면 전체 원본 데이터가 저장된 \_raw 필드 자동 지정
- ⑥ 연산자를 지정하지 않으면 AND 연산자 자동 지정
- ⑦ 필드와 연산자는 대소문자를 구분, 검색어는 구분하지 않음
- ⑧ 와일드카드(\*) 사용 가능

# 스플렁크



## ▶▶ SPL(Search Process Language)

- 가공(eval)
- 문자열/숫자 연산 및 replace, split 등 다양한 함수 기능 지원

The screenshot shows the Splunk 9.0.1 search interface. The search bar contains the SPL command:

```
1 index=apache_sample
2 | eval newfield=method."-".status
```

The results summary indicates 300,000 events found. The main pane displays a histogram of event times. Below the histogram, the event table shows the first few rows of data, including the timestamp, IP address, and the evaluated status field.

시간	이벤트
22/06/07 22:00:00.000	183.60.215.50 - - [07/Jun/2022:13:00:00 +0000] "GET / HTTP/1.1" 200 182 "-" "Mozilla/5.0 (compatible; p://www.easou.com/search/spider.html)" method = GET   newfield = GET-200   status = 200

# 스플렁크



## ▶▶ SPL(Search Process Language)

- eval GUI: 설정 > 필드 > 계산된 필드

The screenshot shows the Splunk Settings interface for creating a new calculated field. The URL in the browser is `localhost:8000/ko-KR/manager/search/data/props/calcfields/_new?action=edit`. The form fields are as follows:

- 대상 앱: search
- 적용 대상: sourcetype (선택한 값: iis)
- 이름 \*: file (계산 결과 저장 필드)
- 평가식 \*: `replace(url, ".*V(.*)", "1")` (선택한 평가식 예제: `x + 3`)

At the bottom right are the '취소' (Cancel) and '저장' (Save) buttons.



## ▶▶ SPL(Search Process Language)

- 통계(stats, chart, timechart)

The screenshot shows the Splunk 9.0.1 search interface. In the search bar, the following SPL command is entered:

```
1 index=apache_sample  
2 | stats count by method
```

The search results table displays the count of events for different HTTP methods:

method	count
GET	298479
HEAD	1379
OPTIONS	14
POST	103
PROPFIND	10



## ▶▶ SPL(Search Process Language)

- 통계(stats, chart, timechart)

The screenshot shows the Splunk search interface with the following details:

- Search Bar:** The search bar contains the SPL command: `index=apache_sample | chart count by method`.
- Results Summary:** Below the search bar, it says "300,000개의 이벤트 (23/04/01 18:54:24.000 이전)" and "이벤트 샘플링 없음".
- Panel Selection:** The "통계 (5)" panel is selected.
- Table Results:** A table displays the count of events for different HTTP methods. The columns are "method" and "count".

method	count
GET	298479
HEAD	1379
OPTIONS	14
POST	103
PROPFIND	10



## ▶▶ SPL(Search Process Language)

- 통계(stats, chart, timechart)

The screenshot shows the Splunk 9.0.1 search interface. In the search bar, the following SPL command is entered:

```
1 index=apache_sample  
2 | stats count by method, status
```

The results table displays the count of events for different methods and statuses:

method	status	count
GET	+	10
GET	200	266161
GET	206	6409
GET	301	7071
GET	304	5518



## ▶▶ SPL(Search Process Language)

- 통계(stats, chart, timechart)

The screenshot shows the Splunk 9.0.1 user interface. In the search bar, the following SPL command is entered:

```
1 index=apache_sample  
2 | chart count by method, status
```

The results show a histogram with the following data:

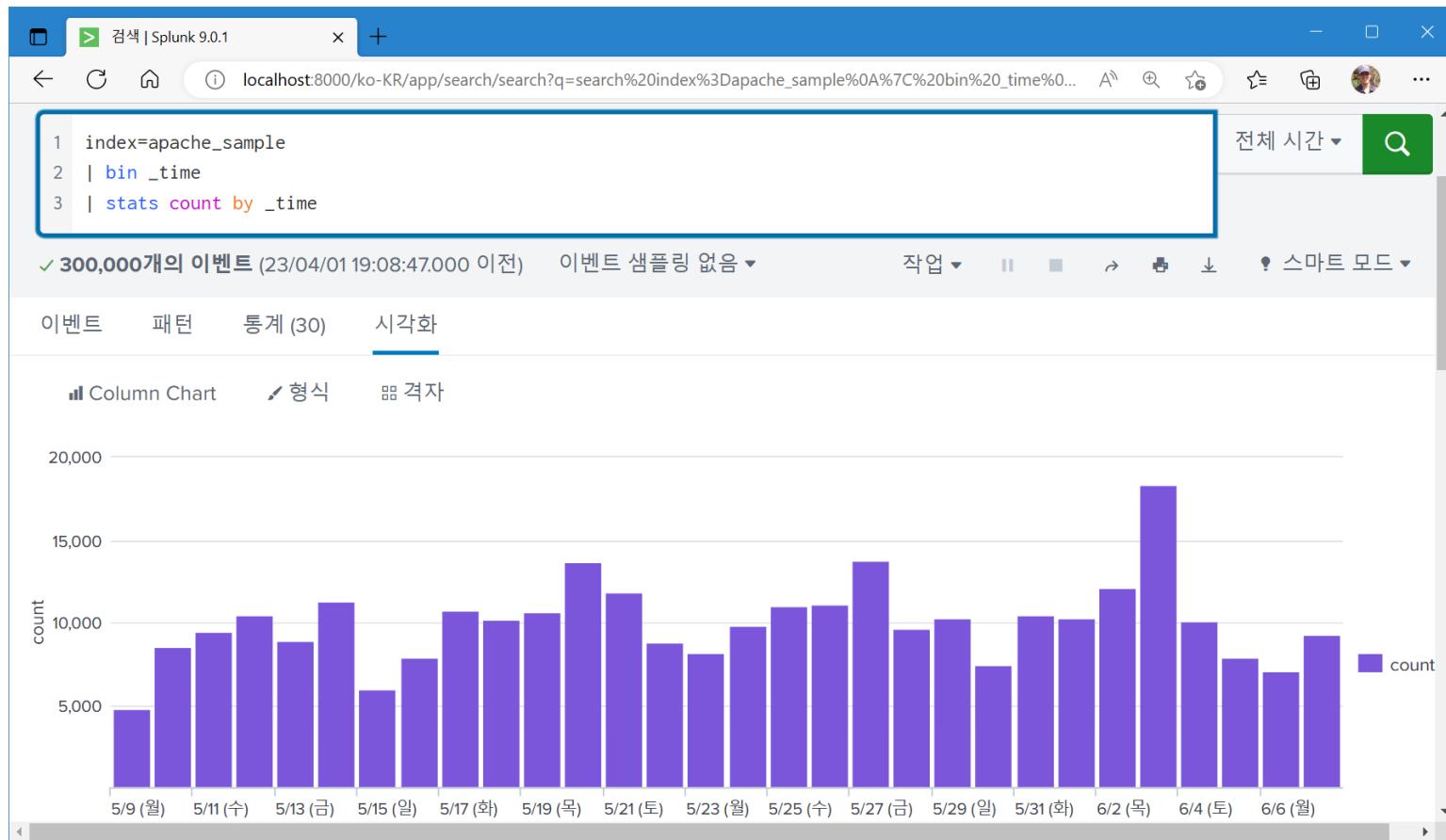
method	200	206	301	304	400	403	404	405	500	OTHER	
GET	10	266161	6409	7071	5518	9	114	13180	0	3	4
HEAD	0	1108	2	34	0	0	1	234	0	0	0
OPTIONS	0	8	0	0	0	0	0	0	0	6	0
POST	0	51	0	0	0	0	0	52	0	0	0

# 스플렁크



## ▶▶ SPL(Search Process Language)

### ▪ 시계열 차트

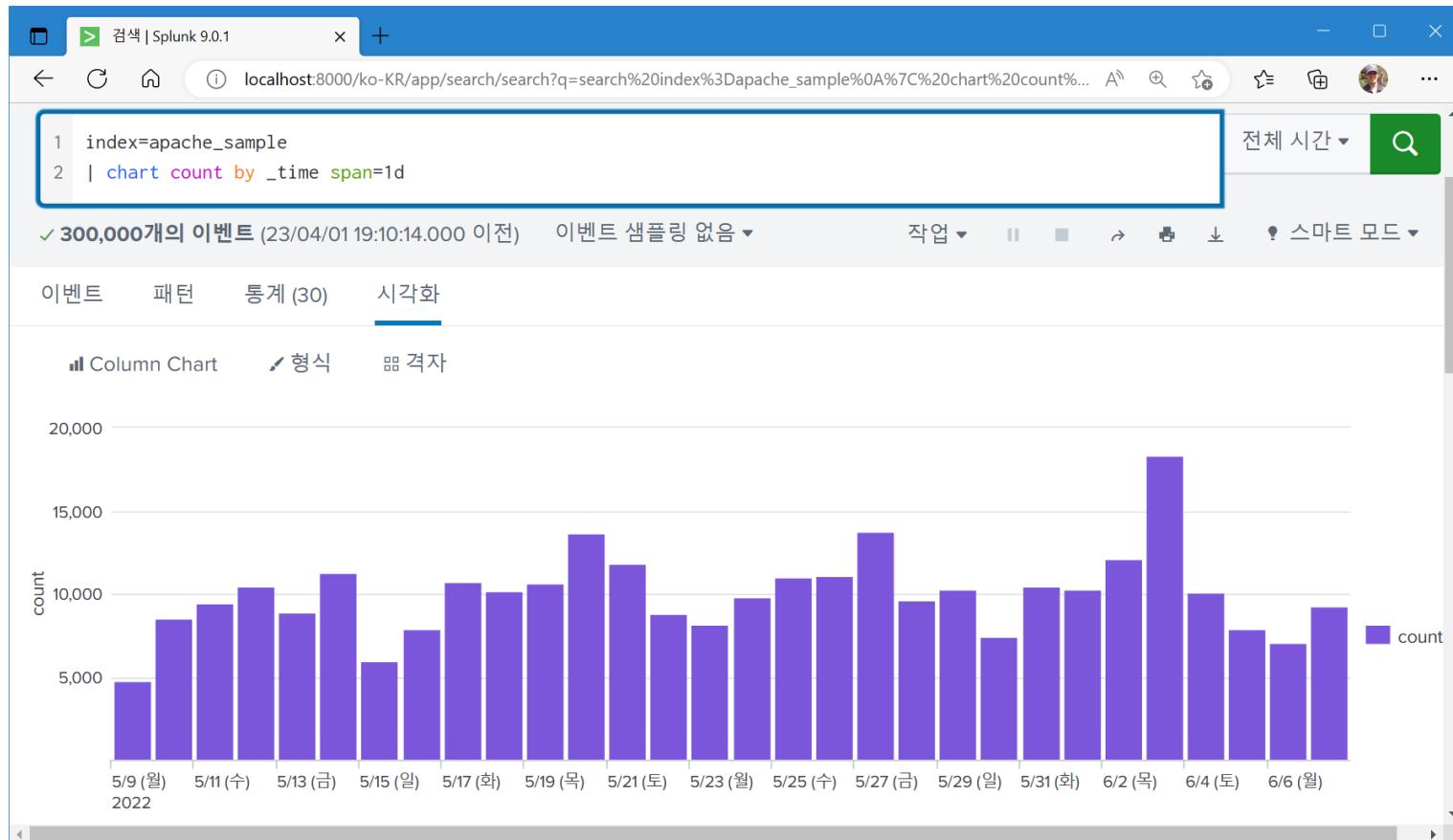


# 스플렁크



## ▶▶ SPL(Search Process Language)

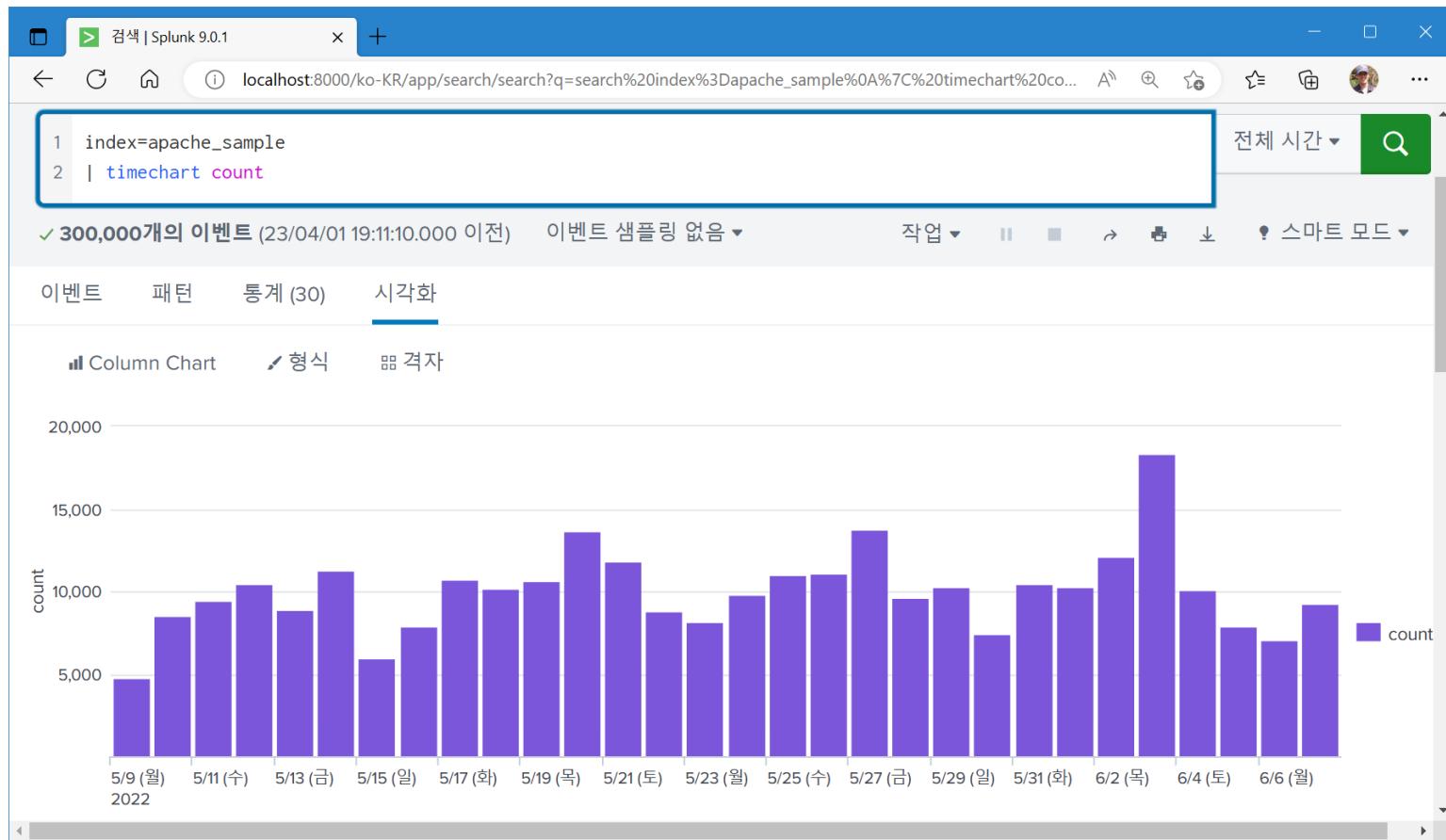
### ▪ 시계열 차트





## ▶▶ SPL(Search Process Language)

### ▪ 시계열 차트





## ▶▶ SPL(Search Process Language)

### ▪ IP 지리정보

검색 | Splunk 9.0.1

localhost:8000/ko-KR/app/search/search?q=search%20index%3Dapache\_sample%0A%7C%20ipl...

```
1 index=apache_sample
2 | iplocation clientip
```

300,000개의 이벤트 (23/04/02 6:05:03.000 이전)      이벤트 샘플링 없음 ▾      작업 ▾      스마트 모드

이벤트 (300,000)      패턴      통계      시각화

시간 표시줄 형식 지정 ▾      - 축소      + 선택 항목 확대/축소      × 선택 취소

리스트 ▾      형식      페이지당 20개 ▾

< 이전      1      2      3      4      5      6      7      8      ...

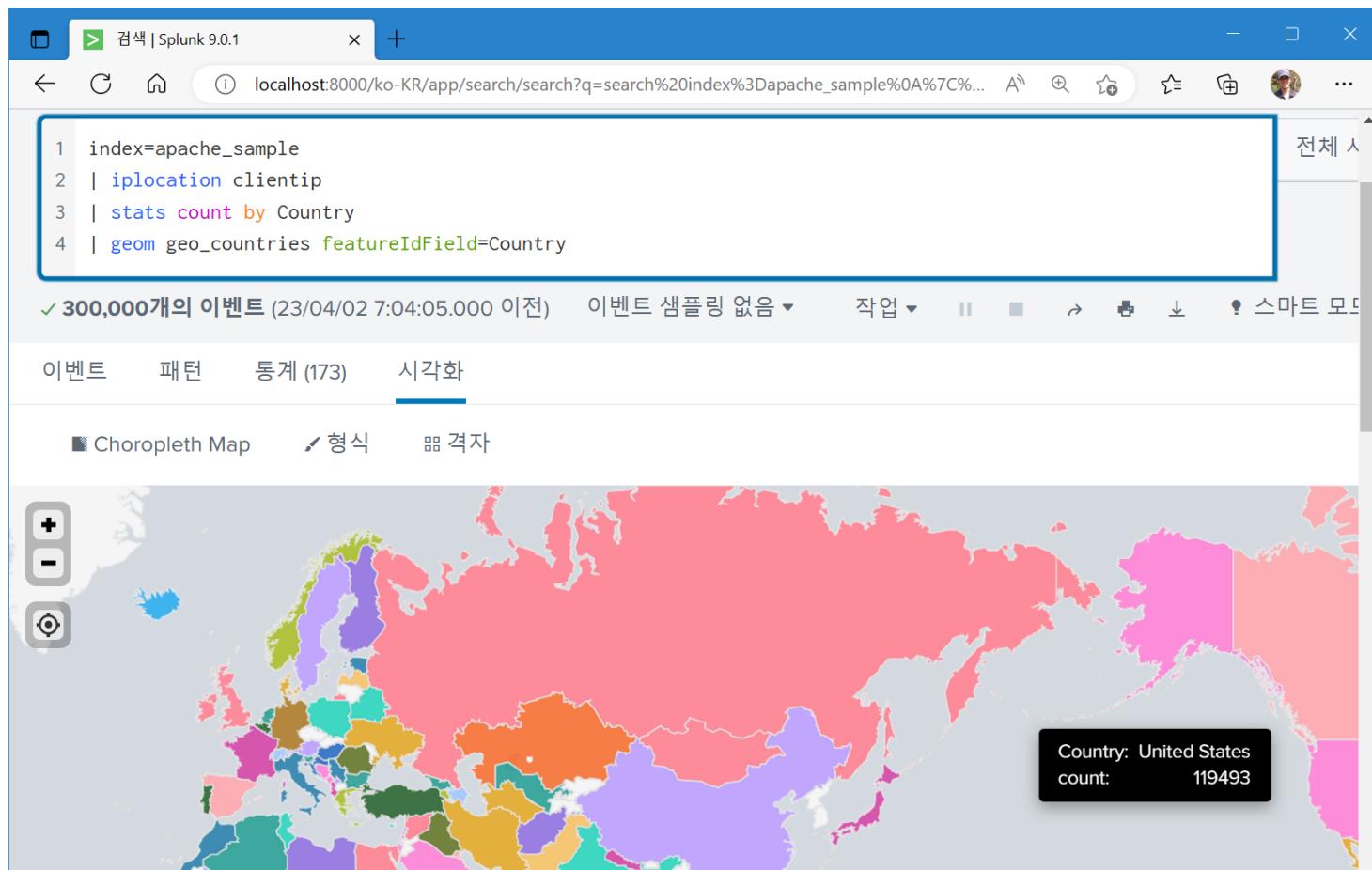
필드	값	시간	이벤트
선택한 필드		> 22/06/07 22:00:00.000	183.60.215.50 - - [07/Jun/2022:13:00:00 +0000] "GET /scripts/server HTTP/1.1" 200 182 "-" "Mozilla/5.0 (compatible; EasouSpider://www.easou.com/search/spider.html)" City = Guangzhou Shi      Country = China      clientip = 183.60.215.50 lat = 23.37900      lon = 113.76300

# 스플렁크



## ▶▶ SPL(Search Process Language)

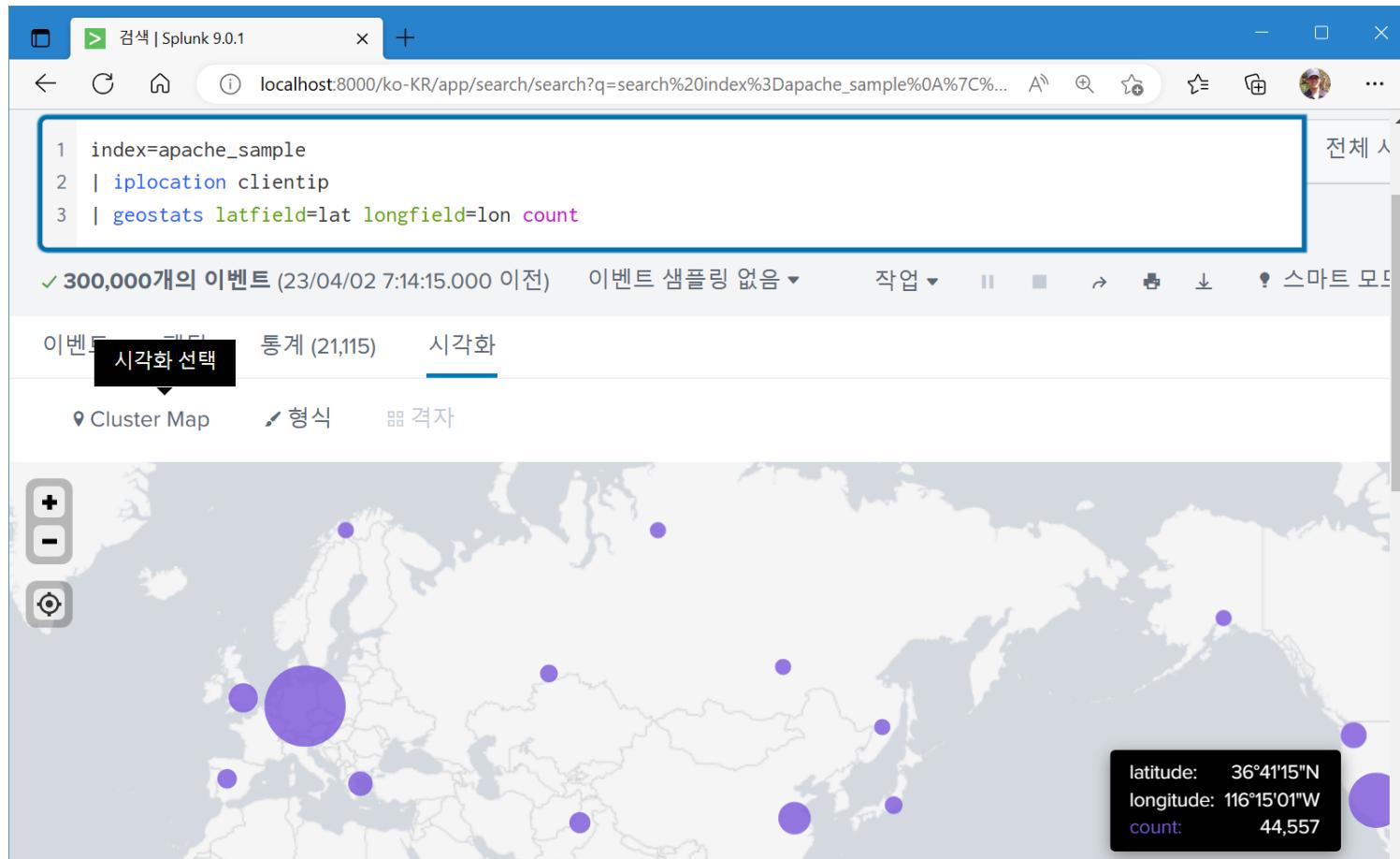
### ▪ IP 지리정보





## ▶▶ SPL(Search Process Language)

### ▪ IP 지리정보





## ▶▶ Splunk DB Connect

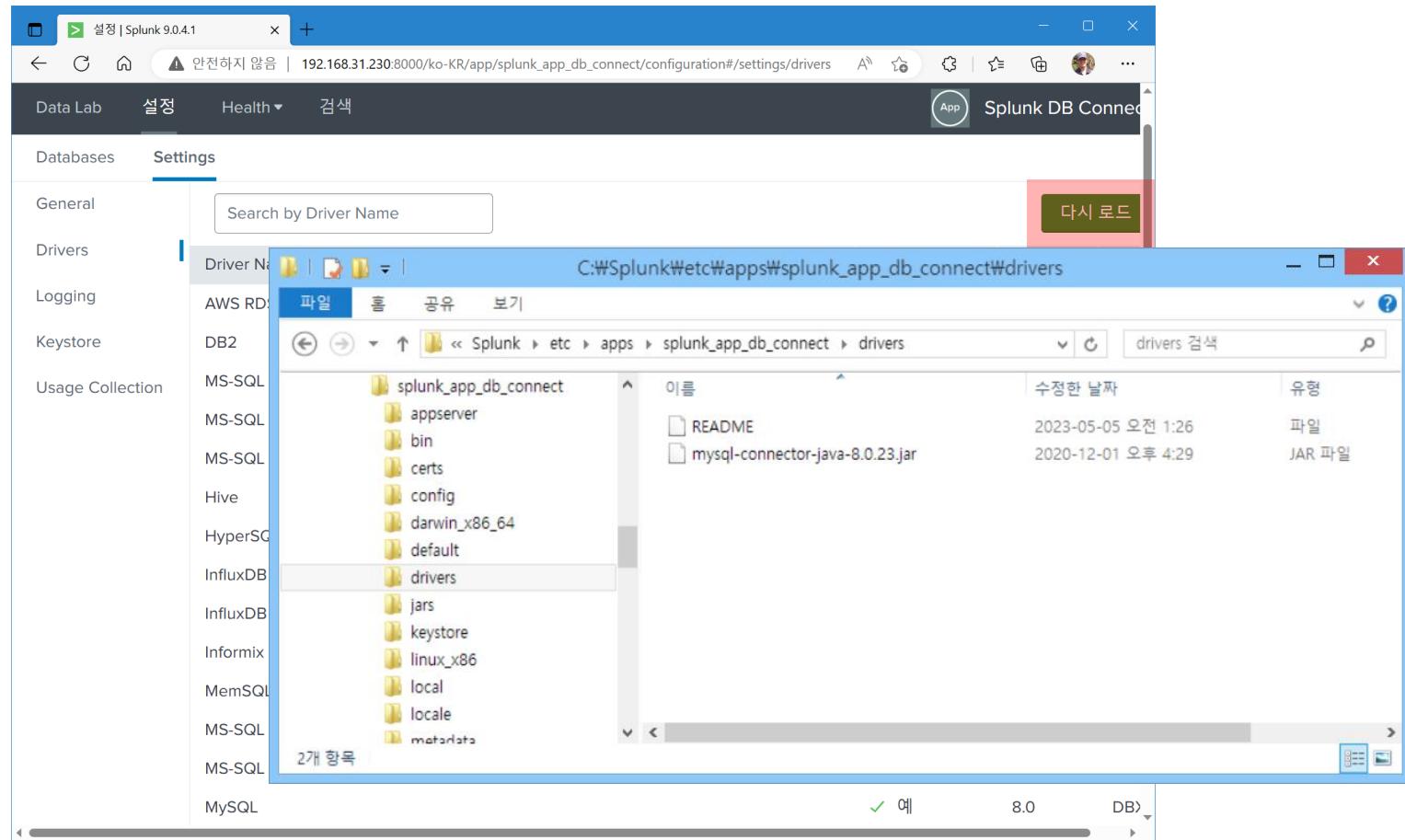
- 자바(v11 이상) 경로 설정

The screenshot shows the Splunk Settings page for DB Connect. The 'Settings' tab is selected. On the left, there's a sidebar with options: General, Drivers, Logging, Keystore, and Usage Collection. The 'General' section is expanded, showing the 'JRE Installation Path (JAVA\_HOME)' field which is highlighted with a red box and contains the value 'C:\jdk-11.0.2\'. Below this, it says 'JRE 11, JRE 17 and JRE 18 are supported.' To the right, there are three main sections: 'Task Server JVM Options' containing the command '-Ddw.server.applicationConnectors[0].port=9998'; 'Query Server' containing the parameter '-Dport=999'; and 'Query Server J parameters'.



## ▶▶ Splunk DB Connect

- mysql connector 드라이버 설치



## ▶▶ Splunk DB Connect

- DB 계정 정보 등록

The screenshot shows the Splunk 9.0.4.1 configuration interface with the 'splunk>enterprise' dashboard. The top navigation bar includes links for Admin, Message, Configuration, Job, and Help, along with a search bar and user profile. The main menu has tabs for Data Lab, Settings (which is selected), Health, and Search. On the right, there's an 'App' icon and a link to 'Splunk DB Connect'.

The main content area is titled 'Databases' and shows a table for 'Identities'. The table has columns for Identity Name, 사용자 이름 (User Name), 앱 (App), 유형 (Type), 상태 (Status), 공유 중 (Shared), and 작업 (Actions). One row is visible: 'snortdb\_user' (User Name), 'root' (App), 'Splunk DB Connect' (Type), 'normal' (Status), a toggle switch labeled '사용 가능' (Enabled), and buttons for '앱' (App), '권한' (Permissions), '편집' (Edit), '복제' (Copy), and '삭제' (Delete). A note above the table states: 'An identity contains the database credentials that Splunk DB Connect uses to access your database.' with a link '자세히 알아보기' (Learn more). A red box highlights the 'New Identity' button at the top right of the table area.



## ▶▶ Splunk DB Connect

- DB 연결 정보 등록

The screenshot shows the Splunk 9.0.4.1 Settings interface. The top navigation bar includes tabs for Data Lab, 설정 (Settings), Health, and 검색 (Search). The main content area is titled "Database Connections". A search bar at the top says "Search by Connection Name". Below it, a table lists one connection: "snortdb\_conn" (Identity: "snortdb\_user", Type: "MySQL", App: "Splunk DB Connect", Status: "사용 가능" - Enabled). A large green button labeled "New Connection" is visible on the right side of the table header. A tooltip above the table explains that a database connection object contains necessary information for connecting to a remote database.

Connection Name	Identity	Connection Type	App	상태	공유 중	작업
snortdb_conn	snortdb_user	MySQL	Splunk DB Connect	사용 가능	앱   권한	편집   복사   삭제

1 connection in total.

# 스플렁크

## ▶▶ Splunk DB Connect

- DB 연동 설정

The screenshot shows the Splunk Data Lab interface for configuring a database input. The top navigation bar includes tabs for Data Lab, 설정 (Configuration), Health, and 검색 (Search). The main content area is titled "Splunk DB Connect" and displays a table of inputs. A red box highlights the "New Input" button in the top right corner of the table header. The table columns include: 이름 (Name), Connection, 템플릿 (Template), 유형 (Type), 인덱스 (Index), Source Type, Frequency, 앱 (App), 상태 (Status), and 작 (Actions). One row is visible, showing "snortdb\_input" as the name, "snortdb\_conn" as the connection, and "event" as the type. The "Source Type" is set to "snortdb". The "Frequency" is set to 5. The "App" column shows "Splunk DB Connect" with a toggle switch turned on. The "Status" column indicates "사용 가능" (Available) with a green checkmark. The "작" (Actions) column contains a "편집" (Edit) link and a "삭제" (Delete) link. At the bottom of the table, it says "1 input in total."

이름	Connection	템플릿	유형	인덱스	Source Type	Frequency	앱	상태	작
snortdb_input	snortdb_conn	-	event	snortdb	snortdb	5	Splunk DB Connect	사용 가능	<a href="#">편집</a> <a href="#">삭제</a>



## ▶▶ Splunk DB Connect

- Input > DB 연동 설정 > Batch 유형(전체 연동)

The screenshot shows the Splunk Data Lab interface for configuring a DB connection. The process is divided into three steps: Set SQL Query, Set Properties, and 완료 (Complete). The 'Set SQL Query' step is currently active, displaying an SQL Editor with the following query:

```

1 select a.cid, a.timestamp, b.sig_name, c
   .data_payload
2 from event a, signature b, data c
3 where a.signature = b.sig_id
4 and a.sid = c.sid and a.cid = c.cid
5 and a.cid > ?
6 order by a.cid

```

The 'SQL Editor' section includes buttons for '형식' (Format) and 'Execute SQL'. To the right, the '설정' (Configuration) section is shown, with the 'Input Mode' set to '이벤트' (Event). The 'Batch' option under '입력 유형' (Input Type) is selected. A callout box labeled '시간 필드 선택' (Select Time Field) points to the 'timestamp' field in the '타임스탬프' (Timestamp) section.

# 스플렁크

## ▶▶ Splunk DB Connect

- Input > DB 연동 설정 > Rising 유형(증분 연동)

The screenshot shows the Splunk Data Lab interface for configuring a DB Connect input. The connection is set to 'snortdb\_conn'. The SQL query is:

```
1 select a.cid, a.timestamp, b.sig_name, unhex(c
   .data_payload)
2 from event a, signature b, data c
3 where a.signature = b.sig_id
4 and a.sid = c.sid and a.cid = c.cid
5 and a.cid > ?
6 order by a.cid
```

The resulting data table is shown below:

	cid	timestamp	sig_name	unhex(c data_payload)
1	5293	2018-10- 10 19:48:00	icmp- request	Non- Displayable Column Type VARBINARY
2	5294	2018-10- 10 19:48:01	icmp- request	Non- Displayable Column Type VARBINARY
3	5295	2018-10- 10 19:48:02	icmp- request	Non- Displayable Column Type VARBINARY

The configuration pane on the right shows the following settings:

- 변경 기준 필드** (Change Criteria Field): Rising Column
- 최초 변경 기준값** (Initial Change Value): cid
- Checkpoint Value**: 0
- 타임스탬프** (Timestamp): timestamp
- 세로막대형** (Horizontal Bar): timestamp



## ▶▶ Splunk DB Connect

### ▪ 16진수 디코딩

The screenshot shows the Splunk search interface with the following details:

- Search Bar:** 검색 | Splunk 9.0.4.1
- URL:** 192.168.31.230:8000/ko-KR/app/search/search?q=search%2...
- Search Command:**

```
index=snortdb  
| eval payload=replace(data_payload, "(..)", "%\1")  
| eval payload=urldecode(payload)
```
- Results Summary:** 12개의 이벤트 (23/05/08 3:00:00.000 ~ 23/05/09 3:09:28.000)
- Event View:** 이벤트 (12) (선택한 필드: payload, sig\_name)  
- 시간: 2023-05-09 01:43:44.000  
- 내용:

```
cid="5315", timestamp="2023-05-09 01:43:44.000", name="web-request", data_payload="474554202F204854542D4167656E743A20576765742F312E313420286C696E75782D63A202A2F2A0D0A486F73743A207777772E62616964752E636F6E3A204B6565702D416C6976650D0A0D0A"  
payload =  
GET / HTTP/1.1 User-Agent: Wget/1.14 (linux-gnu) Accept: */*  
sig_name = web-request
```

# 스플렁크



## ▶▶ Splunk DB Connect

- 16진수 디코딩 (decrypt 앱 설치)

The screenshot shows the Splunk web interface with the following details:

- Search Bar:** index=snortdb | decrypt field=data\_payload unhex() ascii()
- Results Summary:** 8개의 이벤트 (23/05/08 1:00:00.000 ~ 23/05/09 1:19:52.000)
- Event View:** The main pane displays event details. One event is shown in full:

```
2023-05-09 01:00:27.000, cid="5313", timestamp="2023-05-09 01:00:27", si:  
name="icmp-request", data_payload="9B1C596400000000086608000000000010111:  
31415161718191A1B1C1D1E1F202122232425262728292A2B2C2D2E2F30313233345363"  
decrypted = ..Yd....f..... !#$%&'()*+-./01234567  
sig_name = icmp-request | timestamp = 2023-05-09 01:00:27
```
- Left Panel:** Shows fields like <필드 숨기기>, 모든 필드, 선택한 필드, a decrypted 2, a sig\_name 1, a timestamp 8.
- Bottom Navigation:** Includes buttons for 리스트 (List), 형식 (Format), and 페이지당 20개 (20 events per page).



## ▶ 리눅스 환경

- 설치

```
root@Centos7:~]# ls
anaconda-ks.cfg  splunk-8.2.2-87344edfcdb4-linux-2.6-x86_64.rpm
[root@Centos7 ~]#
[root@Centos7 ~]# rpm -i splunk-8.2.2-87344edfcdb4-linux-2.6-x86_64.rpm
warning: splunk-8.2.2-87344edfcdb4-linux-2.6-x86_64.rpm: Header V4 RSA/SHA512 Signature, key ID b3cd4420: NOKEY
useradd: cannot create directory /opt/splunk
complete
[root@Centos7 ~]# whereis splunk
splunk: /opt/splunk/bin/splunk
[root@Centos7 ~]#
```



## ▶ 리눅스 환경

- 실행

```
root@Centos7:/opt/splunk
[root@Centos7 ~]# cd /opt/splunk/
[root@Centos7 splunk]# bin/splunk start --accept-license

This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise, you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.

Please enter an administrator username: root
Password must contain at least:
 * 8 total printable ASCII character(s).
Please enter a new password: █
```

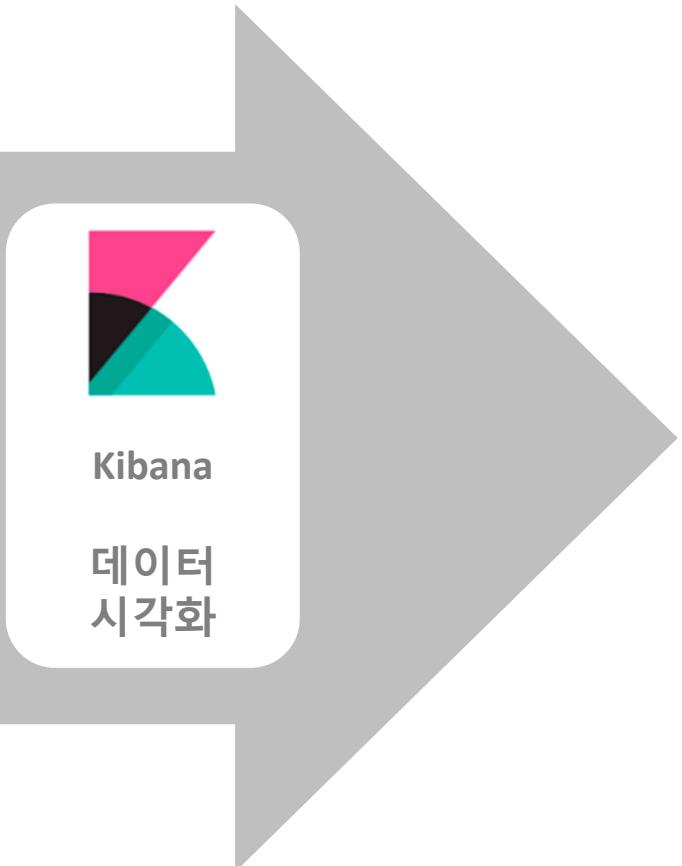
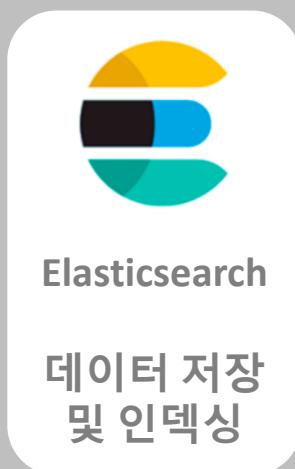
# 엘라스틱



# 엘라스틱

## ▶ 오픈소스 기반의 풀 텍스트 검색 엔진

- 순차적 데이터 처리 구조
- 유연하고 탄력적인 데이터 처리 지원

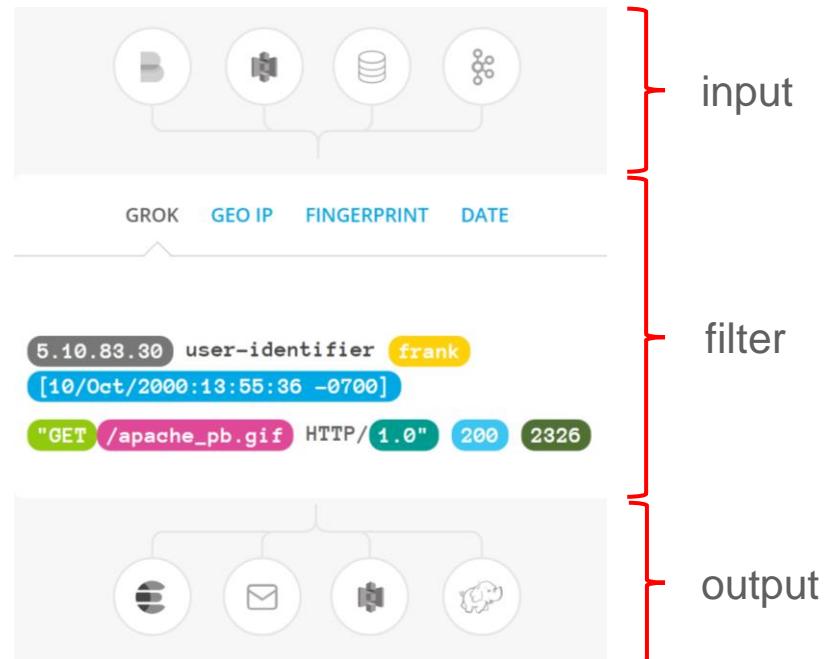


# 엘라스틱

## ▶▶ 로그스타시



- 순차적 데이터 파이프라인 제공
- Jordan Sissel



## ▶▶ 주요 기능

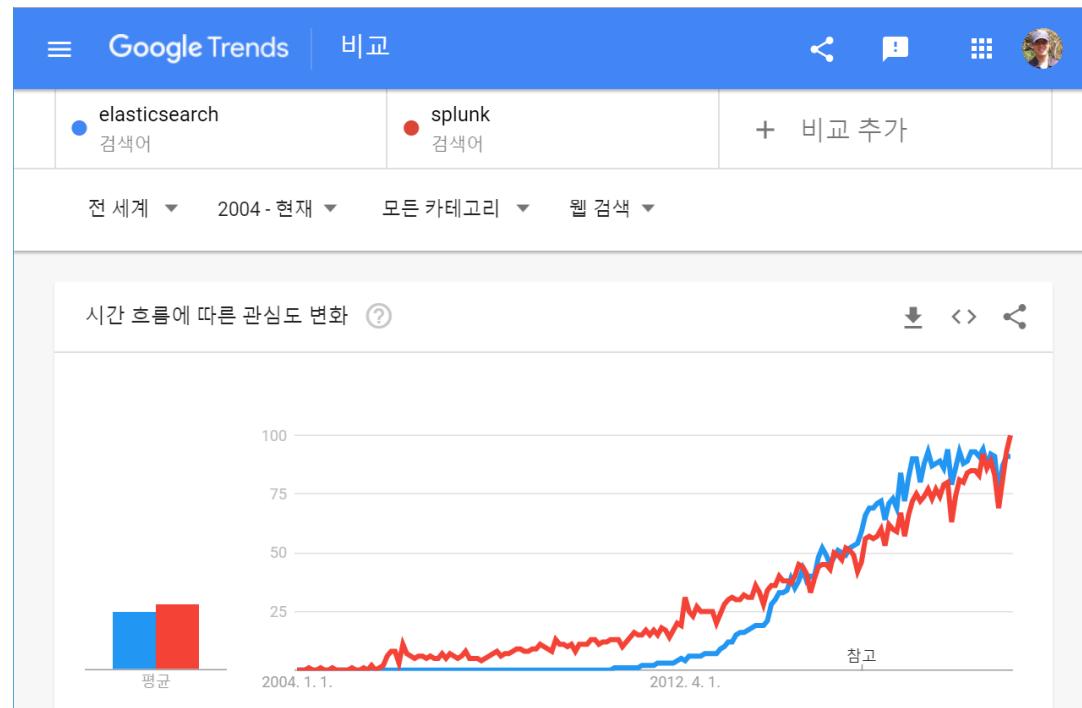
- 다양한 데이터 입/출력 지원
- 다양한 데이터 가공(필터) 지원

# 엘라스틱

## ▶ 엘라스틱서치



- 검색 라이브러리 루신([lucene.apache.org](http://lucene.apache.org)) 기반 검색 엔진
- 2010년 공개(Shay Banon)



## ▶ 주요 기능

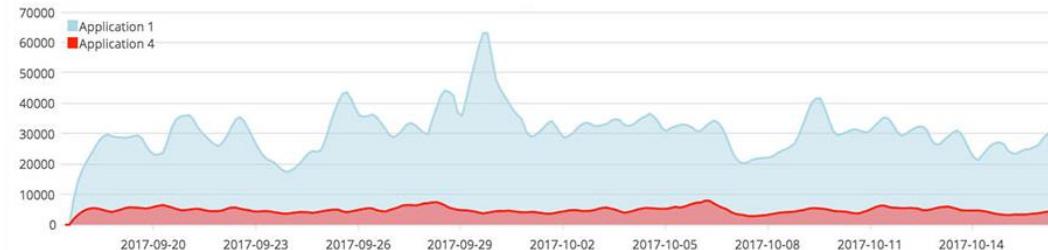
- 데이터 저장 및 인덱싱

# 엘라스틱

## ▶ 키바나



- 웹 기반 데이터 시각화 도구
- Rashid Khan



## ▶ 주요 기능

- 검색, 통계, 대시보드

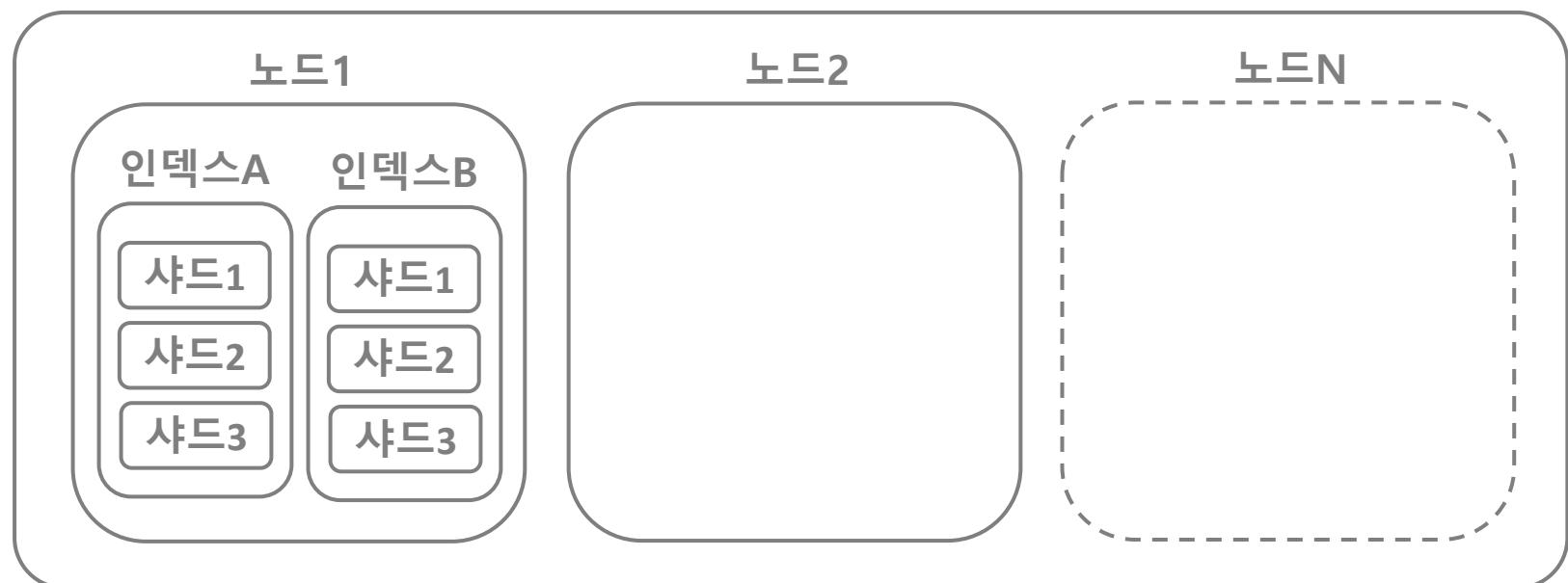
# 엘라스틱



## ▶ 분산 데이터베이스

- 노드 : 엘라스틱서치 실행 단위
- 인덱스 : 논리적 데이터 저장 단위
- 색드 : 인덱스에 대한 논리적 디스크 파티셔닝

클러스터



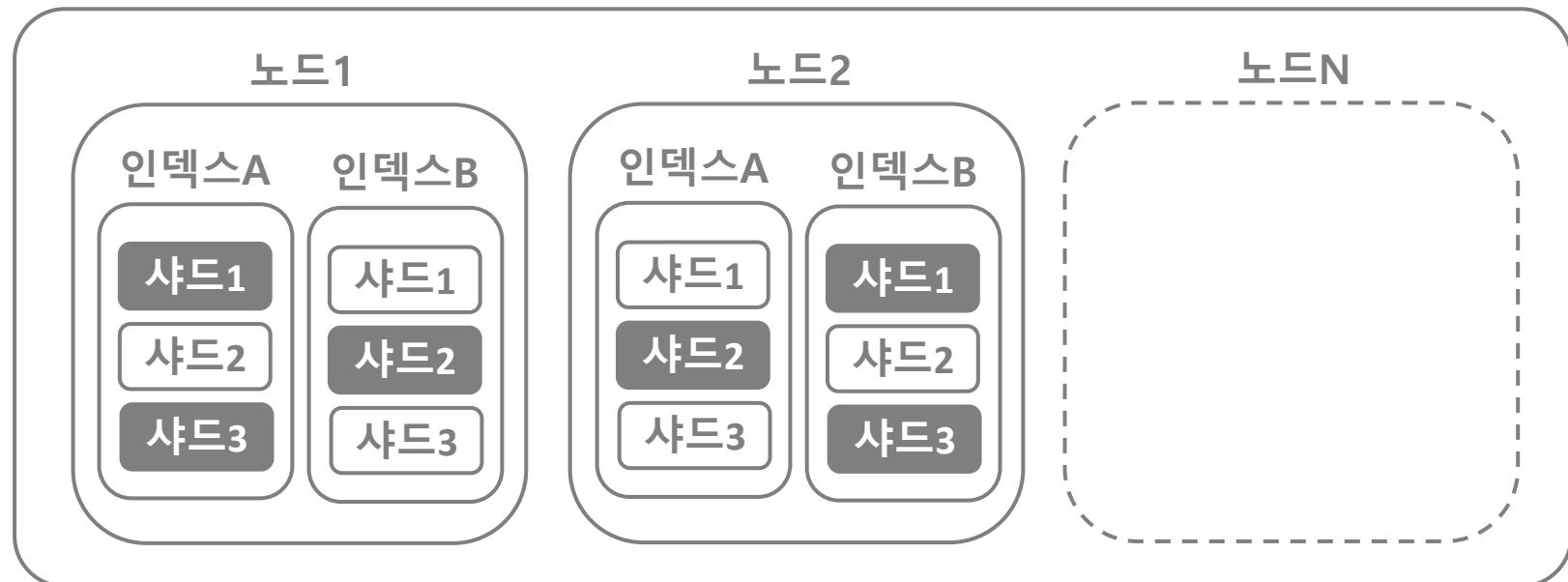
# 엘라스틱



## ▶ 분산 데이터베이스

- 샤드 복제 및 분산을 통해 부하 분산 및 백업 제공

클러스터



# 엘라스틱



## ▶ 데이터 관리 구조

- 엘라스틱서치 VS 관계형 데이터베이스

엘라스틱서치	관계형 DB
인덱스	데이터베이스
필드	테이블
도큐먼트	레코드

mapping      schema

# 엘라스틱



## ▶ 데이터 관리 구조

- 엘라스틱서치

```
{  
    "_index" : "logstash-yyyy.mm.dd",  
    "_type" : "doc",  
    "_id" : "50JOVmgBBgjf nzfQb4-p",  
    "_score" : 4.555061,  
    "_source" : {  
        "성별" : "여자",  
        "이름" : "신사임당",  
        "나이" : "17",  
        "지역" : "강릉",  
        "message" : "여자, 신사임당, 17, 강릉"  
    }  
}
```

- 관계형 데이터베이스

성별	이름	나이	지역
여자	신사임당	17	강릉
남자	홍길동	20	한양
남자	임꺽정	22	황해도



## ▶ 데이터 관리 구조

- Management > Dev Tools (인덱스 생성)

The screenshot shows the Elasticsearch Dev Tools Console interface. The browser title bar reads "Dev Tools - Elastic". The address bar shows the URL "192.168.56.1:5601/app/dev\_tools#/console". The main navigation bar includes "Home", "Console" (which is selected), "Search Profiler", "Grok Debugger", and "Painless Lab (BETA)". On the left, a sidebar menu under "Management" lists "Dev Tools", "Fleet", "Stack Monitoring", "Stack Management", and "Undock navigation". The "Console" tab displays a recent action: "1 PUT testindex". The response details are shown on the right, including a status bar indicating "200 - OK" and "145 ms". The response body text is:

```
enabled. without
authentication, your cluster
could be accessible to anyone
. See https://www.elastic.co
/guide/en/elasticsearch
/reference/7.13/security
-minimal-setup.html to enable
security.

2 {
3   "acknowledged" : true,
4   "shards_acknowledged" : true,
5   "index" : "testindex"
6 }
7 }
```



## ▶ 데이터 관리 구조

- Management > Dev Tools (인덱스 조회)

The screenshot shows the Elasticsearch Dev Tools interface. The browser title bar reads "Dev Tools - Elastic". The address bar shows "안전하지 않음 | 192.168.56.1:5601/app/dev\_tools#/console". The page header includes the elastic logo, a search bar, and navigation links for Home, Console (which is selected), Search Profiler, Grok Debugger, Painless Lab (BETA), History, Settings, and Help.

The left sidebar has a "Recently viewed" section with "Management" selected, and a list of other tools: Dev Tools, Fleet, Stack Monitoring, Stack Management, and Undock navigation.

The main area displays a recent request log:

- Request 1: GET testindex

Details for Request 1:

- Status: 200 - OK
- Time: 14 ms
- Response (highlighted in blue):

```
could be accessible to
anyone. See https://www
.elastic.co/guide/en
/elasticsearch/reference/7
.13/security-minimal-setup
.html to enable security.

2 {
3   "testindex" : {
4     "aliases" : { },
5     "mappings" : { },
6     "settings" : { }
7   }
8 }
```

## ▶ 데이터 관리 구조

- Management > Dev Tools (데이터 입력)

The screenshot shows the Elasticsearch Dev Tools Console interface. The left sidebar has a 'Management' section with 'Dev Tools' selected. The main area shows a 'Console' tab with a history of requests. A recent request is highlighted:

```
1 PUT testindex/_doc/1
2 {
3   "key": "value"
4 }
```

The response is shown on the right:

```
201 - Created    71 ms
{
  "_index" : "testindex",
  "_type" : "_doc",
  "_id" : "1",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 2,
    "successful" : 1,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 1
}
```

# 엘라스틱

## ▶ 데이터 관리 구조

- Management > Dev Tools (데이터 조회)

The screenshot shows the Elasticsearch Dev Tools interface. On the left, there's a sidebar with a navigation menu. The 'Management' section is currently selected, and 'Dev Tools' is highlighted under it. Other options include 'Fleet', 'Stack Monitoring', 'Stack Management', and 'Undock navigation'. The main area is the 'Console' tab, which is active. It has tabs for 'History', 'Settings', and 'Help'. Below these tabs, a search bar says 'Search Elastic'. The console area displays a code editor with a search query and its results. The query is:

```
1 GET testindex/_search
2 {
3   "query": {
4     "match_all": {}
5   }
6 }
```

The results are:

```
17 "hits" : [
18   {
19     "_index" : "testindex",
20     "_type" : "_doc",
21     "_id" : "1",
22     "_score" : 1.0,
23     "_source" : {
24       "key" : "value"
25     }
26   }
27 ]
28
29 ]
```

The status bar at the bottom right indicates a response of '200 - OK' and a duration of '829 ms'.



## ▶ 데이터 관리 구조

- Management > Dev Tools (인덱스 삭제)

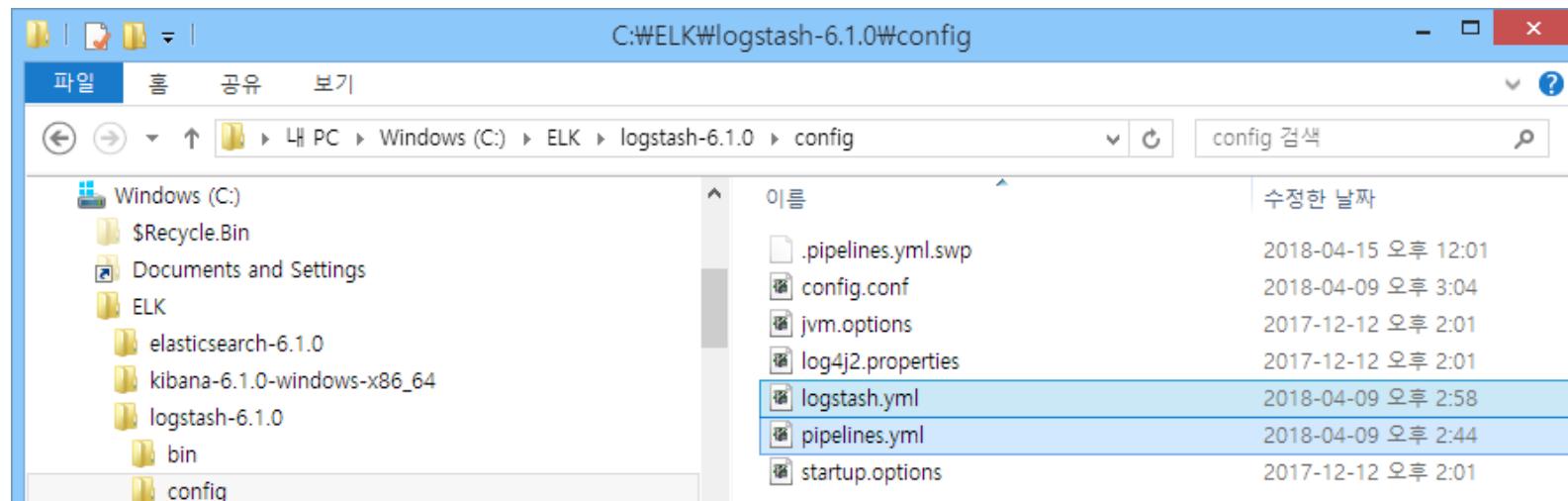
The screenshot shows the Elasticsearch Dev Tools Console interface. On the left, there's a sidebar with 'Management' selected under 'Recently viewed'. Below it are 'Dev Tools', 'Fleet', 'Stack Monitoring', and 'Stack Management'. At the bottom of the sidebar is a 'Undock navigation' option. The main area has tabs for 'Console', 'Search Profiler', 'Grok Debugger', and 'Painless Lab (BETA)'. The 'Console' tab is active, showing a history of requests. The first request is highlighted: 'DELETE testindex'. The response is displayed below, showing a warning about security features being disabled and the index being deleted.

```
DELETE testindex
200 - OK 133 ms
1 #! Elasticsearch built-in
security features are not
enabled. Without authentication
, your cluster could be
accessible to anyone. See https
://www.elastic.co/guide/en
/elasticsearch/reference/7.13
/security-minimal-setup.html to
enable security.
2 {
3
4 }
5 "acknowledged" : true
```



## ▶▶ Logstash를 이용한 데이터 연동

- 설정파일 구성



설정 파일	역할
logstash.yml	로그스태시 프로세스 환경 설정
pipelines.yml	데이터 파이프라인 환경 설정
Test.conf(임의 작성)	데이터 파이프라인(input/filter/output)



## ▶▶ Logstash를 이용한 데이터 연동

- 파이프라인 구조
- <https://www.elastic.co/guide/en/logstash/current/configuration-file-structure.html>

## ▶▶ 데이터 파이프라인

- Input (<https://www.elastic.co/guide/en/logstash/current/input-plugins.html>)
- Filter (<https://www.elastic.co/guide/en/logstash/current/filter-plugins.html>)
- Output (<https://www.elastic.co/guide/en/logstash/current/output-plugins.html>)

# 엘라스틱



## ▶▶ Logstash를 이용한 데이터 연동

- Input/Filter/Output 플러그인 구문

플러그인 {

① 옵션 => “단일 값”

② 옵션 => [ “첫째 값” , “둘째 값” ]

③ 옵션 => { “작업 대상(또는 source)” => “실행 작업(또는 target)” }

}

# 엘라스틱

## ▶▶ Logstash를 이용한 데이터 연동

- [jdbc 인풋 플러그인](#)을 이용한 Snort 데이터베이스 연동

실습



## ▶ 키바나 사이드바 화면 구성

- 데이터 조회를 위해 엘라스틱에 저장된 인덱스 패턴(이름) 등록 과정 실행 필요

The screenshot shows the Kibana Analytics interface. On the left, there is a sidebar menu with the following items:

- Recently viewed**:
  - Analytics**: Overview, Discover (highlighted in red), Dashboard, Canvas, Maps, Machine Learning, Visualize Library (highlighted in red).
  - 원본 로그 조회** (Original Log Search)
  - 대시보드 생성** (Dashboard Generation)
- Maps
- Machine Learning
- Visualize Library (highlighted in red)
- 차트 생성 (Chart Generation)

On the right, there are several cards describing different features:

- Add data**: Import data from various sources.
- Manage**: Manage your dashboards and visualizations.
- Dev tools**: Tools for development and debugging.
- Kibana**: Visualize & analyze data in dashboards. Search and find insights. Design pixel-perfect presentations.



## ▶ 키바나 사이드바 화면 구성

- 데이터 조회를 위해 엘라스틱에 저장된 인덱스 패턴(이름) 등록 과정 실행 필요

Elastic

안전하지 않음 | 192.168.56.1:5601/app/management

Search Elastic

Stack Management

Home

Recently viewed

Observability

Security

Management

Dev Tools 키바나 커맨드라인 툴

Fleet

Stack Monitoring 시스템 모니터링

Stack Management 인덱스 관리

Undock navigation

Data

Index Management 인덱스 조회/삭제 등

Index Lifecycle Policies

Snapshot and Restore

Rollup Jobs

Transforms

Remote Clusters

Alerts and Insights

Rules and Connectors

Reporting

Machine Learning Jobs

Kibana

Index Patterns 데이터 조회를 위한 인덱스 패턴 등록

Welcome to Stack Management  
7.13.4  
Manage your indices, index patterns, saved objects, Kibana settings, and more.  
A complete list of apps is in the menu on the left.

# 엘라스틱

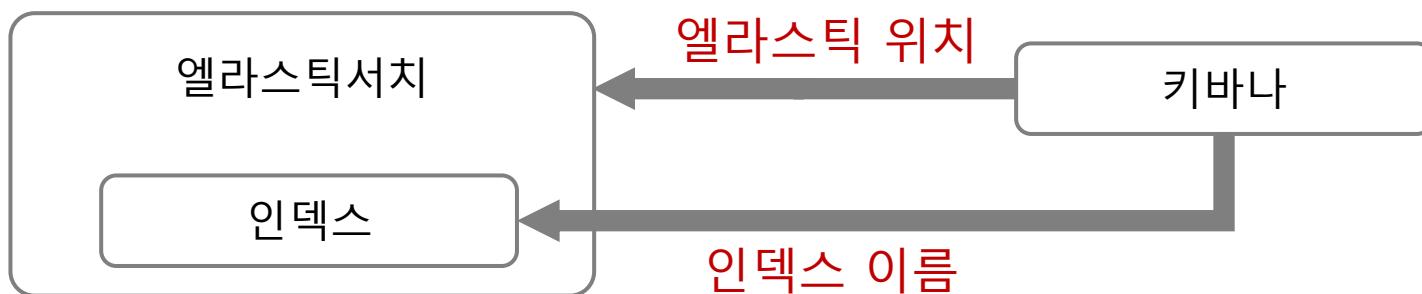


## ▶ 엘라스틱 연결을 위해 키바나에게 필요한 것

- 엘라스틱 위치: kibana.yml > elasticsearch.hosts 옵션

## ▶ 데이터 조회를 위해 키바나에게 필요한 것

- 인덱스 이름: Stack Management > Index Patterns (키바나 사이드바 메뉴)





## ▶▶ Analytics > Discover

### ▪ 원본 로그 조회

The screenshot shows the Elasticsearch Discover interface. At the top, there's a search bar with 'Search Elastic' and a date range selector set to '~ a day ago → now'. Below the search bar, the title 'Discover' is followed by 'filebeat-7.13.4'. A 'Show chart' button is also present. The main area displays '8 hits' with a table header for 'Time' and 'Document'. Two log entries are shown:

Time	Document
May 10, 2023 @ 11:22:53.283	@timestamp: May 10, 2023 @ 11:22:53.283 alert.signature: icmp-request dest_ip: 192.168.31.1 dest_port: 0 payload_printable: m.Zd.....U..... !"#\$%&'()*+,-./01234567 proto: ICMP
May 10, 2023 @ 06:42:55.537	@timestamp: May 10, 2023 @ 06:42:55.537 agent.ephemeral_id: 43491754-392f-492f-95ea-9586cd0181c6 agent.hostname: Snort

On the left side, there's a sidebar with a 'Search field names' input, a 'Filter by type' dropdown (set to 0), and a 'Available fields' section listing fields like '\_id', '\_index', '\_score', '\_type', and '@timestamp'.



## ▶▶ Analytics > Visualize Library

- 차트 생성

The screenshot shows the 'Visualize Library - Elastic' interface. On the left, a sidebar menu includes 'Home', 'Recently viewed', and sections for 'Analytics' (Overview, Discover, Dashboard, Canvas, Maps, Machine Learning, Visualize Library), with 'Visualize Library' currently selected and highlighted in red. The main area displays a list of visualizations with columns for 'Description', 'Tags', and 'Actions'. A large, red-highlighted button labeled '+ Create visualization' is positioned in the upper right of the main area.

Description	Tags	Actions
a		
ge		
t map		



## ▶▶ Analytics > Visualize Library

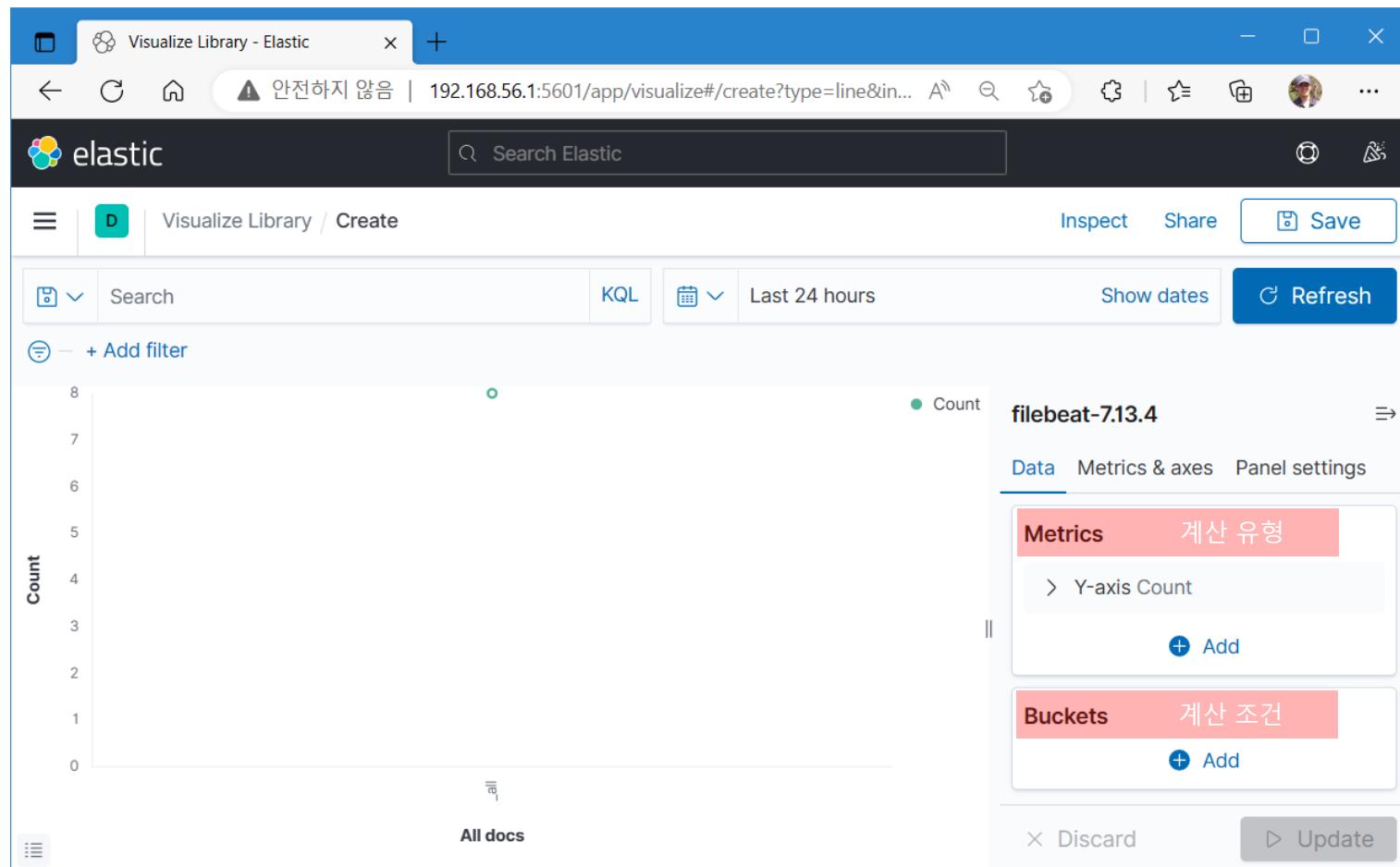
- 차트 생성

The screenshot shows the 'Visualize Library - Elastic' interface. A central modal window titled 'New visualization' is open. Inside, there's a note: 'Create visualizations for your data types at any time. Recommended for most users.' Below this, two options are listed: 'TSVB' (Time Series Visualizer) and 'Custom visualization'. The 'TSVB' option is described as 'Perform advanced analysis of your time series data.' The 'Custom visualization' option is described as 'Use Vega to create new types of visualizations. Requires knowledge of Vega syntax.' At the bottom of the modal, the 'Aggregation based' button is highlighted with a red background, while other options like '[Logs] Goals', 'Gauge', '[Logs] Heatmap', and 'Heat map' are in light blue boxes. To the right of the modal, a sidebar shows sections for 'Actions' and 'Tools', each with a pencil icon. The background shows a blurred view of the library interface with various visualization cards.



## ▶▶ Analytics > Visualize Library

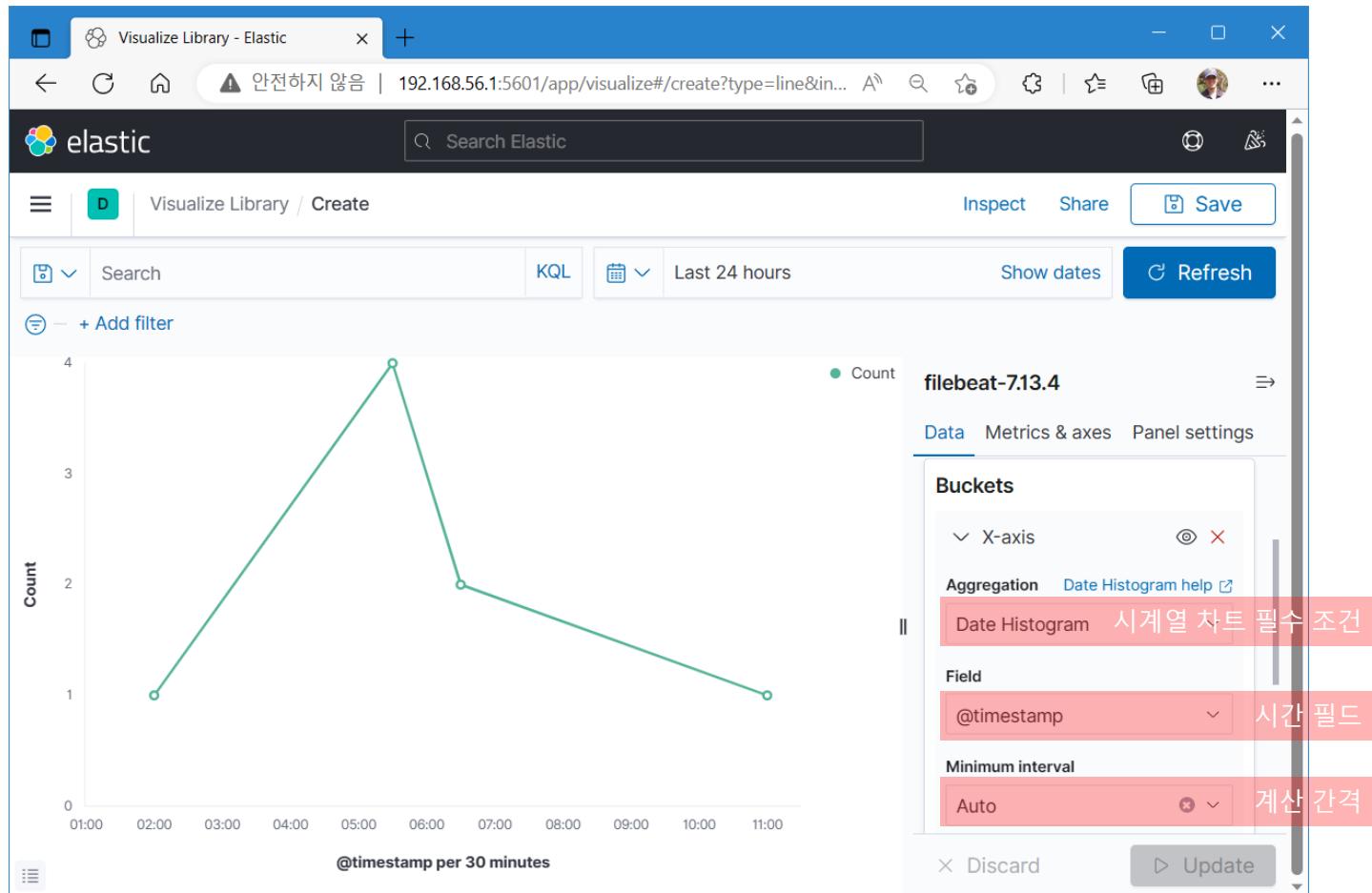
### ▪ 차트 생성





## ▶▶ Analytics > Visualize Library

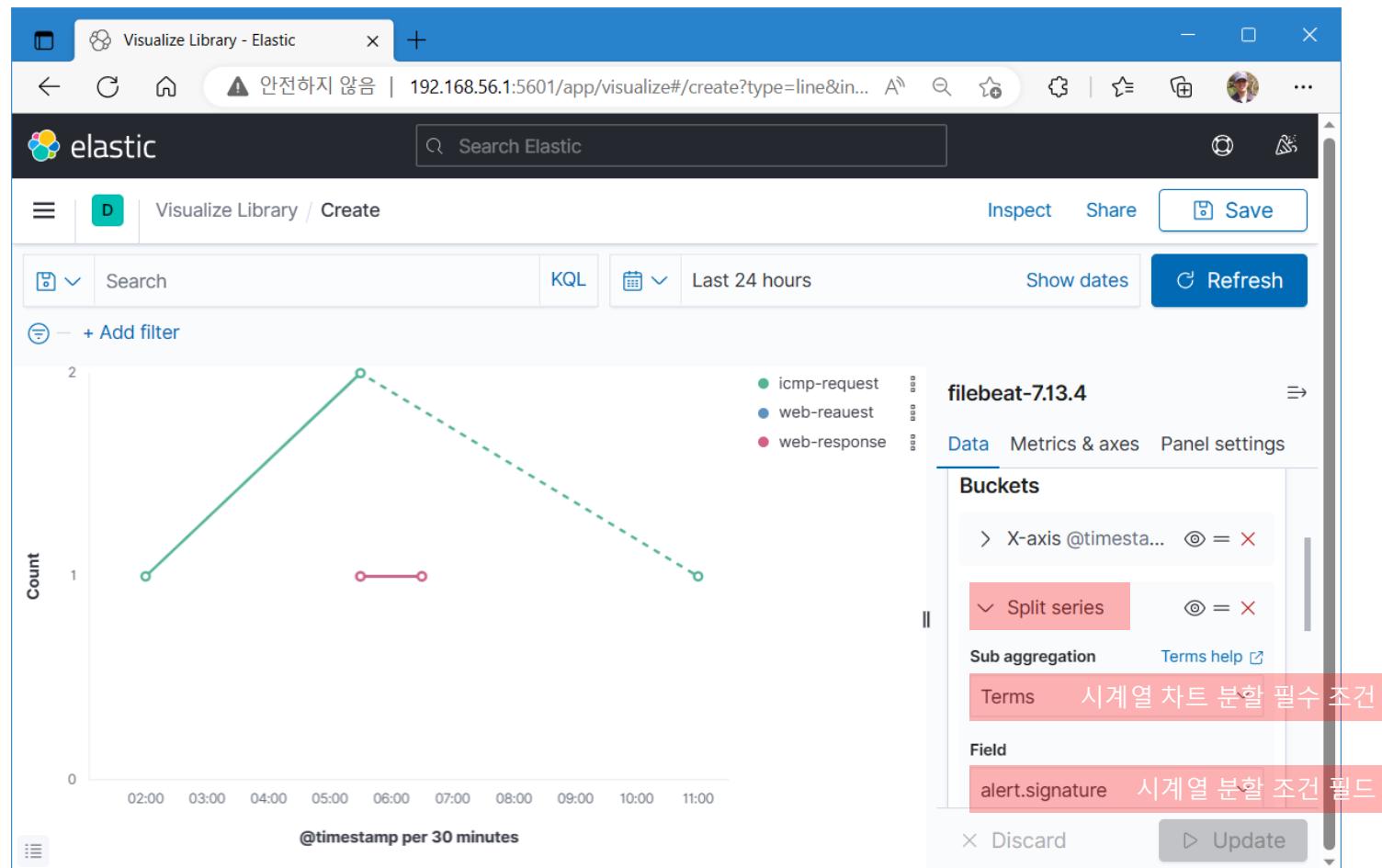
### ▪ 차트 생성





## ▶▶ Analytics > Visualize Library

### ▪ 차트 생성





## ▶▶ Analytics > Visualize Library > Lens

### ▪ 차트 생성

Screenshot of the ElasticLens interface showing a stacked bar chart.

**Left Panel (Search and Filters):**

- Source: filebeat-7.13.4
- Search field names: @timestamp, agent.ephemeral\_id, agent.hostname, agent.id, agent.name, agent.type, agent.version, alert.action, alert.category
- Field filters: 0
- Available fields: @timestamp, agent.ephemeral\_id, agent.hostname, agent.id, agent.name, agent.type, agent.version, alert.action, alert.category

**Middle Panel (Chart View):**

**Visualization Type:** Bar vertical stacked

**Legend:**

- web-response (Red)
- web-request (Blue)
- icmp-request (Green)

**Data:**

Time Period	Count of records
03:00 - 06:00	~4.0
06:00 - 09:00	~2.0
09:00 - 12:00	~1.0

**Right Panel (Configuration):**

- X축:** @timestamp
- Y축:** Count of records
- Break down by:** 시계열 분할 조건 필드 (Top values of alert.signature)
- Buttons:** Reset layer

**Suggestions:**

Current visualization



## ▶▶ Analytics > Maps

### ▪ 지도 차트 생성

The screenshot shows the Elastic Maps Service interface. On the left is a map of Africa with country boundaries. A tooltip window is overlaid on the map, containing the following information:

- LAYERS**: A list of layers, with "dest\_ip" selected.
- winlogbeat-\***: A section showing "Source details".
  - Name: dest\_ip **레이어 이름** (Layer Name)
  - Visibility: Zoom levels 0 → 24
  - Opacity: 75%
- Tooltip fields** **마우스 오버레이 시 표시 정보** (Information to display when mouse hovers):
  - destination.geo.country\_name
  - destination.ip
  - process.name

A red arrow points to the "Add" button in the Tooltip fields section. At the bottom right of the tooltip window are buttons for "Cancel", "Remove layer", and "Save & close".



## ▶▶ Analytics > Maps

### ▪ IP 지리정보 Map

The screenshot shows the Elastic Maps interface. In the top navigation bar, there is a search bar with the text "Search Elastic". Below it, the main title is "Maps / Create". The search bar contains the query "winlog.event\_id:5156" and the time range "Last 24 hours". A "Refresh" button is also present. On the left, there is a map of the world with several colored clusters representing IP locations. A tooltip "Made with NaturalEarth [ ], Elastic Maps Service [ ], OpenMapTiles [ ], OpenStreetMap [ ], contributors [ ] APIA NEW" is visible at the bottom left. On the right, a modal window titled "Add layer" is open. It contains the following fields:

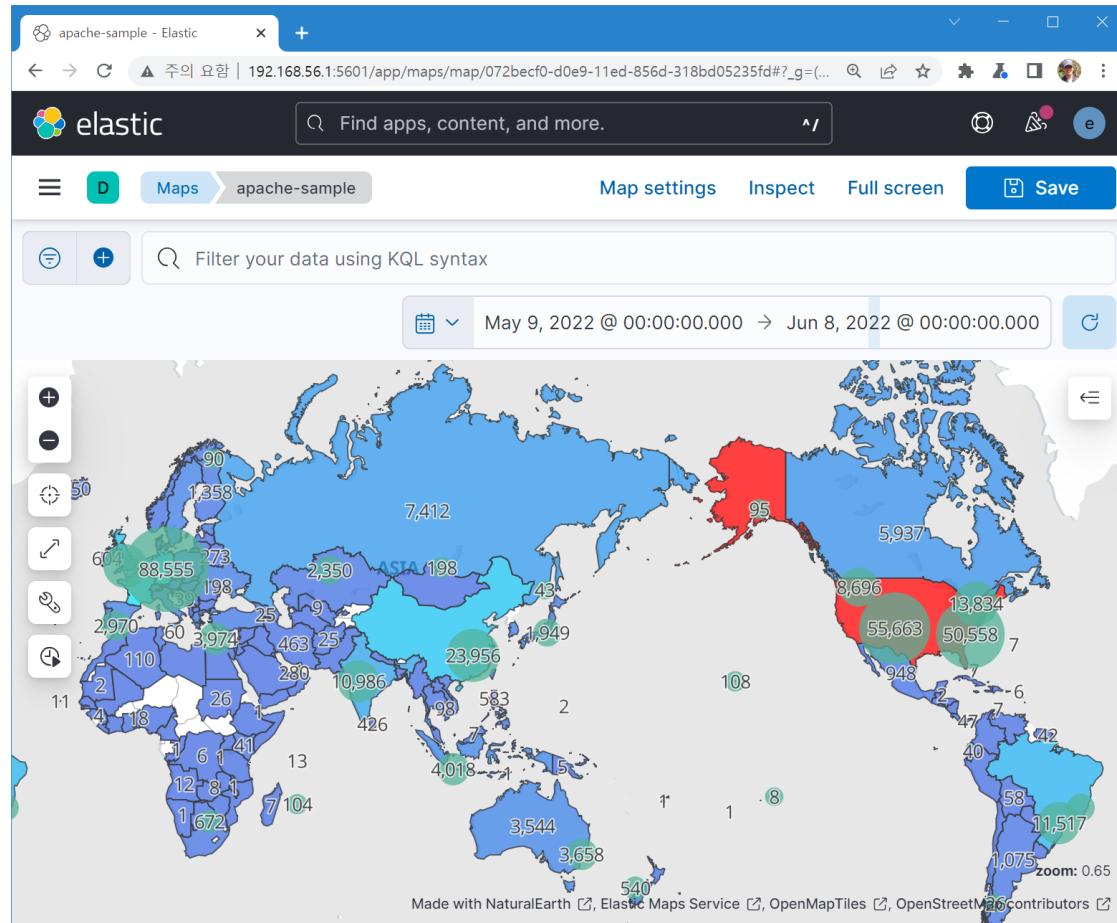
- Layer:** World Countries (선택)
- Join field:** ISO 3166-1 alpha-2 code (지리 구역 코드)
- Statistics source:** Index pattern: winlogbeat-\* (인덱스 선택)
- Join field:** destination.geo.country\_iso\_code (지리 구역 코드 선택)

A "Cancel" button and an "Add layer →" button are at the bottom of the modal.



## ▶▶ Analytics > Maps

### ▪ IP 지리정보 Map





## ▶ 리눅스 환경

- yum 설치를 위한 저장소 설정
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/rpm.html>

```
root@Centos7:~          -  X
1 [elasticsearch-6.x]
2 name=Elasticsearch repository for 6.x packages
3 baseurl=https://artifacts.elastic.co/packages/6.x/yum
4 gpgcheck=1
5 gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
6 enabled=1
7 autorefresh=1
8 type=rpm-md
~
~
<etc/yum.repos.d/elasticsearch.repo" 8L, 226C 1,1      All
```

# 엘라스틱

## ▶ 리눅스 환경

- ELK 스택 설치

The image displays three nested terminal windows, each showing a command being run at a root prompt on a system named 'snort'. The top window shows the command `yum install elasticsearch -y`. The middle window shows `yum install logstash -y`. The bottom window shows `yum install kibana -y`. All three windows have a blue title bar and a black body, with the text of the command highlighted in green.

```
[root@snort ~]# yum install elasticsearch -y
[root@snort ~]# yum install logstash -y
[root@snort ~]# yum install kibana -y
```



## ▶ 리눅스 환경

- ELK 설치 경로 확인

```
root@snort:~ [root@snort ~]# whereis elasticsearch
elasticsearch: /etc/elasticsearch /usr/share/elasticsearch
[root@snort ~]# whereis logstash
logstash: /etc/logstash /usr/share/logstash
[root@snort ~]# whereis kibana
kibana: /etc/kibana /usr/share/kibana
[root@snort ~]#
```