Learning Script - 4
정규표현 활용

보안기술 본부 김윤경

2018.11.26

openbase

The subject comes first, the medium second.

주제가 제일 중요하고 도구는 그 다음이다.

- Richard Prince

스크립트는 단지 하나의 도구입니다.
가장 중요한 것은 원래 하려는 목표를 정확히 이해하고, 관련 지식을 갖춰 적합한 과정으로 수행하는 것입니다.
스크립트는, 할 일의 프로세스가 결정되고 검증된 후, 그 일에 도움을 주려고 사용하는 도구중 하나입니다.

정규표현

정규표현식/정규식/Regular Expression

- 특정한 문자열 집합을 표현하는 형식 언어
- 여러 텍스트 편집기, OS 유틸리티, 프로그래밍 언어 등에서 문자열데이터 **검색, 치환,** 추출을 위해 지원
  - ✓ Snort, Yara, IOC 탐지패턴 표현 옵션
  - ✓ Perl, Python, Rubi 등 스크립트 언어에 내장
  - ✓ Ultra Editor, Notepad++, Visual Studio Code 등 다양한 편집기 검색 옵션 포함

정규표현 - 기본

openbase

설명 자료: Regular Expression Quick Reference

https://neo.dmcs.pl/pios/Regular_Expression_Quick_Reference.pdf

- 정규표현의 basic feature를 용약한 위 자료 페이지로 설명합니다.
- 위 자료에서 "Character Classes", "Repetition", "Anchors" 파트 정도를 익히면 일반적인 정규식 사용이 가능합니다.(추가로 "Options", "Grouping")
- 제가 설명에 이용려고 요약 페이지를 썼지만 잘 설명한 사이트기 많으니, 활용시 정규식의 기본사항을 검색해보기를 권장합니다.

참고자료: 초보자를 위한 정규 표현식 가이드

https://www.slideshare.net/ibare/ss-39274621

- 정규식 기초를 잘 설명하는 참고자료 중 하나입니다.

정규표현 - 실습

데이터

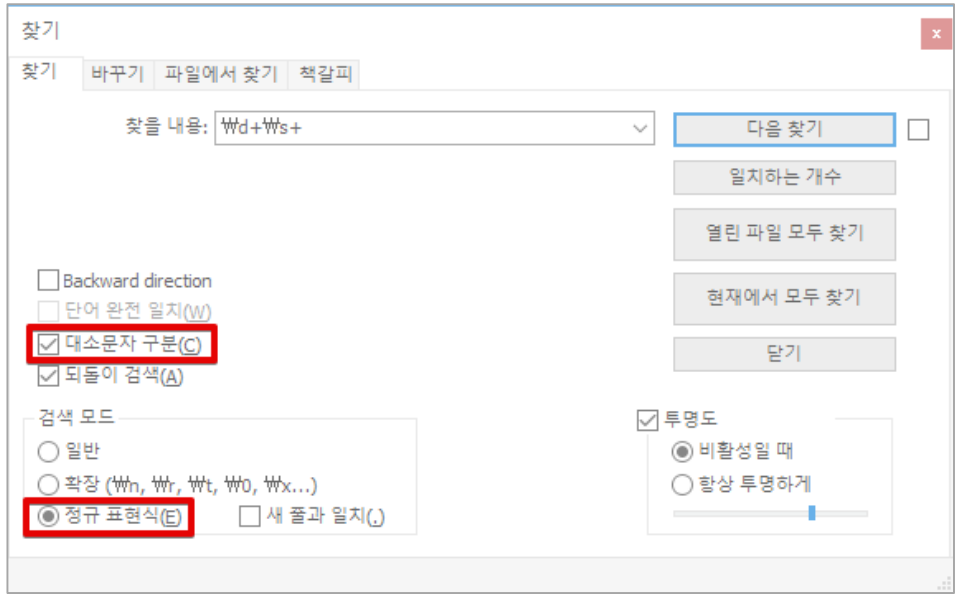
```
a
A
aa
aaa
aaBBBaa
aaaabbccc
AAaaBbb
.....
dot(.)
a(bc)de[fg]
```

정규식

정규식1	정규식2
.	a
...	.{3}
\.{2,3}	
(?s)..	
a{3}	(?i)a{3}
^a.*a$	(?s)^a.*a$
(bc)	\(bc\)

- 왼쪽 박스의 내용을 Notepad++에 넣고 오른쪽 정규식으로 검색해 봅니다.
- 정규식1과 정규식2의 차이를 확인합니다.

※ Notepad++ 정규표현 검색시 체크할 옵션



정규표현 - 실습

openbase

데이터

```
abc
abbbc
abbbbc
```

정규식

```
ab+c
```

위 문자열은 /ab+c/ 정규식으로 매칭시킬 수 있습니다.
아래와 같이 "ac"가 추가되면 어떻게 할까요?

```
ac
abc
abbbc
abbbbc
```

```
ab*c
```

정규표현 - 실습

데이터

```
http://www.daum.net HTTP/1.0
http://some.cloud.net HTTP/1.1
https://www.google.com HTTP/1.1
```

정규식

```
http://.+\sHTTP/1\.[01]
```

잡다한 로그 속에서 위와 같이 http 및 https URL 부분을 골라내려고 합니다. 위 정규식으로 충분할까요?
위 식에서는 https가 빠집니다. 아래와 같이 수정 합니다.

```
(http|https)://.+\sHTTP/1\.[01]
```

이 정규식은 더 간단하게 할 수 있습니다.

```
https?://.+\sHTTP/1\.[01]
```

정규표현 - 실습

데이터: 데이터를 Notepad ++에 넣고 아래의 정규식으로 검색해 봅니다.

```
date=2018-11-23 time=09:11:23 logid="0000000011" srcip=61.82.88.122  dstip=168.126.63.1
date=2018-11-23 time=09:11:23 logid="0000000012" srcip=192.168.2.31  dstip=211.115.106.73
date=2018-11-23 time=09:11:23 logid="0000000013" srcip=61.82.88.6  dstip=95.100.168.67
date=2018-11-23 time=09:11:23 logid="0000000014" srcip=10.0.0.100  dstip=168.126.63.1
date=2018-11-23 time=09:11:23 logid="0000000015" srcip=192.168.2.41  dstip=10.0.10.1
date=2018-11-23 time=09:11:23 logid="0000000016" srcip=192.168.5.1  dstip=172.172.2.21
date=2018-11-23 time=09:11:23 logid="0000000017" srcip=192.168.10.3  dstip=8.8.8.8
date=2018-11-23 time=09:11:23 logid="0000000018" task finished. taskid=901.21.701.000
```

위와 같은 트래픽 로그에서 IP 주소만 찾아내 보안상의 이유로 제거하려고 합니다. IP를 찾는 정규식은 무엇일까요?

```
[\d\.]{7,15}
```

위 식에는 logid, taskid도 매칭됩니다. 부정확하므로 개선이 필요합니다. 아래와 같이 하면 logid는 매칭되지 않습니다.

```
\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}
```

이 정규식은 아래와 같이 간단하게 할 수 있습니다.

```
(\d{1,3}\.){3}\d{1,3}
```

정규표현 - 실습

데이터

```
date=2018-11-23 time=09:11:23 logid="0000000011" srcip=61.82.88.122  dstip=168.126.63.1
date=2018-11-23 time=09:11:23 logid="0000000012" srcip=192.168.2.31  dstip=211.115.106.73
date=2018-11-23 time=09:11:23 logid="0000000013" srcip=61.82.88.6  dstip=95.100.168.67
date=2018-11-23 time=09:11:23 logid="0000000014" srcip=10.0.0.100  dstip=168.126.63.1
date=2018-11-23 time=09:11:23 logid="0000000015" srcip=192.168.2.41  dstip=10.0.10.1
date=2018-11-23 time=09:11:23 logid="0000000016" srcip=192.168.5.1  dstip=172.172.2.21
date=2018-11-23 time=09:11:23 logid="0000000017" srcip=192.168.10.3  dstip=8.8.8.8
date=2018-11-23 time=09:11:23 logid="0000000018" task finished. taskid=901.21.701.000
```

앞서 아래와 같이 IP를 찾는 식을 간단하게 했습니다.

```
(\d{1,3}\.){3}\d{1,3}
```

그렇지만 taskid가 이 정규식에 매칭됩니다. IP 주소에 사용되는 숫자가 3자리인 경우는 1또는 2로만 시작하므로 다음과 같이 정의하면 taskid가 매칭되지 않습니다.

```
(([1,2]\d{2}|\d{1,2})\.){3}([1,2]\d{2}|\d{1,2})
```

IP 주소의 숫자 범위는 1~255 입니다. 이 조건을 적용한 정규식은 아래와 같습니다. 더 정확해 졌습니다.

```
((2[0-4]\d|25[0-5]|1\d\d|\d{1,2})\.){3}(2[0-4]\d|25[0-5]|1\d\d|\d{1,2})
```

정규표현 - 실습

openbase

데이터

```
date=2018-11-23 time=09:11:23 logid="0000000011" srcip=61.82.88.122  dstip=168.126.63.1
date=2018-11-23 time=09:11:23 logid="0000000012" srcip=192.168.2.31  dstip=211.115.106.73
date=2018-11-23 time=09:11:23 logid="0000000013" srcip=61.82.88.6  dstip=95.100.168.67
date=2018-11-23 time=09:11:23 logid="0000000014" srcip=10.0.0.100  dstip=168.126.63.1
date=2018-11-23 time=09:11:23 logid="0000000015" srcip=192.168.2.41  dstip=10.0.10.1
date=2018-11-23 time=09:11:23 logid="0000000016" srcip=192.168.5.1  dstip=172.172.2.21
date=2018-11-23 time=09:11:23 logid="0000000017" srcip=192.168.10.3  dstip=8.8.8.8
date=2018-11-23 time=09:11:23 logid="0000000018" task finished. taskid=901.21.701.000
```

여전히 IP 주소와 같은 형식을 가진 taskid가 매칭될 수 있습니다. taskid가 검색에 포함되는 것을 방지하기 위해 다음과 같이 정의할 수도 있습니다.
숫자와 dot(.) 만으로 구성된 길이 7~15 짜리 문자열 앞에 'srcip=' 또는 'dstip='이 와야 한다는 조건을 붙이는 정규식은 아래와 같습니다.
이것을 후방탐색(Look-behind Assertion)이라고 합니다.

```
(?<=(srcip=|dstip=))[\d\.]{7,15}
```

정규표현 - 실습

데이터

```
bag: $23.45
note: $5.31
chair: $899.00
pencil: $69.96
Total items: 4
today’s date: 11.26
```

정규식

```
(?<=\$)[\d\.]+
```

위와 같은 메모지에서 물건값(주황색 달러가격 부분)만 검색하려면 어떻게 할까요?
달러 표시 뒤의 숫자를 찾으면 됩니다.
이럴 때 앞서 IP 주소 예제와 같이 후방탐색을 사용합니다.

정규표현 - 활용예

openbase

Snort rule

```
1000082 alert tcp any any -> any 143 (msg:"UDS_089_EXPLOIT MDAEMON overflow_080825_01";
flow:established,to_server; content:"FLAGS BODY"; pcre:"/[0-9a-zA-Z]{200,}/R"; content:"|EB 06 90 90 8b 11 DC 64
90|"; distance:0; classtype:successful-user; reference:url,www.milw0rm.com/exploits/5248; reference:bugtraq,28245;
sid:89;)
```

Yara rule: MD5 해쉬 표현 및 기타

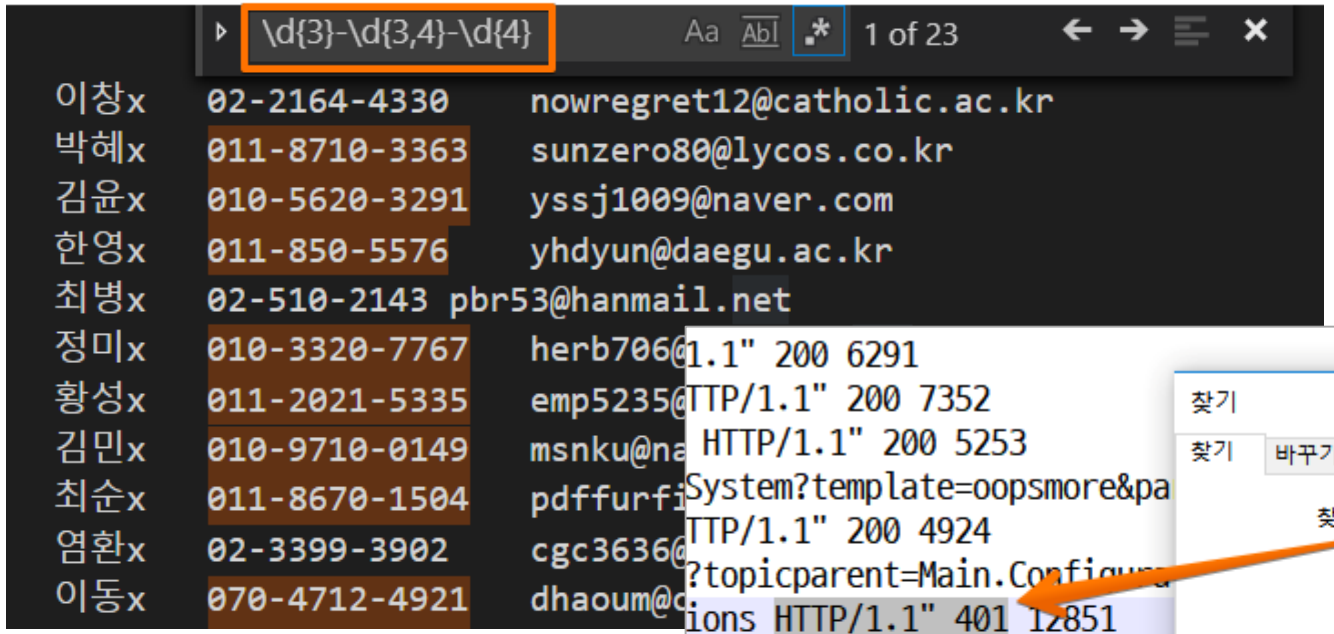
```
rule RegularShow
{
    strings:
        $re1 = /md5: [0-9a-fA-F]{32}/
        $re2 = /state: (on|off)/

    condition:
        $re1 and $re2
}
```

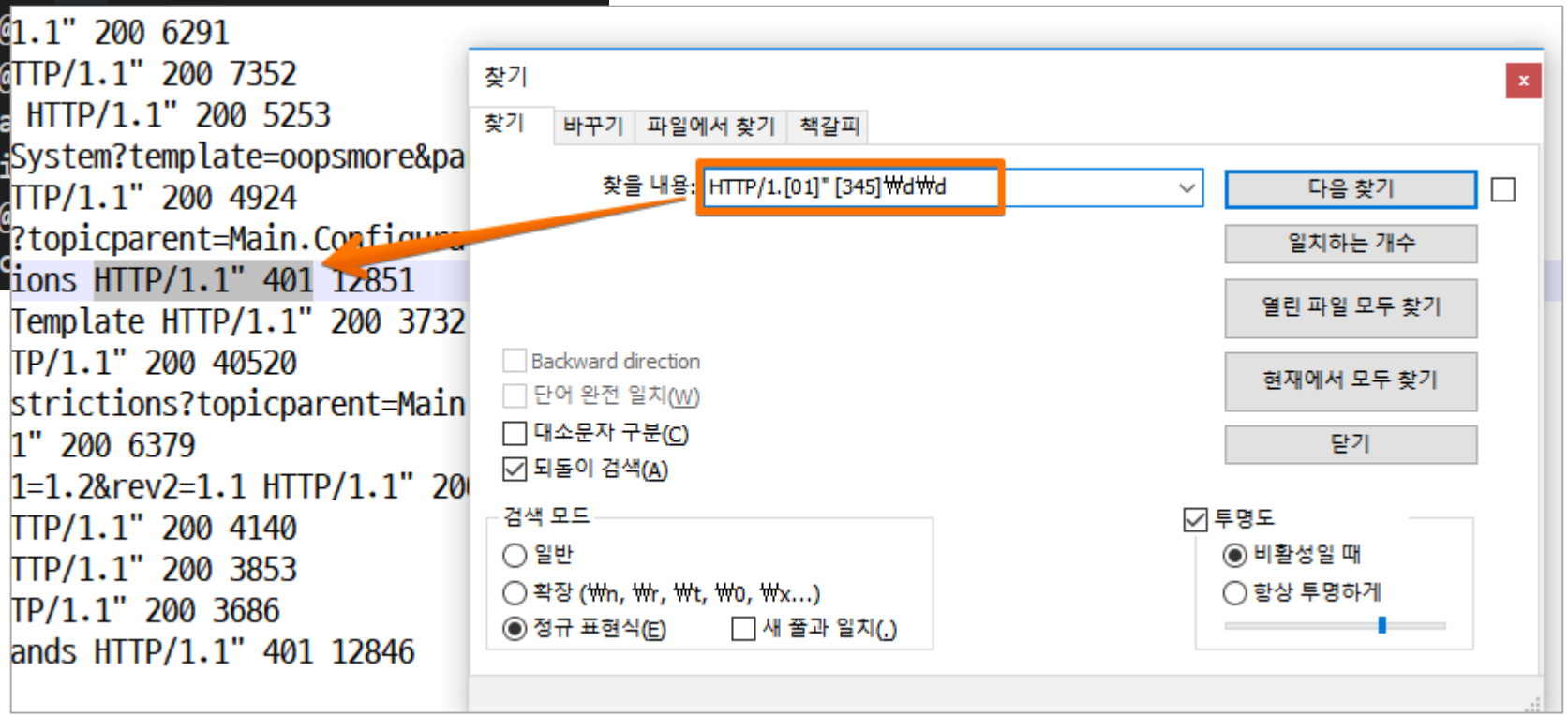
정규표현 - 활용예

openbase

Visual Studio Code 정규표현



Notepad++ 정규표현



자료 :
https://secuwave.github.io/secure3/learn_script/main

openbase