1 Splunk Tutorial 2: Regular Expressions

1.1 Regular Expressions

Using the text below and the following Web site:

http://regex101.com/

Link: https://regex101.com/r/r702PY/2

```
There is not much we can do apart from contacting There is not much we can do apart from contacting f.smith@home.net to see if he would like to reboot the server at 192.168.0.1. If he can do this then I will call him on 444.3212.5431. My credit card details are 4321-4444-5412-2310 and 5430-5411-4333-5123 and my name on the card is Fred Smith. I really like the name domain fred@home. Overall our target areas are SW1 7AF and EH105DT. I tested the server last night, and I think the IP address is 10.0.0.1 and there are two MAC addresses which are 01:23:45:67:89:ab or it might be 00.11.22.33.44.55.

The book we will use is At  Home  and it can be bought on amazon.com or google.com, if you search for 978-1-4302-1998-9. My password is:

a1b2c3
Best regards, Bert.
EH14 1DJ
+44 (960) 000 00 00 1/1/2009
```

Now see if you can detect:

1. Email addresses.
2. IP addresses.
3. Telephone numbers (US style).
4. UK post codes.
5. Credit card details.

1.1.1 Possible Answers

```
Email address:
[a-zA-Z0-9._%+-]+@[a-zA-Z0-9._%+-]

```

```
([a-zA-Z0-9_\-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-
9]{1,3}\.)|(([a-zA-Z0-9\-]+\.)+))([a-zA-Z]{2,4}|[0-
9]{1,3})(\]?)

IP:
[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}.[0-9]{1,3}

Telephone:
\d{3}[-.]?\d{3}[-.]?\d{4}

UK Post code:
[A-Z]{1,2}[0-9]{1,2}[A-Z]?\s[0-9][A-Z][A-Z]

[A-Z]{1,2}[0-9]{1,2}[A-Z]?\s?[0-9][A-Z][A-Z]

?(([BEGLMNSWbeglmnsw][0-9][0-9]?)|(([A-PR-UWYZa-pr-uwyz][A-HK-
Ya-hk-y][0-9][0-9]?)|(([ENWenw][0-9][A-HJKSTUWa-
hjkstuw])|([ENWenw][A-HK-Ya-hk-y][0- 9][ABEHMNPRVWXYabehmnprvwxy]))))
    ?[0-9][ABD-HJLNP-UW-Zabd-
hjlnp-uw-z]{2}

Credit card (Visa):
4\d{3}(\s|-)?\d{4}(\s|-)?\d{4}(\s|-)?\d{4}
[45]\d{3}(\s|-)?\d{4}(\s|-)?\d{4}(\s|-)?\d{4}

Domain name:
[a-zA-Z\.]+\.(com|net|uk)
[a-zA-Z0-9\-\.]+\.(com|org|net|mil|edu|COM|ORG|NET|MIL|EDU|UK)

MAC
([0-9a-fA-F][0-9a-fA-F]:){5}([0-9a-fA-F][0-9a-fA-F])

([0-9a-fA-F][0-9a-fA-F][:.]){5}([0-9a-fA-F][0-9a-fA-F])

Password:
(?=.*[0-9]+.*)(?=.*[a-zA-Z]+.*)[0-9a-zA-Z]{6,}
```

1.2 Splunk Regular Expression Searches

Using Splunk at https://asecuritysite.com:8000 determine the following.

We can use regular expressions to find information. For example, to find the number of accesses from an IP address which starts with “182.”, we can use:

```
get | regex _raw="182\.\d{1,3}\.\d{1,3}\.\d{1,3}"
```

Determine the number of accesses for GET from any address which begins with 182:

The security team search for an address that is ending with .22, and do a search with:

```
get | regex _raw="\d{1,3}.\d{1,3}.\d{1,3}.22"
```

But it picks up logs which do not include addresses with .22 at the end. What is the problem with the request, and how would you modify the request:

You are told that there's accesses to a file which ends in "a.html". Using a regular expression, such as:

```
get | regex _raw="[a]+\.html"
```

Outline three HTML files which end with the characters 'a', or an 'e', and have '.html' as an extension:

A simple domain name check is:

```
get | regex \_raw="[a-zA-Z\.]+\.(com|net|uk)"
```

If we now try:

```
get | regex _raw="[a-zA-Z0-9\-\.]+\.(com|org|net|mil|edu|COM|ORG|NET|MIL|
    EDU|UK)"
```

we will return events with domain names.

Outline which ones have been added:

We can search for email addresses with:

```
get | regex _raw="(?<email>[\w\d\.\-]+\@[\w\d\.]+)"
```

Which email addresses are present:

We can search for times using regular expressions, such as:

```
get | regex _raw="[0-9]{2}\:22\:[0-9]{2}"
```

How many GET requests where there at 22 minutes past the hour:

How many GET requests were made at 14 seconds past the minute: