Prashant Lakhera Follow
Apr 30, 2019 · 3 min read · Listen

100 Days of DevOps — Day 79-Apache Log Parser Using Python

To view the updated DevOps course(101DaysofDevOps)

Course Registration link: https://www.101daysofdevops.com/register/

Course Link: https://www.101daysofdevops.com/courses/101-days-of-devops/

YouTube link: https://www.youtube.com/user/laprashant/videos

Welcome to Day 79 of 100 Days of DevOps, Focus for today is Apache Log Parser Using Python

The aim of this tutorial is to create Apache log parser which is really helpful in determine offending IP addresses during the DDoS attack on your website. This is what we are going to do

- *Read Apache log file(access.log)*
- *Count quantity of requests to your website from each IP address*

If you look at the content of access.log this is how it looks

```
192.168.0.1 -- [23/Apr/2017:05:54:36 -0400] "GET / HTTP/1.1" 403 3985
"-" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/58.0.2661.102 Safari/537.36"
```

Let's break down these fields

```
192.168.0.1 --> IP address
[23/Apr/2017:05:54:36 -0400] --> Date,time and timezone
GET / HTTP/1.1 --> HTTP get request to the page
403 --> Server response code
3985 --> Number of byte transferred
Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/58.0.2661.102 Safari/537.36" --> Finally there are data
of user's hardware,OS and browser
```

But the piece we are interested in is IP address

So as mentioned above our first step is

- *Read the Apache log file*

```
def apache_log_reader(logfile):

# We are saying opened file to the f variable, where f is a reference
to the file object
    with open(logfile) as f:
        log = f.read()
        print(log)

# Create entry point of our code
if __name__ == "__main__":
    apache_log_reader("access_log")
```

Now let's go to the second part

- *Count IP address*

```
import re
from collections import Counter

def apache_log_reader(logfile):
    myregex = r'(\d{3}).(\d{3}).(\d{3}).(\d{3})'

    with open(logfile) as f:
        log = f.read()
        my_list = re.findall(myregex,log)
        my_counter = Counter(my_list)
        for k, v in(my_counter.items()):
            print("IP Address " + "=> " + str(k) + " " + "Count "  +
"=> " + str(v))

# Create entry point of our code
if __name__ == "__main__":
    apache_log_reader("access_log")
```

Let's break this code

- *As we don't need the entire entry we need to do some pattern search and that we can do with the help of regular expression*
- *We imported the re module and then write the regular expression matching pattern, where*

```
\d: Any numeric digit(0-9)
```

For more info please refer to

Python Regular Expression
Regular expressions can be think like a mini-language for specifying text pattern
medium.com

```
r'(\d{3}).(\d{3}).(\d{3}).(\d{3})'
```

- *r stands for raw string*
- *Then we are using the collection module*

Python Collections Module
This module implements specialized container datatypes providing alternatives to Python's general purpose built-in...
medium.com

Output

```
IP Address => 192.168.0.2 Count => 1
IP Address => 192.168.0.3 Count => 1
IP Address => 192.168.0.4 Count => 1
IP Address => 192.168.0.5 Count => 1
IP Address => 192.168.0.6 Count => 1
```

This is not the only way to write this code, there is a much better way to write the same piece of code, so stay tuned :-)

Now the better way to write the apache log parser

```
#!/usr/bin/env python
"""
USAGE:
logparsing_apache.py apache_log_file
This script takes apache log file as an argument and then generates a report, with host
bytes transferred and status
"""
import sys
def apache_output(line):
    split_line = line.split()
    return {'remote_host': split_line[0],
            'apache_status': split_line[8],
            'data_transfer': split_line[9],
            }


def final_report(logfile):
    for line in logfile:
        line_dict = apache_output(line)
        print(line_dict)


if __name__ == "__main__":
    if len(sys.argv) > 1:
        infile = open(sys.argv[1], 'r')
    else:
        print ("You must specify a valid file to parse")
        print (__doc__)
        sys.exit(1)
    log_report = final_report(infile)
    print (log_report)
    infile.close()
```

apache_log_parser.py hosted with ❤ by GitHub view raw

GitHub Link

100daysofdevops/100daysofdevops
Contribute to 100daysofdevops/100daysofdevops development by creating an account on GitHub.
github.com

Looking forward from you guys to join this journey and spend a minimum an hour every day for the next 100 days on DevOps work and post your progress using any of the below medium.

- *Twitter: @100daysofdevops OR @lakhera2015*
- *Facebook: https://www.facebook.com/groups/1953826398038645/*
- *Medium: https://medium.com/@devopslearning*
- *Slack: https://devops-myworld.slack.com/messages/CF41EPG49/*
- *GitHub Link-https://github.com/100daysofdevops*

Reference

100 Days of DevOps — Day 0
D-day is just one day away and finally, this is a continuation of the post(I posted a month earlier)
medium.com

100 Days of DevOps — Day 78- Python OS/Subprocess Module
Welcome to Day 78 of 100 Days of DevOps, Focus for today is Python OS/Subprocess Module
medium.com

8

Get an email whenever Prashant Lakhera publishes.
Your email Subscribe
By signing up, you will create a Medium account if you don't already have one. Review our Privacy Policy for more information about our privacy practices.

More from Prashant Lakhera Follow
AWS Community Builder, Ex-Redhat, Author, Blogger, YouTuber, RHCA, RHCDS, RHCE, Docker Certified,KAIANS, CCNA, MCP, Certified Jenkins, Terraform Certified, NSGP

Apr 28, 2019

100 Days of DevOps — Day 78- Python OS/Subprocess Module

To view the updated DevOps course(101DaysofDevOps) Course Registration link: https://www.101daysofdevops.com/register/ Course Link: https://www.101daysofdevops.com/courses/101-days-of-devops/ YouTube link:...

Dev Ops 6 min read

Share your ideas with millions of readers. Write on Medium

Apr 28, 2019

100 Days of DevOps — Day 77-Process Management in Linux

To view the updated DevOps course(101DaysofDevOps) Course Registration link: https://www.101daysofdevops.com/register/ Course Link: https://www.101daysofdevops.com/courses/101-days-of-devops/ YouTube link:...

Linux 6 min read

Apr 27, 2019

100 Days of DevOps — Day 76-How Linux Kernel is organized

To view the updated DevOps course(101DaysofDevOps) Course Registration link: https://www.101daysofdevops.com/register/ Course Link: https://www.101daysofdevops.com/courses/101-days-of-devops/ YouTube link:...

Programming 6 min read

Apr 26, 2019

100 Days of DevOps — Day 75- Introduction to Fabric

To view the updated DevOps course(101DaysofDevOps) Course Registration link: https://www.101daysofdevops.com/register/ Course Link: https://www.101daysofdevops.com/courses/101-days-of-devops/ YouTube link:...

Python 7 min read

Apr 25, 2019

100 Days of DevOps — Day 74- Introduction to GIT

To view the updated DevOps course(101DaysofDevOps) Course Registration link: https://www.101daysofdevops.com/register/ Course Link: https://www.101daysofdevops.com/courses/101-days-of-devops/...

```
$ mkdir/opt/project/GIT_DEMO
$ cd/opt/project/GIT_DEMO

$ git init
Initialized empty Git repository in /opt/project/GIT_DEMO/.git/
```

Git 5 min read

Love podcasts or audiobooks? Learn on the go with our new app. Try Knowable