Laporan Cyber Security

Analisis Log Menggunakan Python

Muhammad Mardiansyah

PUSAT PELATIHAN KERJA DAERAH JAKARTA UTARA

A. Log Blackop.id-ssl_log-May-2025

1. Memuat file log ke dalam python

```
with open("blackopidssl.txt", "r") as file:
  file_text = file.read()
print(file_text)
```

```
138.246.253.24 - - [01/May/2025:06:33:48 +0700] "GET /robots.txt HTTP/1.1" 404 1251 "-" "Mozilla/5.0 (Windows NT 10.0; Win64;
114.122.210.167 - - [01/May/2025:07:45:04 +0700] "GET /login HTTP/2" 404 1251 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
114.122.210.167 - - [01/May/2025:07:45:05 +0700] "GET /favicon.ico HTTP/2" 404 1251 "https://blackop.id/login" "Mozilla/5.0 (
52.167.144.57 - - [01/May/2025:10:18:24 +0700] "GET /robots.txt HTTP/2" 404 1251 "-" "Mozilla/5.0 AppleWebKit/537.36 (KHTML,
52.167.144.235 - - [01/May/2025:10:18:35 +0700] "GET / HTTP/2" 200 955 "-" "Mozilla/5.0 AppleWebKit/537.36 (KHTML, like Gecko
```

Gambar diatas merupakan proses untuk memuat file log dengan ekstensi .txt ke dalam sebuah variabel baru bernama file_text. karena pada log diatas belum mempunyai fitur, maka harus menambahkan fitur menggunakan regex.

2. Membuat fitur menggunakan regex.

```
[18] log_pattern = re.compile(
        r'^(?P<ip>\d{1,3}(?:\.\d{1,3}){3}) - - '
        r'\[(?P<timestamp>[^\]]+)\] '
        r'"(?P<method>GET|POST|HEAD|PUT|DELETE|OPTIONS|PATCH) '
        r'(?P<path>[^\s]+) (?P<protocol>[^"]+)" '
        r'(?P<status>\d{3}) (?P<size>\d+) '
        r'"[^"]*" "(?P<user_agent>[^"]+)"'
    )
```

```
[19] parsed_data = []
     for line in lines:
       match = log_pattern.search(line)
       if match:
         parsed_data.append(match.groupdict())
```

Pembuatan fitur menggunakan regex, pemberian fitur dilakukan untuk memberi header pada setiap data untuk mengetahui judul pada data. pada regex yang digunakan terdapat alamat ip, timestamp, metode, path, status, dan user_agent.

3. Tampilan Log setelah pemberian fitur.

```
df_log = pd.DataFrame(parsed_data)
print(df_log)
```

```
                ip                    timestamp method          path  \
0     138.246.253.24  01/May/2025:06:33:48 +0700    GET   /robots.txt
1    114.122.210.167  01/May/2025:07:45:04 +0700    GET        /login
2    114.122.210.167  01/May/2025:07:45:05 +0700    GET  /favicon.ico
3      52.167.144.57  01/May/2025:10:18:24 +0700    GET   /robots.txt
4     52.167.144.235  01/May/2025:10:18:35 +0700    GET             /
..               ...                         ...    ...           ...
846   207.154.236.97  31/May/2025:09:16:16 +0700    GET             /
847   207.154.236.97  31/May/2025:09:16:21 +0700    GET  /favicon.ico
848   207.154.236.97  31/May/2025:09:16:22 +0700    GET      /ads.txt
849  104.210.140.140  31/May/2025:11:43:54 +0700    GET   /robots.txt
850  185.247.137.138  31/May/2025:12:18:59 +0700    GET             /

     protocol status   size                                     user_agent
0    HTTP/1.1    404   1251  Mozilla/5.0 (Windows NT 10.0; Win64; x64) Appl...
1      HTTP/2    404   1251  Mozilla/5.0 (Windows NT 10.0; Win64; x64) Appl...
2      HTTP/2    404   1251  Mozilla/5.0 (Windows NT 10.0; Win64; x64) Appl...
3      HTTP/2    404   1251  Mozilla/5.0 AppleWebKit/537.36 (KHTML, like Ge...
4      HTTP/2    200    955  Mozilla/5.0 AppleWebKit/537.36 (KHTML, like Ge...
```

Gambar di atas merupakan tampilan log setelah penambahan fitur, terdapat tulisan seperti ip, timestamp, dan method pada setiap header kolom.

4. Perintah untuk mengidentifikasi celah menggunakan kata kunci

```
for baris in lines:
  cocok = log_pattern.search(baris)
  if cocok:
    data = cocok.groupdict()
    url = data["path"].lower()
    if "vuln" in url or "sql" in url or "deface" in url:
      print(f"[{data['timestamp']}] {data['ip']} mencoba akses ke: {data['path']} (status{[data['status']]}) ")
```

Perintah diatas digunakan untuk mencari kata kunci sebagai indikasi celah pada sebuah web/server seperti menggunakan kata ”vuln”, “sql”, “deface” di dalam variabel url dengan menggunakan data dari log blackop dengan menggunakan fitur “path”, jika ada kata-kata seperti ”vuln”, “sql”, “deface”, maka python akan menampilkan alamat ip, waktu, path, dan status.

5. Hasil identifikasi celah menggunakan kata kunci

```
[02/May/2025:06:47:59 +0700] 157.55.39.62 mencoba akses ke: /vuln-xss.php (status['200'])
[02/May/2025:09:08:16 +0700] 52.167.144.22 mencoba akses ke: /vuln-clickjacking.php (status['200'])
[02/May/2025:09:48:16 +0700] 36.66.160.3 mencoba akses ke: /vuln-xss.php (status['200'])
[02/May/2025:09:49:07 +0700] 36.66.160.3 mencoba akses ke: /vuln-xss.php (status['200'])
[03/May/2025:14:33:04 +0700] 182.2.141.26 mencoba akses ke: /vuln-sql.php (status['200'])
[04/May/2025:17:00:37 +0700] 114.10.142.215 mencoba akses ke: /vuln-sql.php (status['200'])
[05/May/2025:08:35:28 +0700] 180.254.142.236 mencoba akses ke: /vuln-sql.php (status['200'])
[05/May/2025:11:28:38 +0700] 114.10.143.104 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:50:19 +0700] 101.255.20.47 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:50:22 +0700] 101.255.20.47 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:50:35 +0700] 101.255.20.47 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:50:36 +0700] 101.255.20.47 mencoba akses ke: /vuln-upload.php (status['200'])
[06/May/2025:08:50:50 +0700] 101.255.20.47 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:50:54 +0700] 101.255.20.47 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:51:20 +0700] 101.255.20.47 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:51:23 +0700] 101.255.20.47 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:51:26 +0700] 101.255.20.47 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:51:48 +0700] 101.255.20.47 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:52:25 +0700] 101.255.20.47 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:52:30 +0700] 101.255.20.47 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:52:35 +0700] 101.255.20.47 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:52:57 +0700] 101.255.20.47 mencoba akses ke: /vuln-upload.php (status['200'])
[06/May/2025:08:53:00 +0700] 101.255.20.47 mencoba akses ke: /vuln-sql.php (status['200'])
[06/May/2025:08:53:04 +0700] 101.255.20.47 mencoba akses ke: /vuln-csrf.php (status['200'])
[06/May/2025:08:53:08 +0700] 101.255.20.47 mencoba akses ke: /vuln-xss.php (status['200'])
[06/May/2025:09:08:17 +0700] 101.255.20.47 mencoba akses ke: /vuln-xss.php (status['200'])
[06/May/2025:09:08:24 +0700] 101.255.20.47 mencoba akses ke: /vuln-clickjacking.php (status['200'])
[06/May/2025:09:08:29 +0700] 101.255.20.47 mencoba akses ke: /vuln-xss.php (status['200'])
[06/May/2025:09:08:36 +0700] 101.255.20.47 mencoba akses ke: /vuln-xss.php (status['200'])
[06/May/2025:09:08:57 +0700] 101.255.20.47 mencoba akses ke: /vuln-xss.php (status['200'])
[06/May/2025:09:11:27 +0700] 101.255.20.47 mencoba akses ke: /vuln-clickjacking.php (status['200'])
[06/May/2025:09:11:35 +0700] 101.255.20.47 mencoba akses ke: /vuln-deface.php (status['200'])
[06/May/2025:09:11:38 +0700] 101.255.20.47 mencoba akses ke: /vuln-deface.php (status['200'])
[06/May/2025:09:11:40 +0700] 101.255.20.47 mencoba akses ke: /vuln-deface.php (status['200'])
[06/May/2025:09:11:48 +0700] 101.255.20.47 mencoba akses ke: /vuln-deface.php (status['200'])
```

Gambar di atas merupakan hasil dari pencarian celah menggunakan kata kunci, terdapat banyak sekali percobaan untuk mengakses ke dalam alamat yang menjadi celah seperti /vuln-sql.php, /vuln-xss.php dan lain-lain dengan status ‘200’ yang artinya alamat tersebut dapat diakses.

6. Perintah untuk melihat jumlah attempt yang dilakukan.

```
from collections import Counter
import re # Ensure re is imported

# Extract IP addresses directly from the parsed_data list
# This is more efficient as the parsing has already been done.
ips = [data['ip'] for data in parsed_data]

# Use Counter to count the occurrences of each IP address
ip_counter = Counter(ips)

# Print the 10 most common IP addresses and their counts
for ip, count in ip_counter.most_common(10):
  print(f"{ip} melakukan {count} permintaan")
```

Untuk melihat siapa yang paling sering melakukan attempt untuk login, kita membutuhkan class Counter dari library collections yang di dalamnya terdapat fungsi most.common untuk melihat ip yang paling sering muncul. di dalam fungsi most.common kita bisa mengganti parameter sesuai dengan keinginan misalnya kita ingin melihat 10 ip dengan percobaan login terbanyak.

7. Hasil indentifikasi perintah untuk melihat percobaan login.

```
101.255.20.47 melakukan 255 permintaan
15.237.118.88 melakukan 34 permintaan
108.137.123.34 melakukan 34 permintaan
16.78.31.231 melakukan 34 permintaan
180.252.117.100 melakukan 16 permintaan
36.66.160.3 melakukan 13 permintaan
51.44.162.90 melakukan 12 permintaan
15.237.139.149 melakukan 12 permintaan
13.38.230.187 melakukan 12 permintaan
35.180.190.206 melakukan 12 permintaan
```

Hasil dari perintah di atas adalah akan menampilkan ip mana yang paling banyak melakukan percobaan login, pada gambar di atas ip 101.255.20.47 terlihat mencurigakan karena terlalu banyak melakukan permintaan login dengan jumlah 255 kali permintaan. tahapan selanjutnya adalah kita bisa melakukan identifikasi terhadap ip tersebut dan jika terlalu mencurigakan, kita dapat membatasi akses atau bahkan memblokir ip tersebut.

8. Perintah untuk melihat status http error atau ‘404’

```
for baris in lines:
  cocok = log_pattern.search(baris)
  if cocok:
    data = cocok.groupdict()
    if data['status'] == '404':
      print(f"[{data['timestamp']}] {data['ip']} -> {data['path']} ({[data['status']]}) ")
```

Penggunaan perintah di atas adalah untuk menemukan log dengan status error atau ‘404’, perintah ini digunakan untuk mengidentifikasi adanya percobaan dari file yang tidak ada atau mencari-cari celah pada web.

9. Hasil perintah melihat status http

```
[01/May/2025:06:33:48 +0700] 138.246.253.24 -> /robots.txt (['404'])
[01/May/2025:07:45:04 +0700] 114.122.210.167 -> /login (['404'])
[01/May/2025:07:45:05 +0700] 114.122.210.167 -> /favicon.ico (['404'])
[01/May/2025:10:18:24 +0700] 52.167.144.57 -> /robots.txt (['404'])
[01/May/2025:17:44:52 +0700] 180.245.200.93 -> /favicon.ico (['404'])
[01/May/2025:22:38:13 +0700] 52.167.144.235 -> /sitemap.xml (['404'])
[01/May/2025:23:38:59 +0700] 40.77.167.72 -> /sitemap.xml (['404'])
[02/May/2025:09:08:02 +0700] 52.167.144.57 -> /robots.txt (['404'])
[03/May/2025:05:45:02 +0700] 104.210.140.142 -> /robots.txt (['404'])
[03/May/2025:09:36:14 +0700] 138.246.253.24 -> /robots.txt (['404'])
[03/May/2025:14:29:40 +0700] 182.2.141.26 -> /favicon.ico (['404'])
[03/May/2025:14:33:05 +0700] 182.2.141.26 -> /logo.png (['404'])
[03/May/2025:22:46:09 +0700] 52.167.144.57 -> /robots.txt (['404'])
[03/May/2025:22:46:28 +0700] 52.167.144.159 -> /sitemap.txt (['404'])
[03/May/2025:23:48:01 +0700] 40.77.167.136 -> /sitemap.txt (['404'])
[04/May/2025:03:36:58 +0700] 104.210.140.135 -> /robots.txt (['404'])
[04/May/2025:11:32:58 +0700] 110.139.119.186 -> /favicon.ico (['404'])
```

Gambar di atas merupakan hasil dari perintah untuk melihat http error atau status ‘404’, perintah di atas akan menampilkan waktu si penyerang mengakses web, ip yang digunakan, serta path dan status.

10. Perintah untuk melihat pola mencurigakan

```
for data in parsed_data:
  if re.search(r'(\?|%3C|%3E|union|select|--|\*|\')', data['path'].lower()):
    print(" !!! potensi injeksi sql atau xss:", data)
```

Perintah diatas digunakan untuk melihat apakah ada percobaan aneh dari penyerang pada kolom ‘path’, regex-regex yang digunakan merupakan karakter yang biasanya digunakan untuk melakukan serangan seperti sql injection atau xss.

11. Hasil dari melihat pola mencurigakan

```
'timestamp': '07/May/2025:07:50:33 +0700', 'method': 'GET', 'path': '/?locale=ar', 'p
'timestamp': '07/May/2025:07:50:33 +0700', 'method': 'GET', 'path': '/?locale=cs', 'p
'timestamp': '07/May/2025:07:50:33 +0700', 'method': 'GET', 'path': '/?locale=da', 'p
'timestamp': '07/May/2025:07:50:33 +0700', 'method': 'GET', 'path': '/?locale=de', 'p
'timestamp': '07/May/2025:07:50:33 +0700', 'method': 'GET', 'path': '/?locale=el', 'p
'timestamp': '07/May/2025:07:50:33 +0700', 'method': 'GET', 'path': '/?locale=es', 'p
'timestamp': '07/May/2025:07:50:33 +0700', 'method': 'GET', 'path': '/?locale=es_419'
'timestamp': '07/May/2025:07:50:33 +0700', 'method': 'GET', 'path': '/?locale=es_es',
'timestamp': '07/May/2025:07:50:34 +0700', 'method': 'GET', 'path': '/?locale=fi', 'p
'timestamp': '07/May/2025:07:50:34 +0700', 'method': 'GET', 'path': '/?locale=fil', '
'timestamp': '07/May/2025:07:50:34 +0700', 'method': 'GET', 'path': '/?locale=fr', 'p
'timestamp': '07/May/2025:07:50:34 +0700', 'method': 'GET', 'path': '/?locale=he', 'p
'timestamp': '07/May/2025:07:50:34 +0700', 'method': 'GET', 'path': '/?locale=hu', 'p
```

Hasilnya adalah terdapat beberapa url yang masih menggunakan /?, dan terdapat url seperti `/?name=%3Cscript%3Ealert(%27XSS%27)%3C/script%3E'` yang dapat digunakan untuk celah untuk sql injection dan xss.

B. Blackop.id-Mar-2025_NGINX

1. Perintah untuk melihat status http error atau ‘404’

```
for baris in lines:
  cocok = log_pattern.search(baris)
  if cocok:
    data = cocok.groupdict()
    if data['status'] == '404':
      print(f"[{data['timestamp']}] {data['ip']} -> {data['path']} ({[data['status']]}) ")
```

   Pada log blackop bulan maret tidak ditemukan status http error atau status ‘404’

2. Hasil identifikasi celah menggunakan kata kunci

```
[08/Mar/2025:19:48:54 +0700] 40.77.167.136 mencoba akses ke: /vuln-sql.php (status['200'])
[09/Mar/2025:13:17:02 +0700] 52.167.144.221 mencoba akses ke: /vuln-deface.php (status['200'])
[09/Mar/2025:15:06:34 +0700] 52.167.144.208 mencoba akses ke: /vuln-sql.php (status['200'])
```

   Pada log bulan maret hanya terdapat 3 ip yang ingin melakukan akses ke dalam file dengan kerentanan seperti vuln-sql dan vuln-deface.

3. Hasil identifikasi perintah untuk melihat percobaan login.

```
203.175.9.173 melakukan 55 permintaan
13.229.99.185 melakukan 2 permintaan
13.53.135.132 melakukan 2 permintaan
44.202.113.198 melakukan 2 permintaan
52.167.144.230 melakukan 1 permintaan
40.77.167.136 melakukan 1 permintaan
103.247.9.9 melakukan 1 permintaan
23.178.112.104 melakukan 1 permintaan
23.178.112.100 melakukan 1 permintaan
23.178.112.106 melakukan 1 permintaan
```

   Hasil pada perintah untuk melihat percobaan login pada bulan maret terdapat percobaan login mencurigakan dari ip 203.175.9.173 dengan sebanyak 55 permintaan.

4. Hasil identifikasi melihat pola mencurigakan

```
4:18:22 +0700', 'method': 'GET', 'path': '/.well-known/acme-challenge/LPH--O3HQIVQ0RLZVBSB10-FRE5U74TG',
0:18:11 +0700', 'method': 'GET', 'path': '/.well-known/acme-challenge/NPKQ4MOL1K2X-26PEEG--SZ9-J_1WYMN',
3:18:22 +0700', 'method': 'GET', 'path': '/.well-known/acme-challenge/942CY2PPHB03DEKFKUP888TSIPG--4ZY',
```

   Hasil dari perintah untuk melihat pola mencurigakan terdapat url yang aneh seperi terdapat ‘- - ‘ pada url, yang diduga dapat memicu serangan sql injection untuk menghapus query di belakang perintah ‘ - - ‘.