Network Security Assignment2 Report

Method

The purpose of this project is to analyze the system logs and determined what kind of attack it’s under. After analysing the logs, I found some special features for each kind of attack, so a rule-based method was robust enough to deal with this classification problem.

SQL Injection

If the server is under SQL Injection attack, you can easily find some SQL syntax in the log, i.e.,

```
"query": "Submit=Submit&id=1%27+UNION+ALL+SELECT+NULL%2..."}
```

Sometimes you could found some more specific key words,

```
"query": "GET /vulnerabilities/sqli/"
```

```
# python code
def SQL(logs):
    count = 0
    for log in logs:
        try:
            if "SELECT" in log['url']['query']:
                count += 1
        except Exception:
            pass
    return count
```

Phishing Email

In the scenario described in the spec, server under this attack will execute `cmd.exe` to search some desired files. So it’s easy to detect `cmd.exe` in `winlogbeat` logs.

```
{
    "ProcessName": "C:\\Windows\\SysWOW64\\cmd.exe",
    "SubjectDomainName": "DESKTOP-7H8F1TK",
    "ProcessId": "0x228c",
    "SubjectUserName": "dsns",
    ...
}
```

```python
# python code
def email(logs):
    count = 0
    for log in logs:
        try:
            if 'cmd.exe' in log['winlog']['event_data']['ProcessName']:
                count += 1
        except Exception:
            pass
    return count
```

DDoS

When a server is under DDoS attack, it may response 414 error.

```
{
    "http": {
        "response": {
            "body": {"bytes": 348},
            "status_phrase": "request-uri too long",
            "headers": {
                "content-type": "text/html; charset=iso-8859-1",
                "content-length": 348
            },
            "status_code": 414
            ...
}
```

```python
# python code
def DDoS(logs):
    count = 0
    for log in logs:
        try:
            if log['http']['response']['status_phrase'] == \
                    "request-uri too long":
                count += 1
                return True
        except Exception:
            pass
    return count
```

Brute-Force Attack

You can find many login attempts with similar usernames and passwords but fail if a server is under brute-force attack.

```
"query": "Login=Login&password=flower1&username=aaliyah"
...
"query": "Login=Login&password=forall&username=aaliyah"
...
"query": "Login=Login&password=flyguy&username=aaliyah"
...
```

```python
# python code
def brute_force(logs):
    count = 0
    for log in logs:
        try:
            if "Login" in log['url']['query']:
                count += 1
        except Exception:
            pass
    return count
```

Port Scan

If a server is under port scan attack, the number of port being request for connections will be extremely large.

```python
# python code
def count_port(logs):
    ports = set()
    for log in logs:
        try:
            port = log['destination']['port']
            ports.add(port)
        except Exception:
            pass
    return len(ports)
```

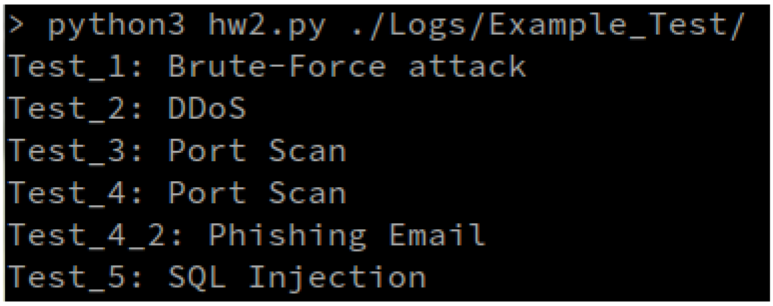
Interesting Things

- There are some logs specificly show the attack name. I don’t think this will happen in real cases.

  ```
  "query": "GET /vulnerabilities/brute/"     # brute force
  "query": "GET /vulnerabilities/sqli/"      # SQL injection
  ```

- I used to think that the logs with largest amount of bytes I/O should be DDoS, but in fact, brute-force attack is the correct answer.

```
traffic magnitude:
    brute-force: 17972076639
    DDoS:        16850761799
```



```
> python3 hw2.py ./Logs/Example_Test/
Test_1: Brute-Force attack
Test_2: DDoS
Test_3: Port Scan
Test_4: Port Scan
Test_4_2: Phishing Email
Test_5: SQL Injection
```