# Lab 5

Create a network intrusion detection system (IDS) using Java. Because you do not have root privilege on computers in EPS 254, you will not monitor live network traffic. Instead, your program will read a *.pcap* trace file using jNetPcap. Because your program is designed to process real-time network traffic, it should not process the trace file more than once. Your program will take two command-line arguments: the name of a policy file and the name of a *.pcap* trace file. When your program processes a packet from the trace, if a policy is matched, it will print an alert on the screen. IDS could be stateful or stateless. For a stateful IDS, it maintains the stream of a TCP connection between the sender and receiver. In this case, a policy will be applied to the entire stream. On the other hand, a stateless IDS only focuses on single packets. For this lab assignment, it is acceptable to wait until a TCP session is finished before checking for matches to a TCP stream.

A policy file consists of exactly one host entry, and arbitrarily many policy entries. The grammar for the configuration file is:

```
         <file> ::= <host><policy>*


         <host> ::= host=<ip>\n\n
         <policy> ::= name=<string>\n
                  <(stateful_policy >|<stateless_policy)>\n
<stateful_policy> ::= type=stateful\n
                  host_port=(any|<port>)\n
                  attacker_port=(any|<port>)\n
                  attacker=(any|<ip>)\n
                  (from_host|to_host)=<regexp>\n
  <stateless_policy> ::= type=stateless\n
                  proto=tcp|udp\n
                  host_port=(any|<port>)\n
                  attacker_port=(any|<port>)\n
                  attacker=(any|<ip>)\n
                  <sub_policy>
                  <sub_policy>*
      <sub_policy> ::= (from_host|to_host)=<regexp> (with flags=<flags>)?\n


        <string> ::= alpha-numeric string
           <ip> ::= string of form [0-255].[0-255].[0-255].[0-255]
         <port> ::= string of form [0-65535]
        <regexp> ::= Regular Expression
         <flags> ::= <flag>*
         <flag> ::= S|A|F|R|P|U
```

Submit your solution, the ***ids.java*** file, on D2L and leave you and your partner's names in the comment section. Submit a **README** file to clearly describe how to compile and execute your program. You can assume that jNetPcap library and policy files are available on the grader's computer. Before turning in your program, please test it using the following test cases.

## Test Cases

1. Blame Attack 1: A very simple policy that looks for the strings "Now I own your computer" contained in network traffic. The content in the policy file is shown as follows. Please test your IDS program using trace1.pcap (false positive), trace2.pcap, trace3.pcap provided on D2L.[1]

```
host=192.168.0.1

name=Blame Attack 1
type=stateless
proto=tcp
host_port=5551
attacker_port=any
attacker=any
to_host="Now I own your computer"
```

2. Blame Attack 2: Please use the following stateful policy to test your program using trace1.pcap, trace2.pcap, and trace3.pcap.

```
host=192.168.0.1

name=Blame Attack 2
type=stateful
host_port=5551
attacker_port=any
attacker=any
to_host="Now I own your computer"
```

3. General Buffer Overflow: A policy is provided bellow, which matches a sequence of NOPs followed by a syscall. Please test your program using trace1.pcap, trace2.pcap, and trace3.pcap.

```
host=192.168.0.1

name=Buffer Overflow
type=stateful
```

```
host_port=5551
attacker_port=any
attacker=any
to_host="\x90{10}.*\xcd\x80"
```

4. Plaintext POP: A policy to detect insecure logins to a mailserver is provided bellow. Please test your program using trace4.pcap provided on D2L.

```
host=192.168.0.1

name=Plaintext POP
type=stateless
proto=tcp
host_port=110
attacker_port=any
attacker=any
from_host="\+OK.*\r\n"
to_host="USER .*\r\n"
from_host="\+OK.*\r\n"
to_host="PASS.*\r\n"
from_host="\+OK.*\r\n"
```

5. Simulated attacker Linux boot: A policy looking at UDP packet to detect a compromised host attempting a network boot via TFTP is provided. Please test your program using trace5.pcap available on D2L.

```
host=192.168.0.1

name=TFTP attacker boot
type=stateless
proto=udp
host_port=any
attacker_port=69
attacker=any
from_host="vmlinuz"
to_host="\x00\x03\x00\x01"
```

[1] http://www.cis.upenn.edu/~stevez/cis551/2009/web/project2.html