A Software Architecture to support Misuse Intrusion Detection
Sandeep Kumar  Engene H. Spafford
Presented by Hank

Motivation
- Devise a model of matching based on Colored Petri Nets specifically targeted for misuse intrusion detection.
- Present a software architecture for structuring a pattern matching solution

Colored Petri Net – an example

```c
void simple() {
    char s[20];
    char *p;
    char *t[10];
    strcpy(s, "hello");
    p = s+5;
    strcpy(s, "world!");
    strcpy(p, s);
}
```
```c
void simple() {
    char s[20]
    char *p
    char q[10]
    strcpy(s, "hello");
    p = s+5;
    strcpy(p, "world!");
    strcpy(q, p);
}
```
Misuse Intrusion Detection Example (A so-called ‘Pattern’)

Here, in order to detect simplified TCP connections not involving retransmissions versus normal for the sequence:
1. A SYN packet from a source S to a destination D.
2. A SYN-ACK from D back to S.
3. An ACK from S to D.

Practically, this looks like:

Figure 1: Matching a TCP connection
By using the Colored Petri net, we can put events in a intrusion scenario in both space and time perspective. Present an flexible, expandable, and distributable architecture.

The architecture they proposed is very intuitive and not hard to think of. The Colored Petri Net approach can not handle undeterministic event sequence. Lacks the ability of building up assurance value.

Pros:
- By using the Colored Petri net, we can put events in an intrusion scenario in both space and time perspective.
- Present a flexible, expandable, and distributable architecture.

Cons:
- The architecture they proposed is very intuitive and not hard to think of.
- The Colored Petri Net approach cannot handle undeterministic event sequence.
- Lacks the ability of building up assurance value.