Incorporating Rule-based Engines in Event Correlation for Cyber Attacks

Min Cheol Kim

Thomas Jefferson High School for Science and Technology/Stanford University

July 30, 2013

Naval Research Lab

Code: 5540

Mentor: Dr. Anya Kim, Dr. Jim Luo

Introduction

The military, or any organization for that matter, has many computer assets that provide critical mission support. Some of these assets – such as a weapons system - may provide direct support to the mission, while others such as routers may provide indirect yet important capabilities. To ensure mission success and the safety of the soldiers these assets must be continuously protected against and monitored for attacks and other threats.

The current operation process for monitoring Navy computer networks involves a handful of cyber warriors protecting 750,000 computers by analyzing 100,000 attack-based events on a daily basis [2]. These events are triggered by potential attacks to the system. Events are generally handled on a first-come first-serve basis and are based on event severity alone. Cyber warriors are overburdened dealing with events, and at the same time do not have actionable information to enable them to make accurate decisions regarding these events. Therefore, there is a need for autonomous event prioritization that relieves the burden of the cyber warrior, and utilizes more information to accurately determine event priority.

An important consideration in calculating the event priority is how different events might be connected to each other. Often, there exist patterns that could prompt the cyber warrior to react in a more efficient manner. These patterns simply might include multiple events of the same nature, or a specific sequence of events at a specific site. This paper focuses on the tools and methods that are implemented for this event correlation.

Background

Event Prioritization

Event prioritization (EP) is a process that allows the cyber warriors to deal with threatening events in the order of importance. Order of importance is determined by examining how an event can affect the health of the system as a whole, and also its effects to critical assets on the network. It requires looking at factors beyond the event to accurately determine its priority. For example, the potential vulnerability of the target host, the mission criticality of a host, the damage that an event can inflict on a host, a target host’s connectivity to other critical hosts and other surrounding events are pieces of the puzzle needed to predict the importance of an event.

The event prioritization framework developed for this project reflects this complexity. The final EP score is dependent on numerous components, all of which are related in a complex manner (Figure 1). For example the Comprehensive Potential Host Importance value – a measure of the importance of a host due to its inherent importance as well as the importance of hosts its connected to - is calculated using Isolated Host Exposure and the host exposure of connected hosts.

ACCEPT (I forget) was developed as a tool to figure out and calculate the Event Priority values for a given set of events and the affected hosts. A more accurate calculation of the Event Priority not only must look at the individual events, but also how different events can together create more concerning situations.

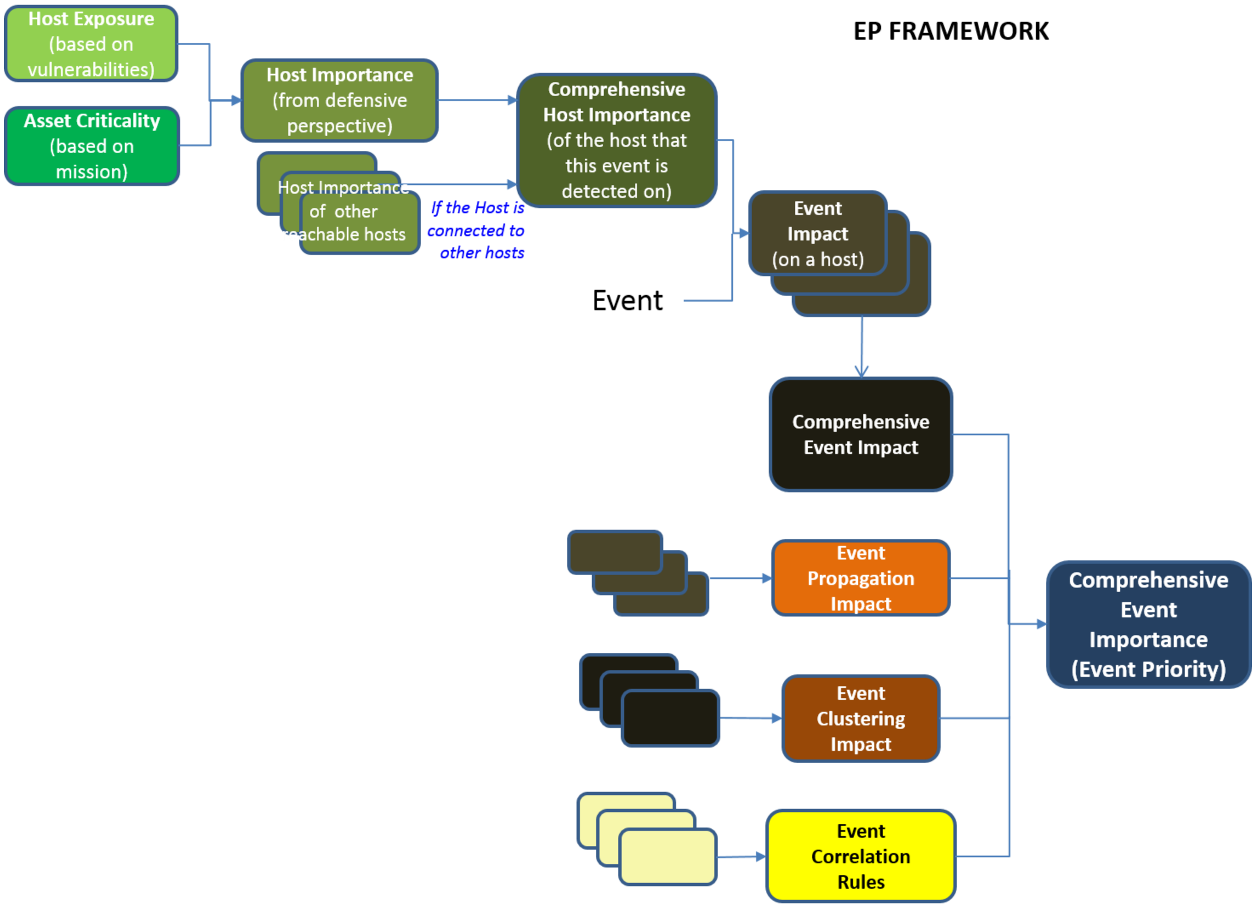

EP FRAMEWORK
Host Exposure (based on vulnerabilities)
Asset Criticality (based on mission)
Host Importance (from defensive perspective)
Host Importance of other reachable hosts
If the Host is connected to other hosts
Comprehensive Host Importance (of the host that this event is detected on)
Event
Event Impact (on a host)
Comprehensive Event Impact
Event Propagation Impact
Event Clustering Impact
Event Correlation Rules
Comprehensive Event Importance (Event Priority)

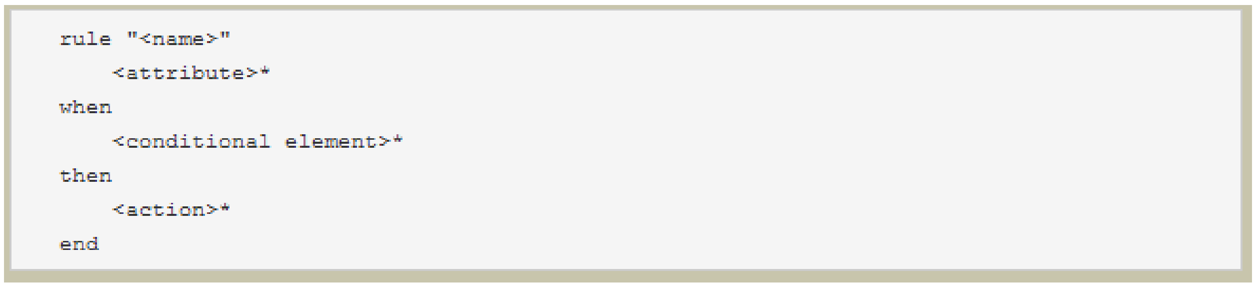
Figure 1. Event prioritization overview

Event Correlation

Event Correlation is the process at which different individual, or atomic, events can be grouped together or modified for a more efficient and meaningful processing. Sometimes the importance of small and random events might just be more than their sum. There could exist patterns in a way that makes an event more dangerous, or could reduce the amount of work that the cyber warriors must put in to understand the whole picture.

Rule-based Engine - Drools

Event correlation, in essence, is a pattern matching of different atomic events and to execute an appropriate response. Drools Expert is a "declarative, rule-based, coding environment" [3]. This tool allows you to focus on what you want the engine to do, and not worry about how each pattern is matched. Drools engine is set up in a way that different fact types can be inserted into a Knowledge Session, where the matches are made and appropriate actions are called. Everything that is inserted into the Knowledge Session goes through the rules, and these rules dictate when and how an action should be taken. A rule in Drools has the following structure:



```
rule "<name>"
    <attribute>*
when
    <conditional element>*
then
    <action>*
end
```

Figure 2. Drools Rule Structure

These types of rules can be applied to every single instance of object that is entered into the process. Drools Expert uses the Rete Algorithm, an efficient pattern matching algorithm developed by a professor at Carnegie Mellon University. The Rete Algorithm itself is quite complicated, but simply put, it works in matching the different atomic events in the given rules.

Drools Fusion uses the Drools Expert Engine and also supports a stream mode, in which events can be processed in a temporal manner. This is ideal for our event correlation scenarios, since the events will flow into the system each with a time stamp.

Module Setup

In the module that we designed, the cyber events are declared as Drools events and all other information (hosts, network topology, etc.) are designated as Drools facts. These events and facts are fed into the Knowledge Session in a timely manner – the timestamp organizes these events in a linear fashion, so that the rules can perform temporal reasoning. To develop a framework with the components involved in a cyber event, we created a Cyber Event class that will represent the actual attacks and a Host class that represent different parts of the network.

The CyberEvent class included the following fields:
The Host class included the following fields:

Not all of these fields were used in my platform for event correlation, but in the future it is certainly possible that people could devise more rules that would utilize these extra fields. Once these two classes were set up and declared, we had to test if Drools was really capable of making the pattern matches that we wanted to make. One of the concerns was that Drools Fusion would not provide a wide enough variety of functions to do so, but with functions such as accumulate() and all the temporal operators, it was evident that Drools Fusion was an acceptable method to perform event correlation. All objects, whether hosts (declared as facts) or cyber events (declared as events) must be inserted into a Knowledge Session to be considered for the rules (Figure 3).

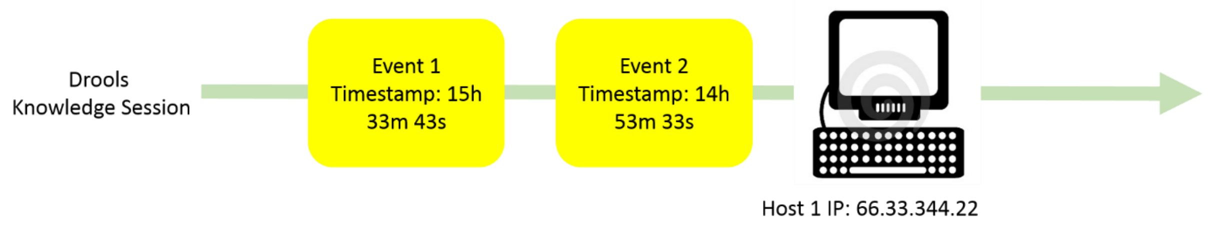


Figure 3. Example Knowledge Session

Scenarios and Simulations

To test out the Drools functions and how we can use them for event correlation purposes, several test scenarios were created and simulated with appropriate rules. The data for the simulations were from the data used to simulate ACCPT, acquired with various network monitoring tools. Host information primarily came from ePO rollup data, Asset Publishing Service Module data, and Asset Configuration Compliance Module data. The cyber event information was gathered from Snort events, ePO events table, and McAfee Network Security Manager.

Scenario 1

Scenario one involved looking for the same types of attacks on a given host in a finite window of time. A graphical representation of the scenario would be the following:

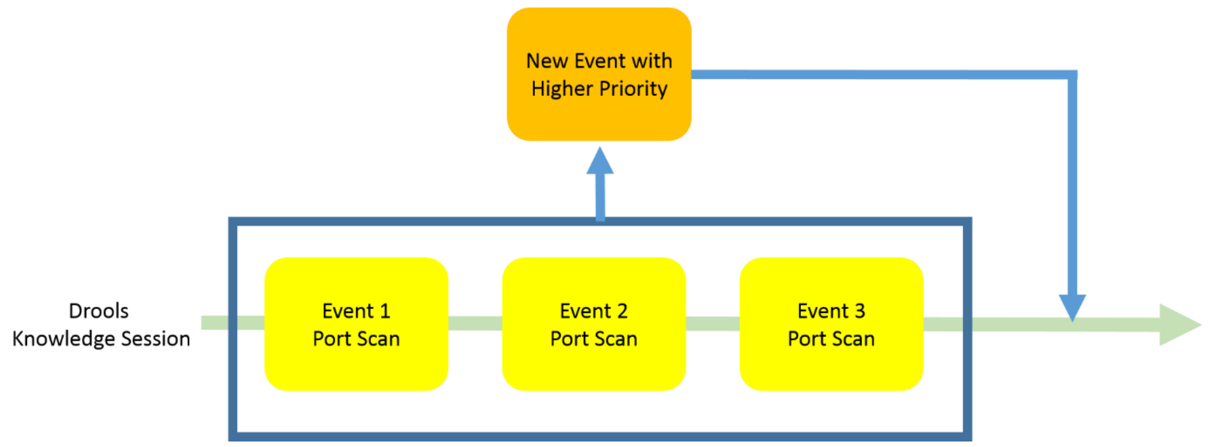


Figure 4. Sequence of Similar Events

The rule specified that if the event was anything related to a port scan, it would combine the individual events to create an aggregate event with a higher overall Event Priority. In this case, the rule was supposed to create matches for a five minute duration, counting only the events with that specific description. The output was an aggregate event with a higher Event Priority that also contained the information on all the atomic events it represented.

Scenario 2

This scenario involved a slightly more complex situation, one where an event, that is individually not very significant, could become more of a threat due to the presence of another event. A good example would be the following (Figure 5): an attack is launched on a web server on the DMZ, and then a separate event detected on a DB server that is behind the firewall which provides information to the web server. They are on separate networks, and are recognized as different events. However, a cyber warrior with the appropriate domain knowledge will know that this is an exfiltration attack to get information out of the DB server using the Web server. Therefore, these two attacks need to be correlated and the event importance of the newly correlated event raised appropriately.

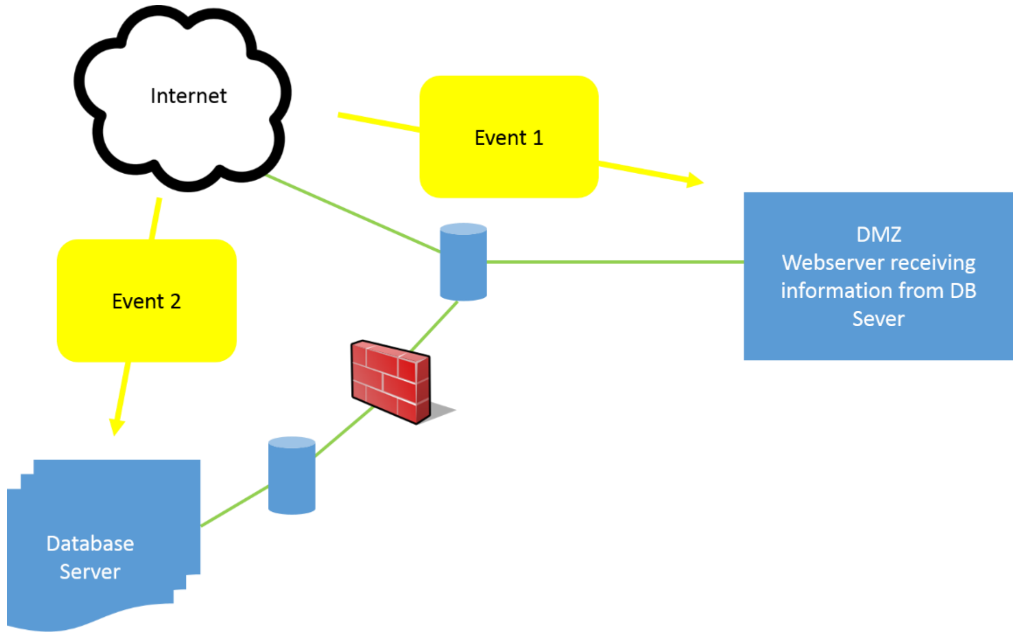


Figure 5. Graph of Scenario Two Network

Within the Drools Knowledge Session, rules can be configured to report when two events that could potentially have an impact on each other appear within a given time frame. Because of the presence of another specific event is taking place, the importance of the originally minor event could be reconsidered (Figure 6).

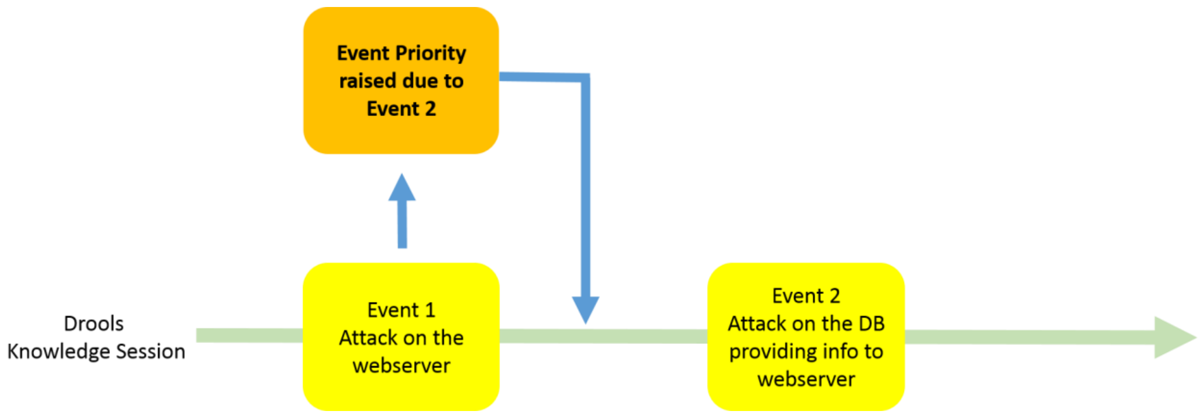

Event Priority raised due to Event 2
Drools Knowledge Session
Event 1 Attack on the webserver
Event 2 Attack on the DB providing info to webserver

Figure 6. Knowledge Session for Scenario Two

Scenario 3

This scenario involves not matching events with other relatable events, but events with their target hosts. A certain host might have a specific set of vulnerabilities, and the monitoring system does not know if the target host has a specific vulnerability. The rule-based engine, however, with all the host information loaded in the Knowledge Session, could easily search for events that targets a host's vulnerability. The engine could find this out, modify the event's Event Priority so that the cyber warriors could deal with the problem appropriately. In this example, the event has an associated vulnerability, and the target host is also known to have that vulnerability (Figure 7). In this case, the event's priority should be modified appropriately.

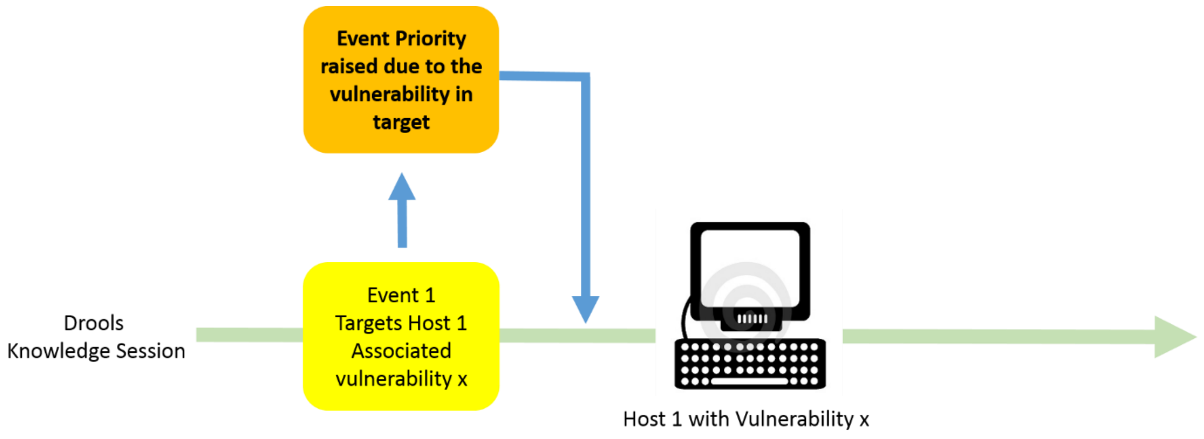

Event Priority raised due to the vulnerability in target
Drools Knowledge Session
Event 1 Targets Host 1 Associated vulnerability x
Host 1 with Vulnerability x

Figure 6. Knowledge Session for Scenario Three

Event Correlation Platform - Guvnor

In order to make the module more flexible and useful for a wide range of people, a GUI was fitted for the project with Drools Guvnor, a web application created for administration purposes over a large set of rules. An example of the rule editor GUI is shown below (Figure 7):

The GUI can be used by anyone, not necessarily programmers, to make simply changes in the module to suit their needs better. A further implementation of Guvnor would be ideal, since Guvnor also has a "test scenarios" feature that could prove useful for our purposes. The module designed by the paper was created to serve as a platform from which the administrators can modify and add their rules regarding their own network.

Conclusion

Drools Expert/Fusion is a rule-based engine that also has temporal reasoning capabilities. A module for the existing ACCEPT was created, and its efficiency was tested using three different but essential situations. Drools was found to be flexible and powerful enough to serve as the implementation of event correlation, and the module itself would eventually be used as a pre-processing tool for ACCEPT. Drools Guvnor was explored in a hopes of finding a more user-friendly administrative version of the engine, but the actual GUI could not be implemented into the module due to the time it would take to learn how to use Guvnor.

Acknowledgements

This project could not have been done without the help of Dr. Anya Kim, who always gives me these nebulous projects that require too much creativity.

References

1. Center for High Assurance Computer Systems Information Technology Division. (2011, September). *Determining asset criticality for cyber defense* (Issue Brief) (A. Kim & M. H. Kang, Authors).
2. Choksi, R., & Yu, C. (2007, April). *Study on VLAN in wireless networks* (Technical Report).
3. JBoss Drools Team. *Drools Expert User Guide*.