Determining Host Connectivity in for Event Prioritization

Min Cheol Kim
Thomas Jefferson High School for Science and Technology
July 25, 2012
Naval Research Lab
Code: 5540
Mentor: Dr. Anya Kim

Abstract

The Navy computer network is monitored and protected by a handful of cyber warriors that deal with an overwhelming amount of potential attacks (in the form of events) to the network in a mostly sequential basis. To more effectively monitor and counter these events, some form of event prioritization method is crucial. Connectivity is a notion of how influential one host is to another – particularly when the former is under attack - and is an essential part of the proposed calculation mechanism for event prioritization. The concept and implementation of connectivity within event prioritization allows priority determination to go beyond looking at the target host, enabling estimations of the effects of an attack to other connected mission-critical hosts and the combined effects of multiple events on connected hosts. To determine connectivity, the concept of network distance – how far two hosts are from each other in terms of ease of access in a network – was developed. A simple, but rational way of calculating connectivity by using Dikjstra's algorithm [1] and network distance using hops between connected was proposed.

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.

This paper describes the overall event prioritization research, and in particular focuses on the notion of connectivity, how it applies to event prioritization, and provides implementation details for determining connectivity.

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. This requires connectivity information to be available. When examining the EP framework, it is obvious that connectivity information is required in many stages of the process.

Connectivity

Connectivity is the component of the EP framework that connects many of these isolated values. It provides information on how easily one host or event can spread through the network and can affect other hosts. For example, two identical events may target two different hosts with similar characteristics. It may seem that the consequences arising from events are similar. However, when connectivity information is available, we may be able to determine that one host is strongly connected to many other hosts that support critical missions while the other does not. As another example, assume a virus-like event that targets specific platforms and applications is detected on a host. The connectivity information lets us see how fast this event can spread, given the number

and strength of connections to other hosts. The connectivity information also assists in determining which hosts with similar platforms and applications are connected to the target host. This enables the cyber warrior to determine how fast and to what degree this event can spread. In essence, it is an extremely component in determining true event priority.

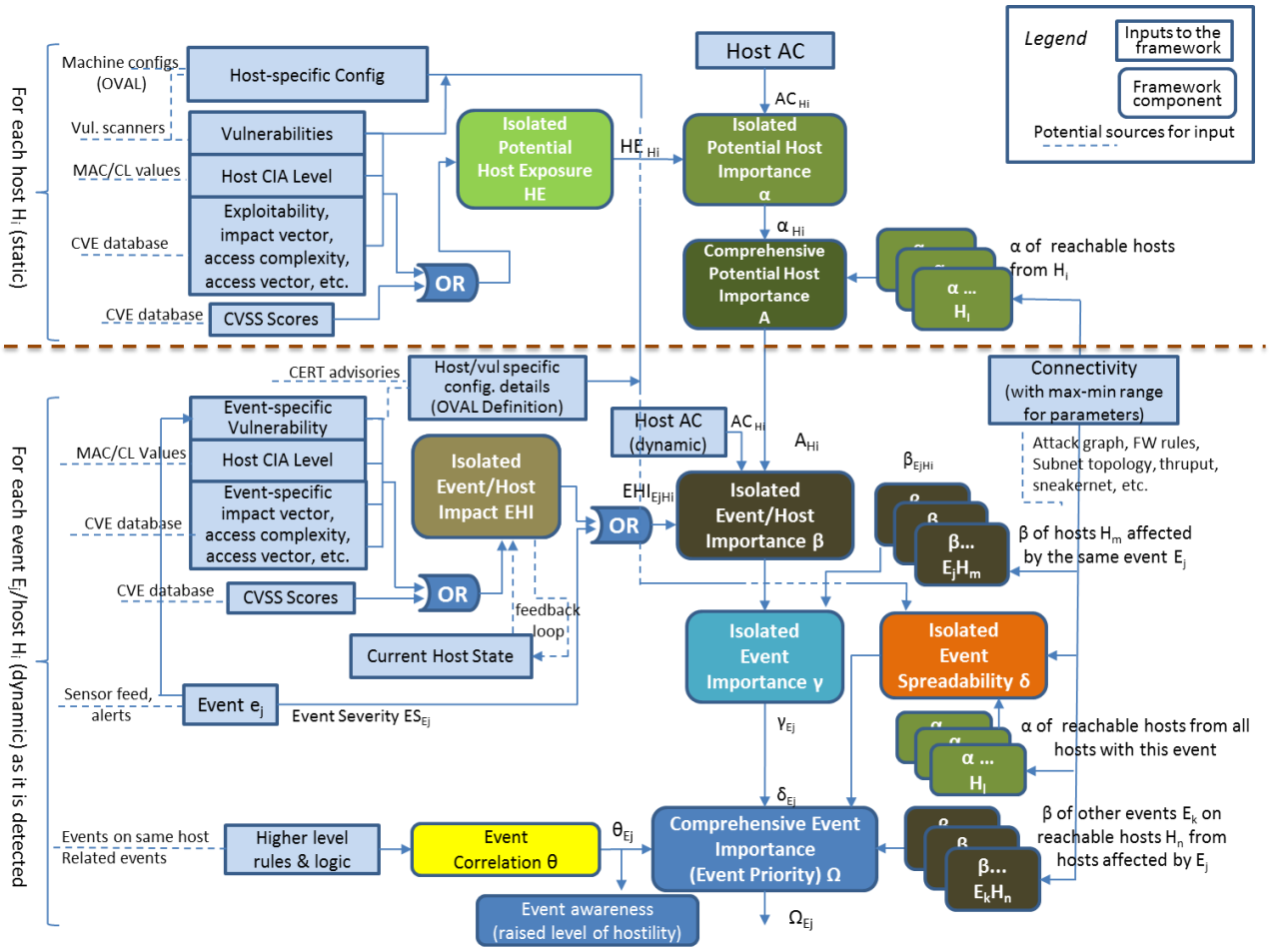

Figure 1. Event prioritization overview

Calculating Connectivity

Connectivity, as explained above, is an essential part of many components used in the calculation of event priority values. It is based on a concept we call network distance. Network distance is an abstract concept of how close two hosts are to each other; the connectivity value reflects the potential influence that a host can make to other hosts in case of an attack or when a host is compromised. Network distance and connectivity, by definition, are inversely related. The

maximum value of connectivity is defined as one, and the minimum as zero. One means fully connected (hundred percent influence) and zero means there is no connection whatsoever. Following these definitions, then, network distance of infinity would indicate a connectivity score of 0, and network distance of 0 would indicate connectivity score of 1. Hosts within a subnetwork are defined to have network distance of 0. After the calculation, the connectivity scores can be displayed in an n x n matrix where n is the number of hosts involved in the cyber domain under consideration. The matrix representation is convenient when considering firewalls, when connectivity from one host to another differs from that from the latter to the former.

Determining connectivity can be broken down into several crucial steps:

1. Mapping out a network topological map.
2. Adjusting each edge length to reflect network distance between two machines.
3. Summing up all the edges that exist between any given two hosts to find the total network distance between the hosts.
4. Calculating the connectivity score based on the total network distance, and displaying it in a connectivity matrix

Step 1: Mapping out a network topological map

Before any relevant information is used, the information on the how the machines in a given network are connected is critical. Before we can begin to calculate the connectivity, we first need to know what machines are connected to each other; this can be represented as a graph with nodes and edges, where notes can be various components of the network, such as hosts, routers, switches, firewalls, etc. Edges represent the physical connections such as cables or wireless connections. A sample representation of network topology is shown in Figure 2.

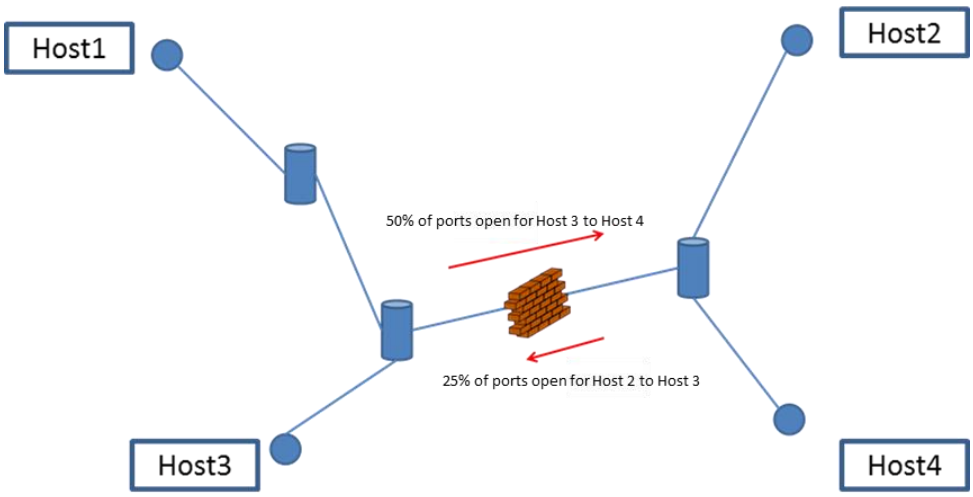

Figure 2. Sample network topology

Step 2: Adjusting edge length

Each edge between the nodes (or devices that are part of the network) represents the degree of connection between the two nodes. This edge length can take into account numerous types of information: firewall rules, throughput, subnet topology, etc. In this project, for simplicity's sake, the default distance (i.e, edge length) was considered one "hop" from one device to another, and in presence of a firewall, the edge length was increased to a certain degree starting from 1. Firewall rules can be quite complex, so in this paper, we only took into account the rules involving the percentage of the network ports open according to the firewall. Network ports are numbers that are associated with each IP address that a packet traveling through a network can be received at or sent from. In presence of a firewall, when no ports are open (i.e, no traffic allowed between different firewall interfaces) that would indicate that there is an infinite network distance between two machines on different sides of the firewall; when all are open, it would be the same as not having a firewall at all. Since the default distance between two machines is 1, the percent of ports open versus the network distance plot must include the points $(0,\infty)$ and $(100,1)$. The predicted graph would show an inverse exponential relationship, since as the percentage of ports open increases, having a small quantity of more open ports matters less and less. Figure 3 shows a predicted graph of percentage of ports open versus the network distance.

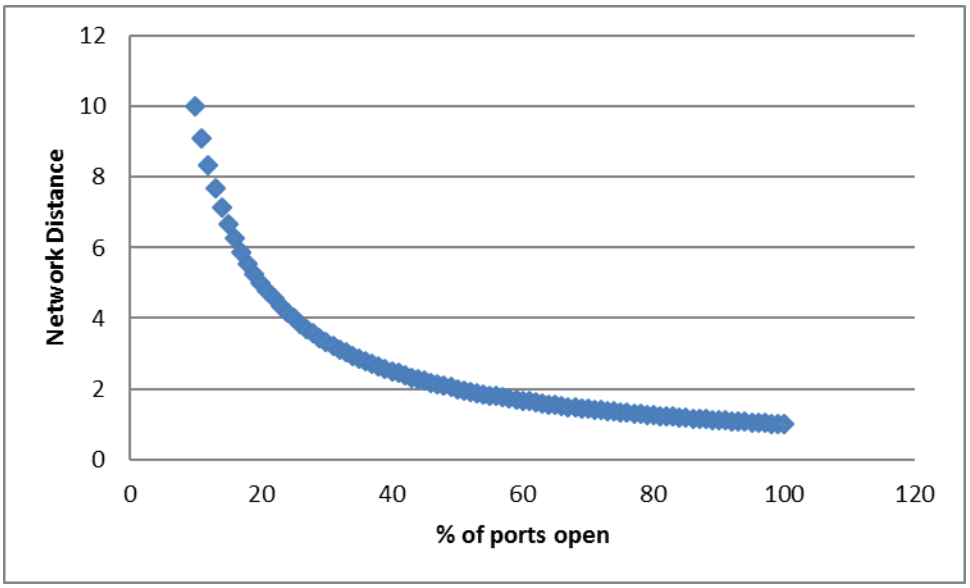

Figure 3: Network distance as a function of % of ports open in a given firewall

Based on our prediction of the network distance as a function of percentage of ports open and input from domain experts, we define a simple network distance algorithm as:

$$\text{Netwok Distance} = \begin{cases} if\ no\ firewall\ present, 1 \\ if\ firewall\ present, \dfrac{\%\ of\ ports\ open}{100} \end{cases}$$

The data involving the inbound/outbound ports, such as percentage of open ports, can be gathered by looking at the firewall rule settings. The locations of the firewall in a network and other general network configuration information can be gathered through network scanners such as Nmap [3].

It is also important to note that the network distance between two machines, in the presence of a firewall, can be different depending on which direction the traffic is heading towards. We take this into consideration by, in presence of a firewall, having multiple edges (depends on the direction and the host-specific ruleset of the firewall) between two devices. This way, distance from one node to another can be treated as direction and host specific.

Other information regarding the firewall or other aspects of the network can be used to more accurately calculate the network distance. For example, proxy rules that allow connections from small groups of hosts to another small group of hosts across the firewall would provide more detailed connectivity information. Incorporating router access control lists (ACLs) and VLAN tagging techniques [4] would further enhance the connectivity data.

Step 3: Summing up all the edges

Once the nodes and the edges that reflect that network distance are set up, we can find the total network distance between two hosts by finding the shortest "path" from one host to another. In this study, we used Dijkstra's algorithm for finding the shortest path. The algorithm looks at the distance between the current node and its entire neighboring node to find the next shortest path, while marking every node that it visits as "visited" and takes it off the "unvisited" set. This way, the shortest distance between two hosts can be represented as the total network distance. Figure 4 shows a sample code that executes Dijkstra's algorithm.

```
while(!queue.isEmpty()
        {
          wVertex v2 = queue.remove();
          Map<wVertex, Integer> edges = v2.getEdges();
          for (wVertex key : edges.keySet())
          {
            if (v2.getMinDistance() + edges.get(key) < key.getMinDistance())
            {
              key.setMinDistance(v2.getMinDistance() + edges.get(key));
              queue.add(key); //add the node as current node
            }
          }
        }
```

Figure 4. Application of Dijkstra's algorithm in finding the network distance between two hosts

Step 4: Calculating Connectivity Score

Once the distances between the hosts are calculated, there must be some way to convert that distance into a usable form of number, or a score. Connectivity, by definition, reflects complete connection when 1, and completely disconnection at 0. An exponential graph is predicted since as the network distance increases, a certain amount of additional increase would have a smaller effect on the connectivity. A predicted network distance vs. connectivity graph is shown in Figure 5.

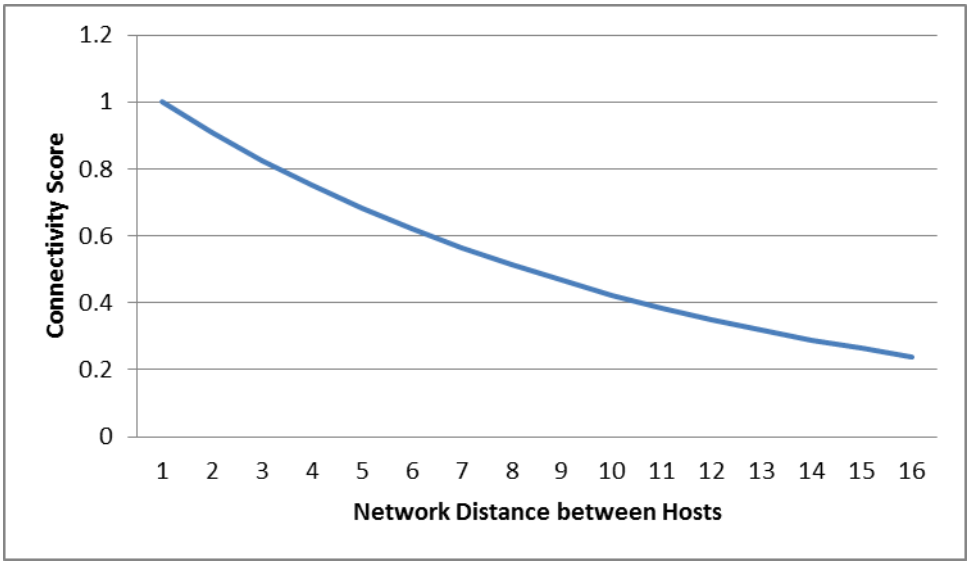

Figure 5. Network distance vs. connectivity score

The equation would be in the form of:

$Connectivity\ Score = k^{-\sum Network\ Distance\ Elements}$, where k is a constant to be determined by empirical data. Upon consulting with domain experts, a value between 1.1 and 1.2 seemed to be the most appropriate. Once the connectivity between each and every host has been calculated, it can be displayed in a connectivity matrix. A connectivity matrix for the sample network shown in Figure 1 is shown in Figure 6.

	Host1	Host2	Host3	Host4
From Host1	1.000	0.621	0.751	0.621
From Host2	0.513	1.000	0.564	1.000
From Host3	0.751	0.683	1.000	0.683
From Host4	0.513	1.000	0.564	1.000

Figure 6. Connectivity matrix for sample network

Conclusion

Event prioritization is a novel way to allow cyber warriors to approach multiple simultaneous events in a system. Connectivity is a crucial step to many of the calculations involved in producing the final EP score. This paper introduces a method to calculate the connectivity via calculation of the network distance between two hosts, enabled by Dijkstra's algorithm. The usage of firewall rules (percentage of ports open) exerts its effect on the connectivity by elongating the default network distance (1) appropriately. The total network distance between two hosts in different subnets can be used to calculate the connectivity score that ranges from 0 to 1.

Acknowledgements

This project could not have been done without the help of Joe Langley. I would also like to thank Anya Kim and Myong Kang for guiding me throughout the project.

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. Fyodor (Presenter). (2006, January). *Advanced network reconnaissance with Nmap* [Lecture].
4. Puthuparampil, M. (n.d.). *Dijkstra's algorithm* (Report).