Cyberspace and Networked Systems – Paradigms for Security and Dynamic Attacks

D. W. Repperger¹, M. W. Haas¹, J. T. McDonald², R. L. Ewing¹

¹Air Force Research Laboratory
Wright-Patterson Air Force Base,
Dayton, Ohio 45433-7022
d.repperger@ieee.org

²Air Force Institute of Technology
School of Electrical and Computer Engineering
WPAFB, Ohio 45433

Abstract—A variety of issues are discussed related to cyberspace warfare and network systems. Present systems are now highly distributed, yet may be extremely vulnerable. Means of developing performance, vulnerability and dynamic response to several types of cyber attacks are considered. It is seen that both the architecture of a distributed network and the characteristics of the constituent nodes that control flows highly influence performance and vulnerability.

I. INTRODUCTION

Modern military, industry and academic systems have now evolved into complex networks which provide numerous advantages. For example, if efficient use of multiple resources on a network can be coordinated and employed at all times of the day and night, a truly powerful arrangement can result. Associated with the coordination of these multiple assets, problems of vulnerability then arise. If a key node or groups of nodes with important functionality are attacked, the entire system may go down resulting in a significant loss of capability. Thus working in cyberspace has significant advantages, through the leveraging of multiple resources, but at the cost of possible vulnerability to cyber attacks.

Analogous to distributed and complex networks, electronic circuits, software, and other systems have similar advantages and problems. Another common example is the satellites that orbit the earth and are the basis of cell phone communications, data links, and other wireless forms of data transfer. If a key satellite or node is attacked or becomes even partially compromised, this can have high sensitivity to the performance of an entire communication’s system.

There are a number of issues this paper will bring to light. For example, if a given electronic circuit was specified, how would it be best to maximize the robustness of the circuit to an external exogenous network attack? Is it possible to build additional links, nodes, circuit elements, etc. around a given system to improve its ability to be resistant to attack but with minimum alterations? Similar research issues can be seen in design of software to be defiant to attack and modification. Another interesting research issue is that if an external observer can monitor the actions of a circuit, software

II. SOME ISSUES ON DISTRIBUTED NETWORKS

Since brevity must be the style here, only the most relevant issues can be addressed. The key points are summarized as follows:

1. Distributed and networked systems are inherently more powerful than their predecessors if they efficiently use their distributed assets in a coordinated manner.

2. Associated with the new capability of a distributed network, the vulnerability of such entities now becomes an important factor. It is much more difficult to protect a distributed network, since it is not at one location. Thus the power of capability gained from distribution inherently produces weaknesses if the network goes down.

3. For cyber attacks, a dynamic viewpoint is essential to understand the efficacy of a cyber attack. The time constant of recovery is one measure of how much capability is lost. For example if a network is down for only a few seconds, the effect of an attack is minimal. However, if a network is down for a long period, when critical events have to be accomplished, then the cyber attack is much more effective.

This paper will examine a number of issues related to how to conceptualize distributed and complex networks. The performance, vulnerability, and other issues are key concerns in utilizing this new technology. One important issue is the architecture or structure of the distributed system.
III. VULNERABILITY - GRAPH ARCHITECTURE

To better understand the vulnerability of networks, the basic architecture or structure is extremely important to analyze. To briefly summarize how the architecture influences the vulnerability, some basics from graph theory [1,2] are brought into this discussion. There are many types of graph architectures but two types are well known:

**Graph Architecture 1: - Random Graph:**
A random graph, Erdös-Rényi graph [3] can be constructed by having approximately the same number of links between the nodes. A simple example is the land highway system in the USA where the major cities have about the same number of links (major highways) between each major city (left plot in Fig. (1) from [3]). In Figure (1) the term random graph means that the distribution of the number of links (x axis) follows a normal curve with the mean of the density as the most common number of links between nodes. This framework is sometimes called a thin tail distribution.

**Graph Architecture 2 – Scale Free Graph:**
A scale free graph means that a power law relationship would exist for the same plot as in Figure (1) for the random graph (number of nodes with k links versus number of links). This is typical of the airline routes, as displayed in Figure (1) on the right most plot. In this case there are many nodes with a small amount of connections and a few key nodes with many connections, which is typical of airline routes with central hubs. There are many common networks that are considered to have a scale-free architecture such as the Internet as displayed in Figure (2). In this instance, as new links are added, the most highly connected nodes more likely gain additional links, which is the “rich get richer” concept. This framework is sometimes called a fat tail distribution because on a linear plot such as Figure (1) on the right, the tail is much higher to the right as compared to the random graph case.

IV. VULNERABILITY OF THE GRAPH TYPES

There are interesting issues with respect to the best ways to mitigate the performance of networks. It should be noted that when designing an attack on a network with the goal of compromising its ability to perform, the type of attack can be very dependent on the architecture of the network [1,2,3]. It can be shown that for a scale-free network, attacking the most highly connected node will produce the greatest performance decrement. Thus a focused attack on a scale-free network is the most productive. For a random graph, however, a focused attack has about the same effect since all nodes are equally connected.

For a random attack (an attack on a random node) is less effective, on average, for a scale free graph. This is because the majority of nodes have the fewest connections. For a random graph, a random attack has the same effect throughout and no node is more vulnerable. Thus the scale free networks are more robust to random attacks and decision makers should allocate their resources to protect the most highly connected nodes in the scale free case.

V. SOME ISSUES ON HARDWARE/SOFTWARE

Networks can represent electronic circuits, other systems, software and numerous other entities. An important issue to consider is what should be done to protect existing hardware, such as satellites that orbit the earth and provide invaluable communications capability? In Fig. (3) is a visualization of the network attack problem as applied to an existing...
electronic circuit (which could be considered a satellite or other communications system).

![Image of a network diagram](image)

**Fig. (3) – How to Robustify a Network?**

In Fig. (3) the network on the left is assumed to have a fixed structure. The network on the right has been augmented with additional nodes and links. The goal is to include the additional nodes and links to the original system such that the new system appears to act almost like a random graph.

As discussed previously, if a network (or electronic circuit) can appear externally to look like a random graph then the two types of attacks (focused attacks and random attacks) will both have a diluted effect. It will be most costly for the attacker to expend his resources with a low payoff in terms of compromising the integrity of a network. This work is currently ongoing at The Air Force Institute of Technology for both hardware and software systems.

**VI. NETWORK ATTACK TIME CONSTANT OF RECOVERY**

Another important consideration is the effect of a network attack on the time interval before the network can recover to its previous level of performance. If network performance could be measured in units of bits/second or events/second, then the percent reduction in performance multiplied by the time the network is down can be equated to bit or events. The dynamic response, such as a time constant (the time to return to 62.7% of its prior performance) may be a valuable measure. Obviously networks with short time constants can recover quickly and the effect of the network being down is minimal. This measure directly addresses the question: “How long before the network can return to its previous level of performance?”

**VII. THE DYNAMIC ELECTRONIC FOOTPRINT**

An additional interesting issue related to the dynamic response of networks is to identify the electronic signature of a system and to reverse engineer its electronic footprint. Thus by externally monitoring an electronic circuit or software system, is it possible to determine the functionality of the object under scrutiny? This also applies to human-machine systems. If an external observer can monitor the keystrokes, actions, outputs of a human-machine system, is it possible to identify what type of activity is being performed? Referring back to the example of Fig. (3), the goal would then be to disguise the actions of the human-machine system by building an external set of links and nodes about the original system to help camouflage its functionality. There are a number of issues for research of this type which generalizes from software and hardware to human machine systems.

Again, as mentioned with the technique to robustify a network, if the human-machine system could be made to appear random from an external perspective, this design produces the maximum amount of uncertainty to the attacker. Making the attack on a network cost ineffective is one means to help reduce vulnerability.

**VIII. OPTIMIZATION/VULNERABILITY OF A USAF SYSTEM**

Finally, a example of a USAF network is presented to show at least one means to better measure the vulnerability/performance of such entities. To briefly describe the problem [1, 2], an air logistics system is modeled in Fig. (4) as a distributed network. This model represents the problem of refreshing and resupplying an aircraft after it has landed and then having the aircraft take off. Fig. (4) portrays the architecture chosen and the goal is to determine the 15 flow vectors that optimize performance in the system. The performance is measured by mutual information flow through the network which has units of bits/second.

To determine the values of the 15 unknown flow variables, from the architecture in Fig. (4), a technique from Graph Theory is used to develop constraint equations. Constraint equations arise through a cut set drawn around any node or groups of nodes. Fig. (5) illustrates the cut set concept which basically states that flows cannot be created or destroyed. The sum of the flows into or out of a cut set must sum to zero if no source or sink is included in the cut set. The five equations for the five nodes in Fig. (4) are specified as follows:

- **ATOF:** \[ f_x + f_2 + f_4 + f_6 + f_8 = f_1 + f_3 + f_5 + f_7 + f_x \] (1)
- **PS:** \[ f_1 + f_2 = f_4 + f_6 + f_8 \] (2)
- **RS:** \[ f_1 + f_2 + f_3 + f_5 + f_9 = f_10 + f_12 \] (3)
- **FS:** \[ f_10 + f_12 + f_3 = f_14 + f_13 + f_4 + f_11 \] (4)

![Image of a logistics network diagram](image)
CS: \[ f_5 + f_{14} = f_{13} + f_6 \]

(5) fitness function maximizing the mutual information. Fig. (7) shows the corresponding minimization of the mutual information.

Fig. (5) – The Cut Set Concept From Graph Theory

Of the five equations in (1-5), four are independent. To find the optimal vector of flows, a genetic algorithm procedure was then employed. The fitness function (objective function) was the mutual information flow through the entire network. The genetic algorithm was run twice, first to maximize the mutual information flow and then to minimize the mutual information flow. The minimum mutual information flow was also very important to determine because it provides a paradigm to attack a network. For example, the best set of node flows to congest the network also provides an exemplar on how to attack a network. One can view this procedure as the control at each of the nodes being modulated for maximum or minimum mutual information flow.

Fig. (6) portrays a statistic of the elite pool with fitness function maximizing the mutual information. Fig. (7) – Minimization of Mutual Information

To determine a measure of vulnerability, a definition of sensitivity is presented as follows [1,2]:

\[
S_{W}^{T} := \lim_{\Delta W \to 0} \frac{\Delta T}{\Delta W} = \frac{\partial T}{\partial W} W
\]

(6)

Where \( T \) is the cut set flow of interest and \( W \) is the overall network’s mutual information flow change (\( \Delta W \neq 0 \)). This definition is derived from the concept that certain sets of node(s) will be affected by the overall mutual information flow. Finally the system in Fig. (4) was simulated 5 times with both maximization and minimization of the mutual information. \( S_{W}^{T} \) of equation (6) was evaluated and averaged for a cut set around a highly connected node in Fig. (4) (ATOF) and compared to a lesser connected node (PS). Fig. (8) shows the comparison using the vulnerability measure specified in equation (6). It is clear that the more highly connected nodes tend to have substantially higher vulnerability as specified via equation (6).

Fig. (8) – Sensitivity Measure for Two Cut Sets

171
IX. CONCLUSIONS AND DISCUSSION

Analysis of a USAF logistics system is conducted using principals from Graph Theory, optimization, and information theory. A number of issues are raised on the factors behind vulnerability including architecture and flow control variables. A brute force method to determine vulnerability can be obtained using genetic algorithms.

REFERENCES

