On Cyber Warfare Command and Control Systems

Norman R. Howes

Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311
(*howes@ida.org*)

Michael Mezzino

University of Houston - Clear Lake
2700 Bay Area Blvd.
Houston, TX 77058
(*mezzino@cl.uh.edu*)

John Sarkesain

Missile Defense Agency
7100 Defense Pentagon
Washington, DC 20301-7100
(*john.sarkesain@mda.osd.mil*)

Abstract

As Defense agencies and services expand their reliance on computer networks, risk to information availability and integrity increases. It is no longer adequate to rely solely on the now traditional defense-in-depth strategy. We must recognize that we are engaged in a form of warfare, cyber warfare, and deploy our resources using the strategy and tactics of warfare. Most Defense organizations have not yet developed strategies or tactics for cyber warfare. This causes security devices to be used ineffectively and responses to be untimely. Cyber warfare then becomes a one-sided battle where the attacker makes all the strikes and the target of the attack responds so slowly that the attacker usually gets away without being identified.

Employing cyber warfare strategy and tactics requires a cyber warfare command and control system. Responses to cyber attacks do not require offensive measures outside our own network boundaries to be effective, but they do require timely responses. Timely offensive action taken within our own network boundaries can lead to an identification of the attacker.

During the past two years we have developed a prototype cyber warfare command and control system to demonstrate that defense-in-depth can be taken to a new level that is active and anticipatory rather than passive and reactive.

1. Introduction

Names like *cyber command and control system* or *network defense management system* are sometimes used to describe systems that are used for the remote management of firewalls, intrusion detection systems, and other network components and subsystems. The term *cyber warfare command and control system,* as used in this paper, means something quite different. Certainly, the remote management of firewalls and intrusion detection systems, etc. should be part of a cyber warfare command and control system, but what we have in mind is something far more extensive. To help the reader understand exactly what we mean by a cyber warfare command and control system, we begin by defining the term.

As with any definition of an unfamiliar term, we precede our definition with some motivation. Intuitively, what we mean by cyber warfare command and control is the analogue of the term *command and control* (C2) as applied to conventional (kinetic) warfare. In order to motivate our definition, we need to explain the analogy and also explain why the analogy is important. Thereafter, we will be able to define what we mean by a cyber warfare command and control system by telling the reader what our analogy is for each component of a kinetic warfare command and control system. It is assumed that the reader already understands what a kinetic warfare command and control system is, whether at the tactical, operational, or strategic level.

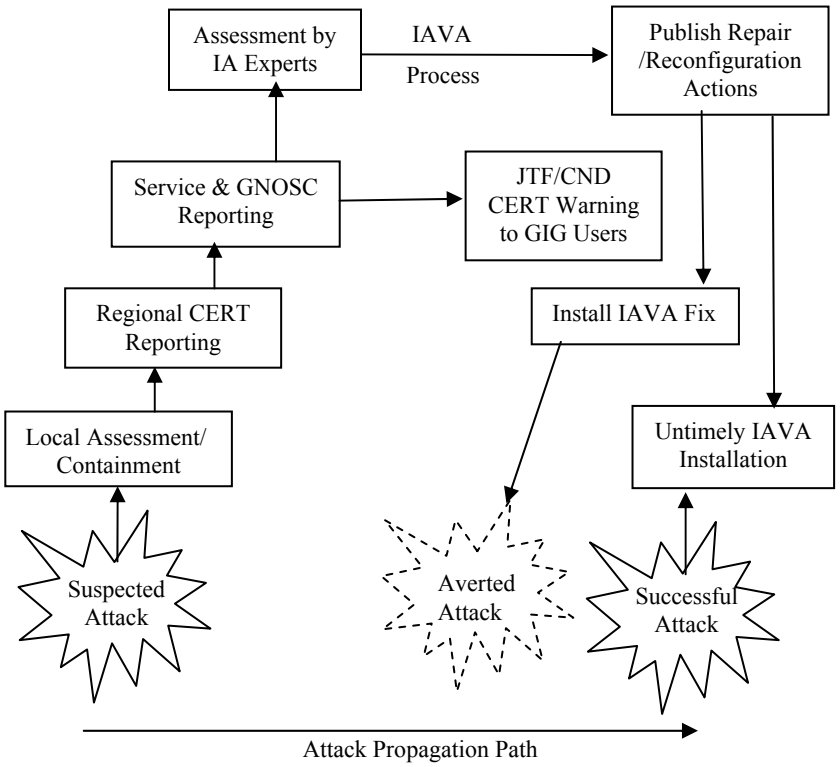
Finally, in order that the reader may not think this is merely an intellectual pursuit, we conclude with a description of a prototype cyber warfare C2 system that the authors, and others, have been developing during the past two years.

2. The Analogy

We start our discussion of the analogy of cyber warfare C2 systems to kinetic warfare C2 systems with some observations about (1) why the kinetic warfare C2 model cannot be applied directly to cyber warfare, and (2) what concepts of kinetic warfare C2 are missing from current cyber defense philosophies that inhibit the defenders from being as effective as the attackers.

Within defense communities, cyber defense is currently organized along the same lines as kinetic warfare C2 systems as shown in Figure 1. Suspected attacks are assessed locally and an attempt is made to contain them. Often, containment options are limited due to requirements for obtaining permission before taking action, from a higher-level organization. Thereafter, a reporting sequence begins that is similar to the reporting of events up the chain of command in a kinetic warfare C2 system.

Figure 1. Kinetic C2 model of cyber defense



As the report of the attack reaches higher levels, as shown in Figure 1, other organizations and commands are alerted to the possibility of a similar attack, and Information Assurance (IA) experts analyze the attack as part of the IAVA process. As higher-level commands receive information about an attack, they pass orders down the chain of command regarding how to response to the attack. Eventually, Information Assurance Vulnerability Assessments (IAVAs) are produced that detail fixes that may eliminate the vulnerability or reduce its seriousness. If the IAVA is not installed in a timely manner, which is often the case, the risk of attack remains at locations that have not complied with the IAVA.

Figure 1 shows the attack propagating as we move to the right of the figure. Also, it shows the IAVA process (which involves time) moving to the right. The upward propagation of reports is shown staggered to the right to indicate the passage of time in the figure. The time between the original discovery of a suspected attack and the JTF/CNO CERT warnings can take hours to days. In addition to this delay, many defensive measures (like the segmentation of a network or the shutting down of certain services) often have to await an order from a higher level of command. Eventually, IAVA fixes are published that reduce the risk of this type of attack. This can take several more days.

This type of hierarchical organization that relies on situation reports going up the chain of command for decision making and orders coming back down the chain of command that implement these decisions, does not work well for cyber defense. Cyber battles usually take place in the seconds to minutes range whereas kinetic warfare battles occur in the hours to days range. Consequently, we cannot hope to use the kinetic warfare organizational model of command and control effectively for cyber warfare. On the other hand, we do not want to lose the kinetic warfare command structure when we integrate cyber warfare C2 into the overall kinetic warfare command and control.

Kinetic warfare command and control is based on the concept of *cells* at each level of command. For instance there is an *operations cell* (OPS cell), an *intelligence cell* (INT cell), and a *logistics cell* (LOG cell), etc. These are *physical cells* in the sense that they are located in different places, and you cannot be in multiple cells at the same time because you cannot be in multiple places at the same time. While there is interaction among these physical cells, there is detailed information in them that is not in the other cells. What the other cells get is summarizations of this information usually referred to as a *situational pictures*, e.g. the intelligence picture or the operational picture.

In what follows, we propose a cyber warfare organizational model based on *virtual cells* (also referred to as *logical cells*) as opposed to physical cells. Virtual cells exist in cyberspace rather than in the physical space of a command center. It is possible for a *cyber warrior* to be in multiple virtual cells simultaneously. The ability to be in multiple virtual cells at the same time is a powerful C2 abstraction. It avoids much of the need for hierarchical reporting of situational information. Cyber warfare commanders can be members of multiple lower level virtual cells, multiple *peer cells* (virtual cells at their own level of command at other locations) and, if permitted, they can be members of higher-level virtual cells.

Physical command and control cells only permit a single organizational structure that we refer to as the *chain of command*. This structure is determined by the *reports to* relationship. The reports to relationship generates *hierarchical relations*. It is an example of a *many-to-one* relationship. In contrast, virtual cells are organized by the *membership* relationship that will be explained in the next section. The membership relationship is an example of a *many-to-many* relationship. Many-to-many relationships are more general than one-to-many relationships. They generate *network relations*. Hierarchical relations are sub-relations of network relations. As a result, we can have an organizational structure for cyber warfare in which the chain-of-command relation is embedded.

This fact allows us to integrate the cyber warfare organizational structure with the kinetic warfare organizational structure in a natural way that allows us to maintain the conventional chain of command for command purposes while providing a more general cyber warfare organizational structure for conducting cyber warfare. How this is done will be explained in the next section.

As important as organizational models are for command and control, C2 systems are used for more than just providing an organizational structure for communicating in a formal way within the organization. For instance, C2 systems are used for developing *strategy*, executing *tactics*, maintaining a common operational picture, developing courses of action, and maintaining intelligence information.

Classical kinetic warfare has a tradition of studying the motives, tactics, and weapons of potential enemies in order to develop strategies and tactics in advance. Such strategies include a

mature understanding of operational art to include, organizing for warfare, the means of communicating within the organization, likely events during warfare, courses of action (COAs) to respond to them, how battlespace intelligence will be gained, how situational awareness will be presented, and so forth. Each strategy element of kinetic warfare has a parallel in cyber warfare. This paper puts forward a concept of operations for dealing with cyber warfare in Section 4 and shows in Section 5 how our prototype cyber warfare command and control system supports this concept of operations.

Today, cyber defense philosophies make little use of military strategy and tactics. Military commanders know that there are times when the best defensive strategy is to take the offensive. They also know the value of the tactics of *deception* and *maneuver*. The fact that cyber defense philosophies, such as the defense-in-depth philosophy, do not take advantage of offensive operations, or use the tactics of deception and maneuver, inhibits the defenders from being as effective as the attackers. Current cyber defense strategies tend to be *static* and their tactics tend to be *reactive*. The trend is to build layers of static defenses in the hope that every attack will be defeated by at least one of the layers. When this fails, there is a reaction that consists of determining where and how the defenses were penetrated, patching the defenses to stop future similar penetrations, and restoring the system to a coherent state.

It is important to note that most current attacks are of a specific type such as a single virus launched as an email attachment or a denial of service attack against a specific type of server with a specific vulnerability. For the most part these attacks have not been orchestrated and executed with a strategy designed to cause strategic damage to multiple systems within a network. It is likely that in times of war, nation state attackers will launch multiple coordinated attacks against multiple targets using a variety of attack types. Such attacks will attempt to neutralize multiple layers of defense-in-depth assets simultaneously, leaving the systems on a network open to a second wave of attacks that create extensive damage that takes hours or days to repair. Such attacks to mission critical combat systems could be disastrous. Our cyber warfare C2 prototype system addresses such attack scenarios by providing rapid coordination, dynamic network defense mechanisms, deception, and predefined courses of action based on both monitoring of actual attacks and simulating complex attacks.

Consequently, we should be striving for a cyber defense strategy that is *dynamic* which is supported by tactics that are *anticipatory*. Kinetic warfare strategies and tactics are already dynamic and anticipatory. Perhaps there are things we can learn from them that will be beneficial in the execution of cyber warfare. We have already mentioned that kinetic warfare makes use of the tactics of deception and maneuver. We have also mentioned that kinetic warfare consists of both offensive and defensive operations. In order to apply a strategy that includes transitioning between offensive and defensive operations and tactics that involves deception and maneuver, our cyber resources must be mobile and we must have a capability to coordinate the movement of resources to out maneuver the adversary, to feint and to deceive the adversary. In the kinetic warfare domain, we call this a *command and control* system. That is also what we will call such a capability in the cyber warfare domain.

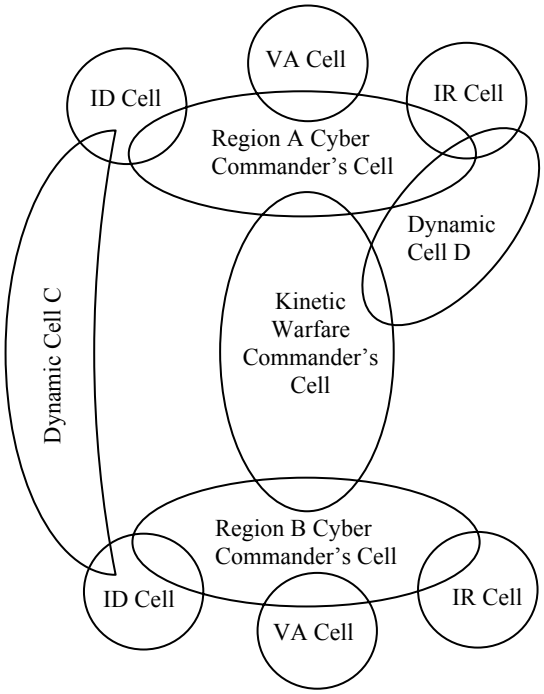
In the next section we present an organizational model for conducting cyber warfare that is supported by our prototype cyber warfare C2 system. This organizational model supports a strategy of using both offensive and defensive cyber operations.

3. The Organizational Model

We mentioned that the organizational model we are proposing for cyber warfare is based on virtual cells. Figure 2 shows some of the virtual cells of this model. Before we discuss these cells, we first explain the drawing conventions in this figure. The various shapes in the figure (circles, ovals, etc.) represent virtual cells. When two of these figures intersect, it signifies that these two cells have at least one member in common, i.e. there is some cyber warrior that is a member of both of these cells simultaneously.

In the mathematical Theory of Sets, sets are defined by the membership relationship, namely, a set *S* is defined by specifying which elements are members of *S*. In Set Theory we call diagrams like the one shown in Figure 2 *Venn diagrams*. We can think of a virtual cell as a set that consists of the members of the cell. But to do so only conveys part of the concept of a virtual cell.

Figure 2: Cyber Warfare C2 Organizational Model



Unlike sets, virtual cells do not always have the same members. Members may come and go. For instance, virtual cells may be in operation 24 hours a day. There may be multiple shifts during the day, and the entire membership of a cell may change when the shift changes. Also, unlike sets, virtual cells may exist for a while and then they no longer exist, as for instance as in the case of *dynamic virtual cells*. Notice that there are two cells in Figure 2 that are called dynamic cells. The other cells are referred to as *core cells*. Core cells are cells that are always present in the cyber warfare command and control system. In contrast to core cells, dynamic cells are created on the fly, so to speak, are used for conducting some task or operation, and thereafter decommissioned. When a virtual cell is decommissioned, it no longer exists. So at any given instant, a virtual cell is a set, defined by the set theoretic membership relationship, but at

some other instant, while it is still a set, it may be another set because the membership has changed.

We will first discuss the core cells and then we will discuss the dynamic cells. Each of the core cells has a cell commander and possibly a deputy commander. Since cyber warfare C2 systems are intended to operate 24 hours a day, there will be multiple cyber warriors that are authorized to serve as the cell commander (or deputy cell commander). The cell commander of the Kinetic Warfare Commander's (KWC) cell may be the theatre commander where the kinetic warfare C2 system operates and of which the cyber warfare C2 system is a part. But more likely, the theatre commander will delegate the command of the KWC cell to another senior officer. The KWC cell oversees the interface of the cyber warfare C2 system with the kinetic warfare C2 system.

While the KWC oversees cyber operations, the actual command and control of cyber warfare is left to the regional Cyber Warfare Commanders (CWCs). Each CWC commands a regional CWC cell. Figure 2 shows two regions (A and B as an example), but in practice there can be any number of CWCs depending on how many regions are in the kinetic warfare C2 system that the cyber C2 system protects. What constitutes a region depends on factors such as the level of the C2 system (tactical, operational, etc.), the size of the theatre, and the topology of the C2 network.

Since the KWC cell and the regional CWC cells in Figure 2 intersect, we can conclude that there is some member of the KWC cell that is a member of Region A's CWC cell and another member (possibly the same member) that is a member of Region B's CWC cell. In our prototype cyber C2 system we have always assumed that the KWC cell commander is a member of all the regional CWC cells. But the prototype system has only been tested on small networks, often with simulated attacks. It has been used to *monitor* actual attacks on a test network on the Internet. In fact, the monitored attacks have provided the data for the attack simulator that is part of the cyber warfare C2 system. In large distributed C2 systems it is likely that a single KWC will not be able to monitor all of the regional CWC cells. Our prototype allows the KWC, or other KWC cell members, to be in as many CWC cells as they want to be a member of.

A general principle of all cells is that *a cell commander* (at whatever level) *controls who can be a member of the cell*. We assume here that the KWC authorizes all the CWCs to be members of the KWC cell, but there may be issues in coalition warfare that invalidate this assumption. In such a case, KWC cell members can be members of CWC cells without CWC cell members being members of the KWC cell.

Each regional CWC is supported by a number of other cells. Some of them shown in Figure 2 are the Intrusion Detection (ID) cells, the Intrusion Response (IR) cells, and the Vulnerability Assessment (VA) cells. We will not discuss the functions of all of these cells in detail. More information about the functions of some of these cells can be found in [1] and [2]. Another important core cell is the test bed cell. Our prototype cyber warfare C2 system includes an integrated test bed that members of the test bed cell can use for a variety of things including testing new ideas and experimenting with new applications as well as for testing new releases of software and integration testing of various IA capabilities. The test bed is an integral part of our operational capability.

The other two cells shown in Figure 2 are labeled Dynamic Cell C and Dynamic Cell D. As previously mentioned, dynamic cells are created as needed to support C2 tasks or operations. Any cell commander can approve the creation of a new cell. The new cell is considered to be at the level of command of the cell commander that authorized its creation. The cell commander

that authorizes the creation of a new cell is also responsible for authorizing those who will fill the role of cell commander for the new cell. The reader should understand that anyone with authorization, whether part of the C2 organization or not, can join a cell. For instance, an IA expert with a needed expertise (e.g. a university researcher) can be brought into a cell when the need arises. Such experts, possibly without security clearances, can be admitted to dynamically created virtual cells that have specifically been created to deal with a critical issue. It is intended that admission to the core cells is more restricted, but advance preparation such as the obtaining of security clearances and installing secure communication capabilities for a team of experts from other organizations could provide a reserve capability for crisis situations.

The dynamic cell C in Figure 2 is intended to indicate a cell that has been created for inter-regional collaboration on an intrusion detection problem that is currently affecting each region. The dynamic cell D is intended to indicate a cell that has been created for an offensive cyber warfare operation in region A. The operation is being run by a member of the region A CWC cell and the interest in the operation is high enough that a member of the KWC cell is monitoring the operation.

4. The Operational Model

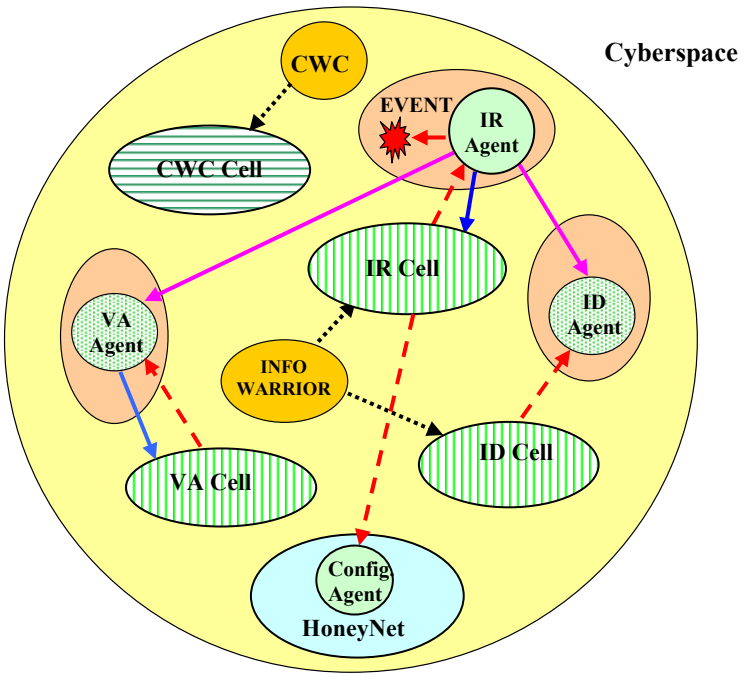
The *operational* model shown in Figure 3 is a high level depiction of the cyber warfare C2 model our prototype supports. The functionality indicated in Figure 3 is only a subset of the functionality of the prototype. But the following discussion of Figure 3 will indicate the dynamic strategy and anticipatory tactics of the model. Some of the operational functions of this model are:

- Cyber intelligence analyses e.g., intrusion event and attack signatures, intrusion event correlation, attack determination, status of C2 networks, cyber alerts from other organizations
- Cyber operations management e.g., maintaining a cyber operational picture, cyber order of battle display, and attack status display; determining COAs for responding to attacks and raising or lowering levels of protection
- Cyber operations planning e.g., managing honeynets (subnetworks of honey pots) to observe intruder strategy and tactics, development of cyber warfare strategy and tactics, COA development by attack type,
- Cyber operational control e.g., monitoring attacks and COAs, dispatching mobile agent patrols, relocating critical applications, and shepherding attackers into honeynets

Figure 3 shows some of the cells depicted in Figure 2 performing various cyber warfare functions. At the top of the figure we see that the CWC for the shift has joined the CWC cell (indicated by the dotted arrow). In the middle of the figure we see the IR cell, the VA cell, and the ID cell that support the CWC. The VA cell has dispatched a mobile agent patrol to search for vulnerabilities such as unauthorized modems or platforms that are not in compliance with IAVAs that have been issued. One of these agents is shown sending a message back to the VA cell to report its findings. The dashed arrows indicate mobile agent dispatching. The solid arrows indicate mobile agent messages being transmitted to one of the virtual cells. The oval that are not labeled indicate platforms within the C2 network where mobile agents may visit.

Figure 3 also shows that the IR cell has dispatched a mobile agent in response to some perceived intrusion event and that this agent is communicating information about the event back to the IR cell, and to other agents. The IR mobile agent is also shown stopping a process that is running on the platform it is visiting because the process appears to be hostile. The perceptive reader may be wondering how the IR mobile agent knows where the VA agent and the ID agent are so that it can send messages to them. The answer is that the IR agent does not know where the VA and ID agents are. Message passing in the prototype system is primarily handled using the publish and subscribe paradigm. With this paradigm, the publisher does not need to know where its subscribers are. When a mobile agent visits a node in the C2 network it notifies the local communications agent what category of messages it wants to subscribe to. Before leaving the node it collects all messages it has subscribed for at that node and then unsubscribes. It renews its subscription at the next node if it travels to another node.

Figure 3: Cyber Warfare C2 System Operational Model



A random algorithm, which also guarantees uniform coverage of C2 assets, controls the dispatching of agent patrols. The randomness insures that both intruders and legitimate users do not know when an agent will show up unannounced at any node. This makes it harder for intruders to defeat the built-in security measures of the system and for legitimate users to circumvent the security policies of the system. It also makes it difficult for adversaries to conduct intelligence preparation of the battlespace (IPB) of our C2 cyberspace because the mobile cells and mobile agents are continually presenting the adversary a different picture of the logical and physical organization of our C2 systems.

4.1 ID and Attack Identification

The ID cell receives both host and network intrusion incident information from a variety of sources. Internally it receives so-called *detects* from intrusion detection scanners, from host based ID systems, and from ID agents patrolling the C2 system. It also receives *event* information from firewalls within the C2 system. Beyond this, it receives *alerts* and other information about attacks from other organizations. All this information is correlated to determine if there are attacks underway or likely. Most of the correlation involves humans in the loop.

The correlated information is passed to the IR cell to determine what actions need to be taken. The operational model provides for correlation of incident information without first assembling it into a common database as is done in kinetic warfare C2 systems, e.g. track correlation in tactical air defense systems. Some existing intrusion event correlation systems like DISA's AIDE system first gather intrusion detection events from multiple sensors into a relational database. The correlation AIDE provides is done on the data in this relational database using SQL queries and data table sorts. The problem with this approach is that it is very time consuming and does not scale well due to the fact that many SQL queries are proportional to an exponential power of the number of elements in the data tables. As the size of C2 networks grow the volume of incident information becomes enormous. The time it takes to assemble this data into a relational database and correlate it can allow attackers to be finished with their attack before the incident information can be correlated to detect the attack.

In addition to using mobile agents for detecting intrusion events, the ID cell can dispatch agents to reconfigure sensors, read system logs and messages, and maintain any intrusion detection databases containing important historical packet data. This type of data is ultimately used to coordinate and further automate the intrusion detection process.

4.2 VA and Attack Simulation

Vulnerability assessment in our prototype is done using several different methods. Classical network scans are done using scanning tools that are available within the VA cell. VA mobile agent patrols provide data on vulnerabilities that are found during agent visits to specific platforms. Attack simulation tools are available from within the VA cell to study the results of simulated attacks against the C2 networks. These simulations are based on data from multiple sources. First there is the attack data that we have gathered from our own modest honeynet. Second, there is attack data that we have received from other organizations. In the future, we anticipate that our test bed will provide more sophisticated attack simulations and the replaying of actual attacks for training and COA development.

Comprehensive vulnerability assessment is likely to require a decision system. Unlike traditional expert systems, decision systems can provide results with incomplete and even ambiguous data. This is particularly important since the data available can be sparse and volatile. For example, a wireless network card shared by several individuals presents different system configurations with different vulnerabilities all with the same MAC address. Public wireless access points add yet another dimension to the vulnerability equation. These are all issues we are currently working on and believe we will have some capability in these areas in the future.

Mobile agents designed to query USB hubs and modems, examine patch history, and perform local port scans are currently being developed for use in the VA cells.

4.3 IR and Counter Attacks

The IR cell performs several functions. IR is a broad area that includes, immediate responses to contain attacks if possible, less immediate responses to stop attacks and to inform the CWC of what happened and what measures were taken, counter attacks with the CWC and/or KWC approval, COA recommendations to the CWC and/or KWC, and damage assessments of attacks. The IR cell members are also involved in cyber warfare strategy and tactics development and COA development.

The IR cell can dispatch mobile agents to kill selected processes, reconfigure firewalls, and if necessary, remove or restrict suspicious users. In addition, offensive attack or covert agents could be deployed, within the context of our cyber C2 concept of operations, in response to certain cyber threats. We have not developed mobile agents that can operate outside our own network boundaries, but regard the ability to do so as being very feasible if it were required during a kinetic warfare battle.

4.4 Test Bed Cell

One feature of cyber warfare that is different from kinetic warfare is this: in kinetic warfare engineers and scientists primarily support the war fighters but in cyber warfare the war is fought by the engineers and scientist. They are the ones that have created the cyber battlespace and the ones who understand it best. Our model of warfare calls for the operational integration of kinetic war fighters and engineers and scientists. This integration requires joint training of kinetic war fighters with the engineers and scientist. It also requires the development of COAs that integrate kinetic war fighting with cyber war fighting. How this will all proceed is currently largely unknown. Experiments in this area are going on in the war colleges and elsewhere. We envision that the integrated test bed will play an important role in determining the best way to train jointly and to evaluate COAs that integrate kinetic and cyber warfare.

The Test Bed Cell is one of the core cells because, at least for some time to come, it will be needed to make dynamic adjustments to this integration both in exercises and in continuing operations. It is envisioned that the test bed cell will provide the interface to a separate testing environment that can monitor the operational environment, and which, in an emergency can be used as a backup system if the operational system fails.

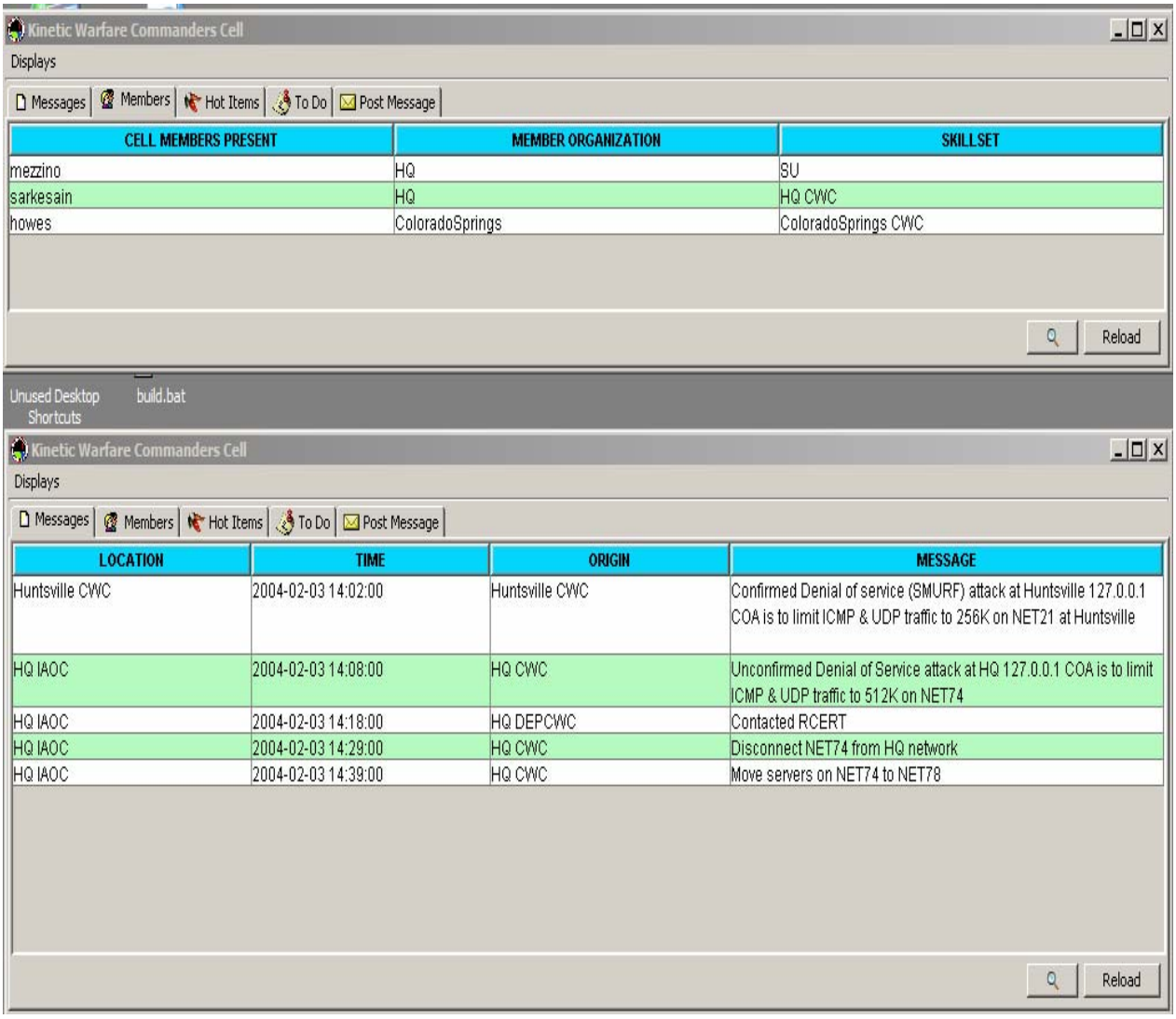
The test bed cell will have a cell commander just like the other core cells. The cell commander for the test bed cell will be the chief engineer for the cyber warfare C2 system. Test bed cell members will primarily be computer scientists, software engineers, network administrators, and electrical engineers. But the test bed cell will also include kinetic warfare specialist that have a deep knowledge of kinetic warfare operational art.

5. Prototype Cyber Warfare C2 System

Our prototype cyber warfare C2 system was preceded by a demonstration cyber warfare C2 system that we refer to as the “demo version.” The demo version consisted of a few hundred PERL scripts and some HTML code whereas the prototype version is written primarily in Java and runs over a publish and subscribe messaging infrastructure called *Splice* which is a product of Thales, Netherlands. The specification for the demo version [2] was in the form of a users manual. We used the User Manual approach for developing the requirement specification, i.e. we conceptualized what such a system would do, what its inputs would be, what its displays would

look like, and wrote a users manual as if the system already existed. We then turned the users manual over to the developers who developed a demo version that behaved just like the users manual said explained.

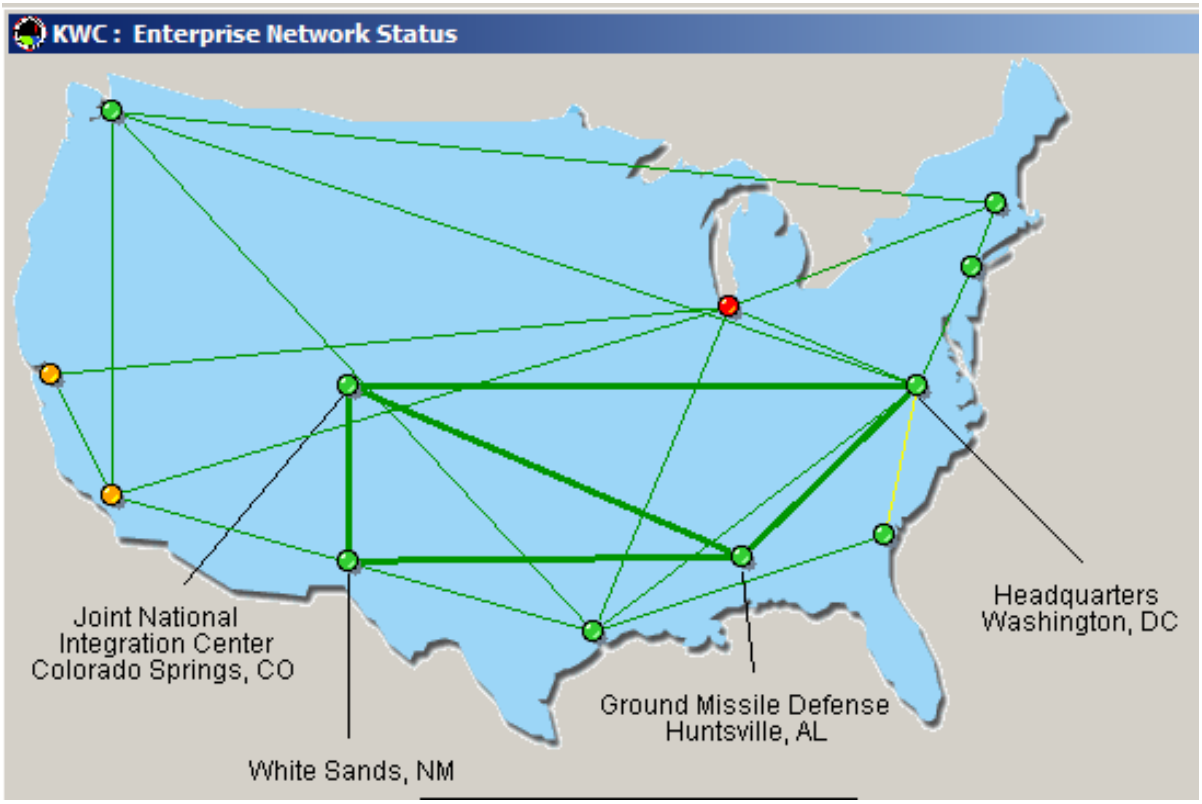
Figure 4: KWC Cell Windows



We knew in advance that we could not do everything we wanted to do in PERL, but we needed a demonstration version as soon as we could get it, so we opted for rapid development at the cost of functionality and performance. The demo version allowed us to demonstrate our concept and get the funding we needed to build a real prototype. Work on the prototype began about a year ago. We recently retired the demo version since the prototype can now do more than the demo version, and it is more robust, is much closer to what a production version will be like, and has many security features that were probably impossible with PERL. Using Java for the development language allowed us to use the *Java 2 Platform Security Model* [3] that is useful in securing individual computational platforms. It enables the protected execution of computer code received from remote (and possibly untrusted) network locations.

Java was designed to be platform independent in the sense that mobile code written in Java can execute on any platform where the Java run-time environment exists, without recompiling the code. Java supports platform independence by creating *Java Virtual Machines* (JVMs) that are a type of container, or computer within a computer, in which to execute Java programs. The JVM is also the mechanism that provides secure execution of foreign code by not allowing the execution to spill outside of the JVM, or by restricting access to resources from within the JVM. The Java security model provides facilities for authentication, access control, data integrity, data confidentiality, and non-repudiation within and among JVMs by providing encryption facilities, security policy objects, permission objects, access control manager objects, and security manager objects. These features are well documented in several books on Java such as [4].

Figure 5: The Enterprise Network Display

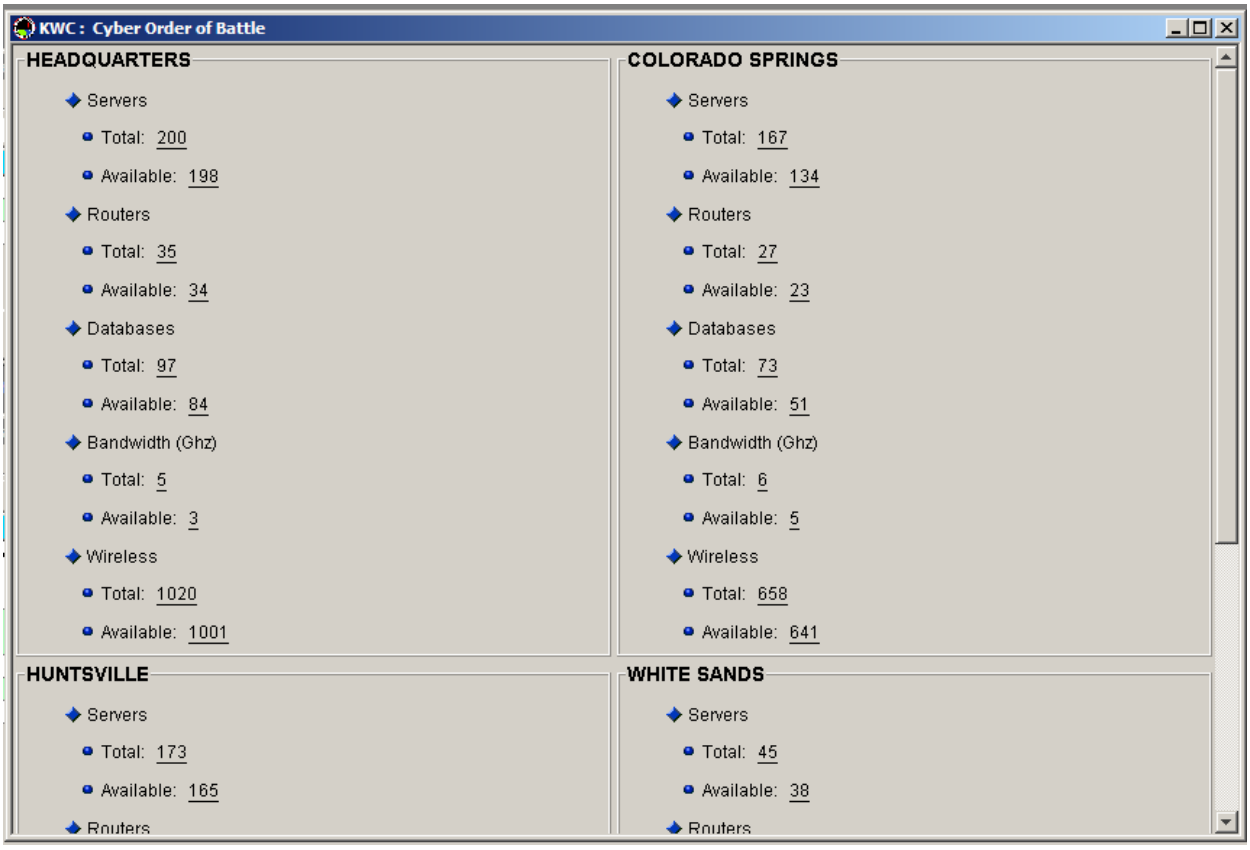


Figures 4 through 10 show some of the types of displays that are produced by our prototype cyber warfare C2 system. The first window in Figure 4 shows which cell members are currently present in the cell, what organizations they represent, and the member's skill set or function. The second window provides a log of the recent message traffic within the cell.

The Enterprise Network display in Figure 5 allows a cyber warrior to get an overview of the networks that make up the cyber battlespace. This display has a "drill-down" capability. By clicking on one of the nodes in the network, the regional network represented on the display as a node is displayed. This capability can be extended via the system configuration utilities to allow drill-down to individual LANs. The network links and nodes are color coded by the colors red, yellow, and green. Red means "unavailable," yellow means "partially available," and green means "fully available."

The fact that the Chicago region is designated by the red node means that message traffic to and from the Chicago region is essentially unavailable, even though some messages may get through. It may also mean that messages that do get through may be corrupted and therefore the integrity of any message received from that region cannot be relied upon. The fact that the Los Angeles region is designated by a yellow node means that message communication with that region is degraded, but that the integrity of the messages getting through can be relied upon.

Figure 6: Cyber Order of Battle Display



The Cyber Order of Battle display shown in Figure 6 is analogous to a kinetic warfare order of battle. It provides an overview of the cyber resources available to the KWC and the CWCs. Like the Enterprise Network display, it has a drill-down capability. By clicking on the **Servers** button in the Huntsville region of the display, it displays a list of all the servers in the Huntsville region color coded by red, yellow, and green (unavailable, partially available, and available). If you then click on an individual server in that list, it will display why its status is unavailable or partially available. If its status is shown as available, this second level of drill-down does not apply.

Figure 7: Enterprise Status Display after a Real-Time Network Scan

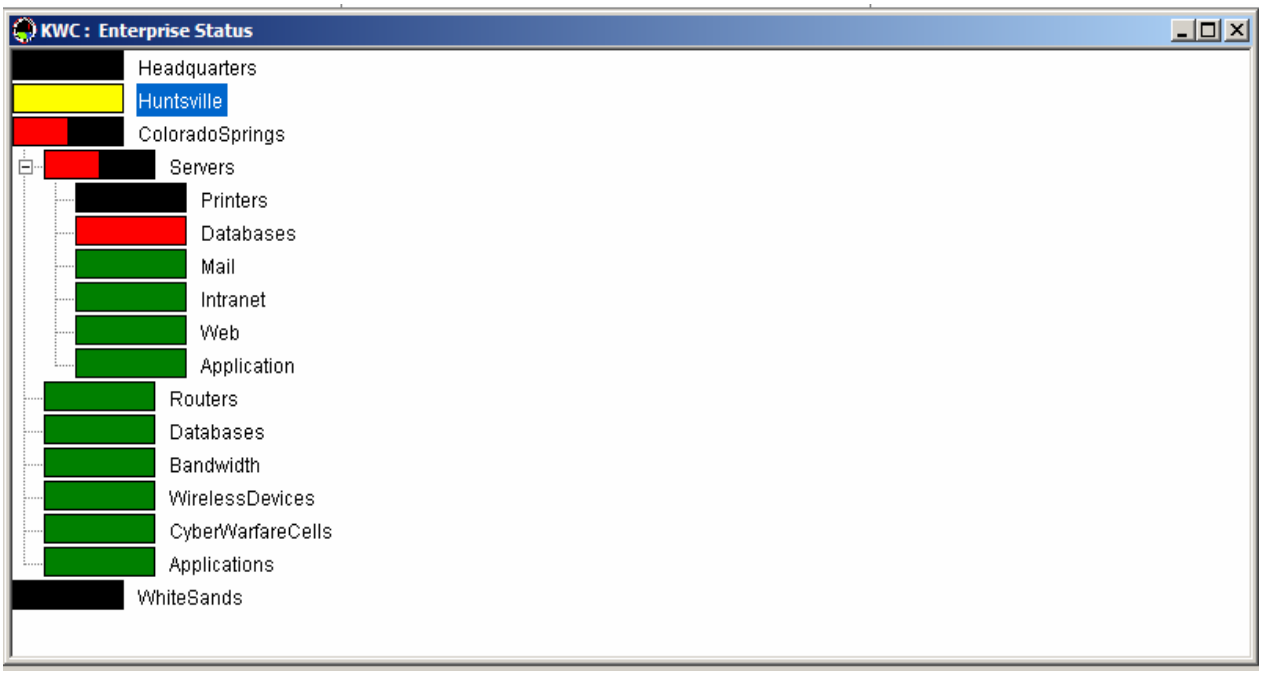
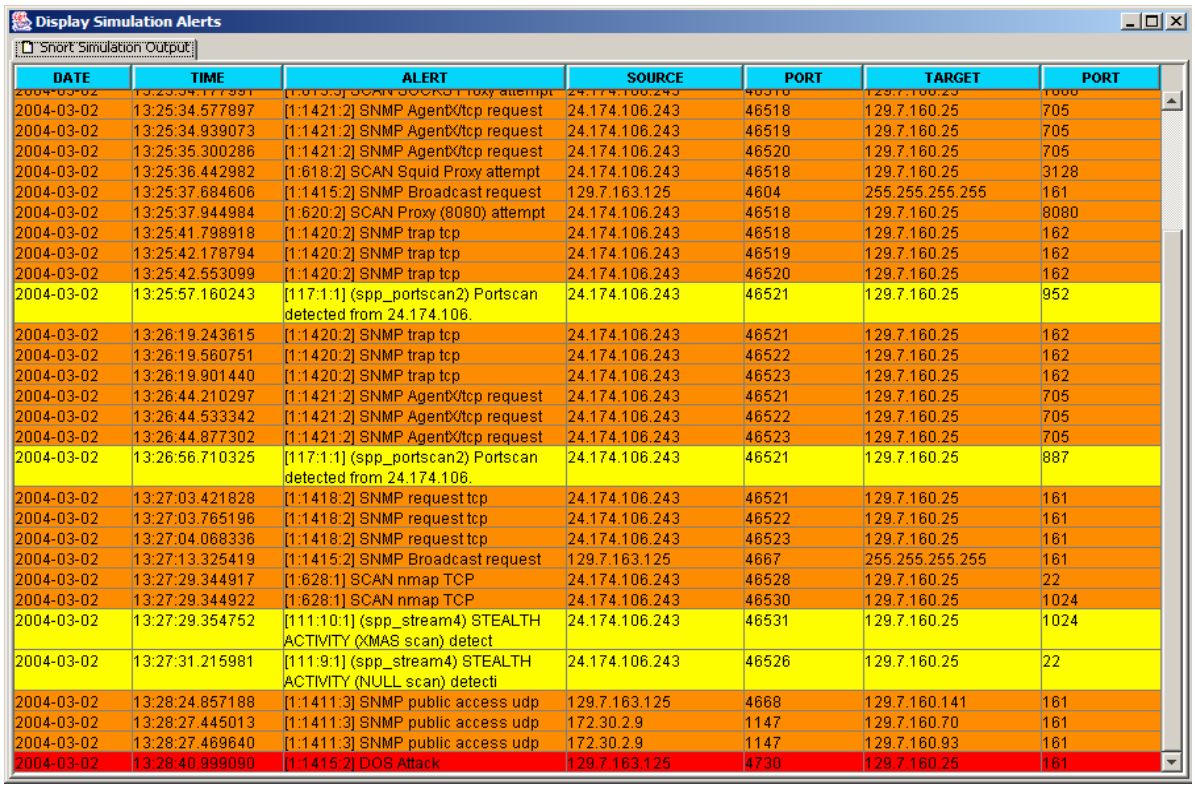


Figure 7 shows the Enterprise Status display after a real-time scan of the network has been performed. The Colorado Springs drill-down is showing that at least one database is unavailable and at least one printer has no data (color coded as black). Figure 7 also shows that Huntsville is not drilled down but we can still see that some of the cyber resources at Huntsville are only partially unavailable.

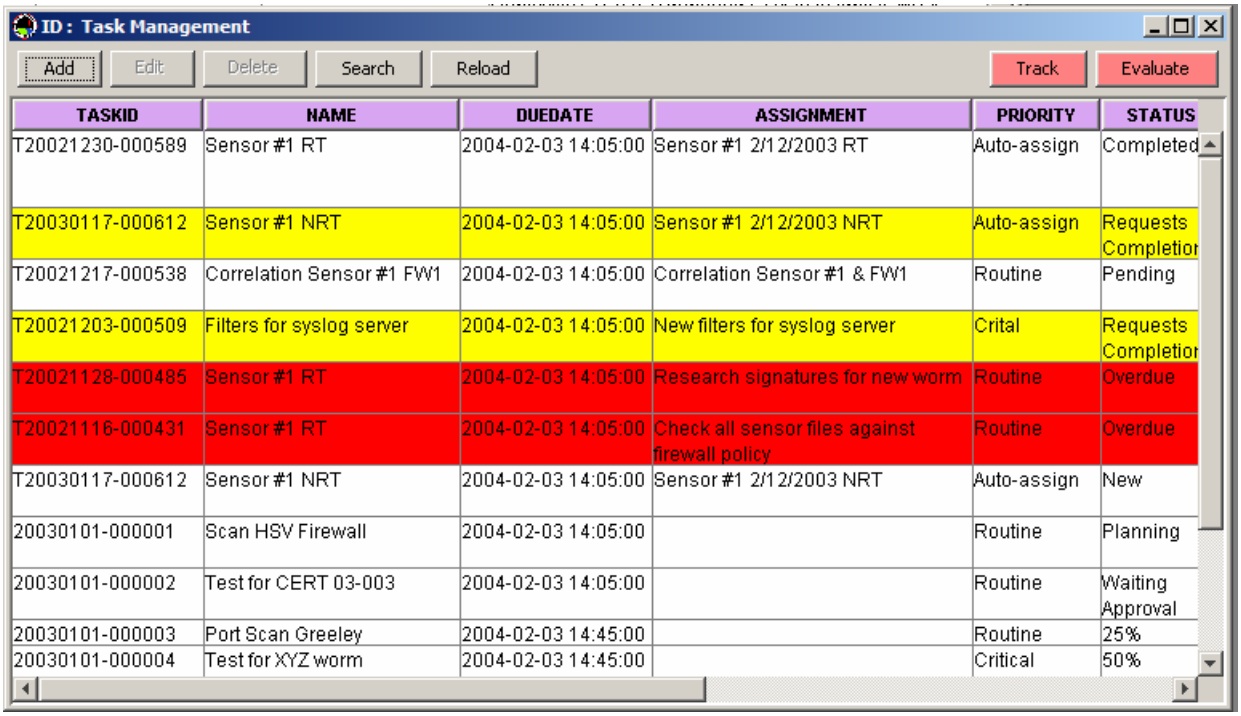
Figure 8 shows a Simulation Alert screen after a simulated DoS attack at Huntsville. The intrusion events are color coded in yellow to indicate the significant events that are occurring during the attack. The intrusion event that is color coded red at the bottom of the alert screen indicates that at this point some cyber resource becomes unavailable.

Figure 8: Drill-down of Simulated Attack



DATE	TIME	ALERT	SOURCE	PORT	TARGET	PORT
2004-03-02	13:25:34.577897	[1:1421:2] SNMP AgentX/tcp request	24.174.106.243	46518	129.7.160.25	705
2004-03-02	13:25:34.939073	[1:1421:2] SNMP AgentX/tcp request	24.174.106.243	46519	129.7.160.25	705
2004-03-02	13:25:35.300286	[1:1421:2] SNMP AgentX/tcp request	24.174.106.243	46520	129.7.160.25	705
2004-03-02	13:25:36.442982	[1:618:2] SCAN Squid Proxy attempt	24.174.106.243	46518	129.7.160.25	3128
2004-03-02	13:25:37.684606	[1:1415:2] SNMP Broadcast request	129.7.163.125	4604	255.255.255.255	161
2004-03-02	13:25:37.944984	[1:620:2] SCAN Proxy (8080) attempt	24.174.106.243	46518	129.7.160.25	8080
2004-03-02	13:25:41.798918	[1:1420:2] SNMP trap tcp	24.174.106.243	46518	129.7.160.25	162
2004-03-02	13:25:42.178794	[1:1420:2] SNMP trap tcp	24.174.106.243	46519	129.7.160.25	162
2004-03-02	13:25:42.553099	[1:1420:2] SNMP trap tcp	24.174.106.243	46520	129.7.160.25	162
2004-03-02	13:25:57.160243	[117:1:1] (spp_portscan2) Portscan detected from 24.174.106.	24.174.106.243	46521	129.7.160.25	952
2004-03-02	13:26:19.243615	[1:1420:2] SNMP trap tcp	24.174.106.243	46521	129.7.160.25	162
2004-03-02	13:26:19.560751	[1:1420:2] SNMP trap tcp	24.174.106.243	46522	129.7.160.25	162
2004-03-02	13:26:19.901440	[1:1420:2] SNMP trap tcp	24.174.106.243	46523	129.7.160.25	162
2004-03-02	13:26:44.210297	[1:1421:2] SNMP AgentX/tcp request	24.174.106.243	46521	129.7.160.25	705
2004-03-02	13:26:44.533342	[1:1421:2] SNMP AgentX/tcp request	24.174.106.243	46522	129.7.160.25	705
2004-03-02	13:26:44.877302	[1:1421:2] SNMP AgentX/tcp request	24.174.106.243	46523	129.7.160.25	705
2004-03-02	13:26:56.710325	[117:1:1] (spp_portscan2) Portscan detected from 24.174.106.	24.174.106.243	46521	129.7.160.25	887
2004-03-02	13:27:03.421828	[1:1418:2] SNMP request tcp	24.174.106.243	46521	129.7.160.25	161
2004-03-02	13:27:03.765196	[1:1418:2] SNMP request tcp	24.174.106.243	46522	129.7.160.25	161
2004-03-02	13:27:04.068336	[1:1418:2] SNMP request tcp	24.174.106.243	46523	129.7.160.25	161
2004-03-02	13:27:13.325419	[1:1415:2] SNMP Broadcast request	129.7.163.125	4667	255.255.255.255	161
2004-03-02	13:27:29.344917	[1:628:1] SCAN nmap TCP	24.174.106.243	46528	129.7.160.25	22
2004-03-02	13:27:29.344922	[1:628:1] SCAN nmap TCP	24.174.106.243	46530	129.7.160.25	1024
2004-03-02	13:27:29.354752	[111:10:1] (spp_stream4) STEALTH ACTIVITY (XMAS scan) detect	24.174.106.243	46531	129.7.160.25	1024
2004-03-02	13:27:31.215981	[111:9:1] (spp_stream4) STEALTH ACTIVITY (NULL scan) detecti	24.174.106.243	46526	129.7.160.25	22
2004-03-02	13:28:24.857188	[1:1411:3] SNMP public access udp	129.7.163.125	4668	129.7.160.141	161
2004-03-02	13:28:27.445013	[1:1411:3] SNMP public access udp	172.30.2.9	1147	129.7.160.70	161
2004-03-02	13:28:27.469640	[1:1411:3] SNMP public access udp	172.30.2.9	1147	129.7.160.93	161
2004-03-02	13:28:40.999090	[1:1415:2] DOS Attack	129.7.163.125	4730	129.7.160.25	161

Figure 9: Task Management Display



TASKID	NAME	DUEDATE	ASSIGNMENT	PRIORITY	STATUS
T20021230-000589	Sensor #1 RT	2004-02-03 14:05:00	Sensor #1 2/12/2003 RT	Auto-assign	Completed
T20030117-000612	Sensor #1 NRT	2004-02-03 14:05:00	Sensor #1 2/12/2003 NRT	Auto-assign	Requests Completion
T20021217-000538	Correlation Sensor #1 FW1	2004-02-03 14:05:00	Correlation Sensor #1 & FW1	Routine	Pending
T20021203-000509	Filters for syslog server	2004-02-03 14:05:00	New filters for syslog server	Crital	Requests Completion
T20021128-000485	Sensor #1 RT	2004-02-03 14:05:00	Research signatures for new worm	Routine	Overdue
T20021116-000431	Sensor #1 RT	2004-02-03 14:05:00	Check all sensor files against firewall policy	Routine	Overdue
T20030117-000612	Sensor #1 NRT	2004-02-03 14:05:00	Sensor #1 2/12/2003 NRT	Auto-assign	New
20030101-000001	Scan HSV Firewall	2004-02-03 14:05:00		Routine	Planning
20030101-000002	Test for CERT 03-003	2004-02-03 14:05:00		Routine	Waiting Approval
20030101-000003	Port Scan Greeley	2004-02-03 14:45:00		Routine	25%
20030101-000004	Test for XYZ worm	2004-02-03 14:45:00		Critical	50%

The Task Management screen shown in Figure 9 has been launched from the ID Cell but the task management tool is one that is common to all the cells in the cyber C2 system. It is used to keep up with the assignments made to the cell members by the cell commander. As can be seen from the screen shot in Figure 9, the slider bar at the bottom of the screen indicates that there are other fields that where not captured when the screen shot was made. One of these other fields is a field that records who is responsible for each task. The tasks are color coded to give a rapid indication of which tasks require attention. Yellow entries indicate tasks that are requesting completion status and red entries mean that a task is overdue.

While the cyber C2 applications such as the ones indicated in the above figures are written in Java, the publish and subscribe messaging infrastructure we are currently using is written in "C" but has a Java interface. The Java security features will be used to secure these applications as they run on the various platforms within the cyber C2 network. But the overall cyber C2 system security is only as strong as the security of the operating systems on the platforms on which it runs and the messaging infrastructure that provides the distributed system communication capability. The publish and subscribe messaging system we are currently using is implemented as a family of cooperating distributed agents that handle the multicast communications of publishers and the building of memory resident databases for subscribers. The original version of Splice was conceived by Maarten Boasson [8]. There are several other publish and subscribe middleware products on the market, but the Splice implementation had many features that were particularly attractive and our testing showed it to be extremely robust and efficient.

We are currently designing a secure publish and subscribe infrastructure for use with future versions of our cyber warfare C2 System. We hope to be able to prove the correctness of this new publish and subscribe messaging infrastructure using the methods described in [5], [6], and [7]. To date we have been able to prove the correctness of a basic publish and subscribe system that implements a distributed heart-beat system [7] using the *temporal logic of behaviors* (TLB). This leads us to believe we will be able to prove the correct behavior of the secure publish and subscribe system if we can specify it in TLB. We are currently trying to write such a formal specification for the new secure publish and subscribe infrastructure we are developing.

6. References

[1] Howes and Sarkesain, *Dynamic Virtual Communities and Mobile Agent Architecture*, Proc. 3rd Annual IEEE Information Assurance Conference, WestPoint, NY, June 2002.

[2] N. Howes, et. al, *Cyber Warfare Command and Control System Users Manual*, Draft IDA document, July 2003.

[3] Li Gong, *Inside Java 2 Platform Security*, Addison Wesley, Reading, MA, June 1999.

[4] Felten and McGraw, *Securing Java*, John Wiley and Sons, New York, NY, 1999.

[5] N. Howes, *State Space Topology and a Theory of Convergence for Temporal Logics*, Proc. Conf. on Topology in Computer Science, City College, NY, 2002

[6] N. Howes, *The Temporal Logic of Behaviors*, to appear.

[7] N. Howes, *Temporal Logics for Distributed Systems*, to appear.

[8] Maarten Boasson, et. al., *A software architecture for distributed control systems and its transition system semantics*, Proceedings of the ACM Symposium on Applied Computing (SAC '98), Atlanta, pages 159-168. ACM press, 1998