Guarding the Cybercastle in 2020

by Todd McDonald, Bert Peterson, Dan Karrels, Todd Andel, and Rick Raines

Abstract

Monitoring and defending current and future US Air Force (USAF) networks will require a synergy of emerging technologies and some degree of novelty in both acquisition and operational art. In this article, we examine possibilities for future distributed defensive architectures and consider them in light of security and trust. As we consider current research efforts devoted to information and network security, we catch a brief glimpse at what the future cyber defense landscape, or "Cybercastle," may look like.

Disclaimer

The views expressed in this article are those of the authors and do not reflect the official policy or position of the US Air Force, Department of Defense, or US Government.

Introduction

Many of the commercial systems found in the developed and developing world depend on computers and communication networks for the ability to conduct enterprise activities. Similarly, the Department of Defense (DoD) has overlaid major operational capabilities on information networks that support command, control, and communications (C3) at various levels. In 2005, the USAF officially recognized the criticality of the information domain as a strategic warfighting resource and redefined its mission statement to include "deliver sovereign options for the defense of the United States of America and its global interests—to fly and fight in Air, Space, and Cyberspace". [1] Secretary of the Air Force Michael W. Wynne subsequently reinforced this vision with the creation of a Cyberspace Command (AFCYBER). [2] The DoD has recently refocused its formal definition of cyber as "a global domain within the information environment consisting of the interdependent network of information technology infrastructures, including the Internet, telecommunications networks, computer systems, and embedded processors and controllers," consistent with presidential cyber security policy. [3] In any current military understanding, cyber defense squarely encompasses computers (embedded and standalone) and their interconnectivity.

Currently, we are seeing a flood of threats to the electronic infrastructure of governments around the world, including our own. As we consider the landscape of the USAF network infrastructure over the next decade, we may also consider possibilities for defending that infrastructure in a holistic, secure, and trusted manner. It makes sense as well that whatever revolutionary changes we may ultimately consider, the entire panorama of the defense industrial base (DIB) and our national commercial interests are envisioned. The Cybercastle, in this view, encompasses (but is not limited to) portions of the Internet that support military and high-encryption systems, DoD intranets, external information systems, wireless/radio communications systems, and infrastructure control systems using Supervisory Control and Data Acquisition (SCADA) systems.

Today, nearly 10% of all Internet nodes belong (unknowingly) to a malicious multi-agent system whose owner waits for a high bidder to make use of its services. As Internet usage worldwide continues to grow, and with average users unaware of their vulnerability to assimilation into a malicious C3 network, the next decade promises huge challenges directly rooted in cyber-network defense and protection. To deal with the possibilities for cyberterrorism in the ongoing Global War on Terrorism (GWOT) or possible malicious attacks of nation-state actors against the Cybercastle, we turn our attention to high-level goals for building defensive systems. What will the castle look like a decade from now? How strong will its walls be, or how strong do we *need* the walls to be in light of the veracity of those on the other side of the moat? What is a wall or a moat in cyberspace, given that threats can also come from inside the network? We discuss thoughts on these topics and give some insight on what technological advances or prices will most likely be paid to ensure the Cybercastle's resilience.

Currently, we are seeing a flood of threats to the electronic infrastructure of governments around the world, including our own. As we consider the landscape of the USAF network infrastructure over the next decade, we may also consider possibilities for defending that infrastructure in a holistic, secure, and trusted manner.

Defensive Cyber Platforms

In the domain of space and air, the USAF focuses strategic and operational capabilities across a range of platforms, where we define a platform as a delivery vehicle for some suite of weapons (bullets from a Gatling gun, air-launched cruise missiles, air-to-air missiles, *etc.*) that support some set of missions and objectives (suppression of enemy air defense, close air support, air superiority, *etc.*). In some cases, the platform itself provides the actual strategic advantage; in other cases, the weapons (or payloads) delivered by the platform are of greater significance. Platforms, in general, exhibit long service life, incur large capital investment, support a variety of missions (consider the evolution of the B-52 bomber), and undergo intense scrutiny to guarantee reliability or operational qualities. Payloads, on the contrary, emerge from rapid development lifecycles and achieve specific effects in some tactical, operational, or strategic context.

Targeting cyber as an operational domain will produce a wide variety of cyber platform and payload manifestations over the next decade. Though cyber as a warfighting domain is not limited to information networks and their underlying capabilities, we abuse the term slightly so we may consider platforms that might support holistic network defense capabilities. In considering defensive cyber platforms, we use the term "Cybercraft" to embody the notion of a delivery platform for C3 defensive capabilities. In our technical paper we posed the first conceptual use of Cybercraft as an autonomous, intelligent agent that accomplishes military purposes across a wide variety of electronic-based media. [4] The Air Force Research Laboratory (AFRL) Information Directorate has continued the vision with a project aimed at furthering basic research areas that support platform and payload integration for defensive missions.

As AFRL's research partner in this endeavor, the Air Force Institute of Technology (AFIT) is considering a broad range of development possibilities for future defensive cyber platforms and is supporting technologies for the Cybercraft project itself. While keeping one eye on the needs of current USAF network defenders and keeping the other eye on the horizon to see what the art of the possible may be, we consider what the platforms used to defend the Cybercastle in the future may look like.

Building the Cybercastle

The defense of USAF networks in the future must not rely on the notion that a Maginot line exists that is beyond penetration. As history clearly reveals, the ways around a wall are more numerous than one might assume. It is common knowledge that enemies exist almost as abundantly within walls as they do outside of walls. For example, the Computer Emergency Response Team (CERT) Coordination Center found that the vast majority of the government insider incidents (90%) were caused by current employees, and most (58%) were people in administrative positions requiring limited technical skill. [5] Despite proof of security or empirical demonstration of our finest defensive tools, acts prompted from social networking attacks that are foisted on non-malicious insiders can void the best technical layers of defense.

The defensive cyber platforms of the future must have several qualities that will provide them freedom to maneuver and operational resilience in the face of such adversarial waters, whether the attempts to compromise mission activities come from inside or outside the network infrastructure. We envision platforms with the ability to execute a wide variety of generic capabilities using a common, payload-based framework. Until we find a more appealing future abstraction, distributed multi-agent systems (DMAS) is a close picture for what this defensive architecture could look like: compositions of light- to heavy-weight agent components that securely communicate, collaborate, and respond to cyber attacks of a wide variety. Because this cyber DMAS will defend existing and future military C3 systems, we have additional constraints that typical commercial networks may not be concerned with. Namely, cyber platforms must operate under tighter security, ensure fault tolerance and self-healing, and ultimately affect human life in some way (*i.e.,* failing to protect critical mission systems).

Figure 1 depicts, in standard Unified Modeling Language (UML) format, a conceptual domain model of interest to consider future defensive architectures. At a basic level, *platforms* serve to protect specific *host* nodes with a network or IP-based context (desktop machines, routers, hand-held devices, radio equipment, *etc.*). At a basic level, platforms possess *state* and execute *payloads* of various kinds: *sensors*, *behaviors*, and *effectors,* just to name a few. Payloads provide the touch point to the *environment*, which we define as the collective space of hosts connected to hosts *via networks* where hosts are controlled by some *OS* that runs *applications*. Payloads gather data about environmental components or alter the environment through specific effects. We define information about the environment and the conceptual glue that binds platforms and payloads to hierarchical levels of command and control as state, and we use behavior payloads to describe decisionmaking engines that bring logical correlation with sensor conclusions. We note that platforms share some state in a global context when payloads need to collaborate and keep other's knowledge local when payloads need only host-based context.

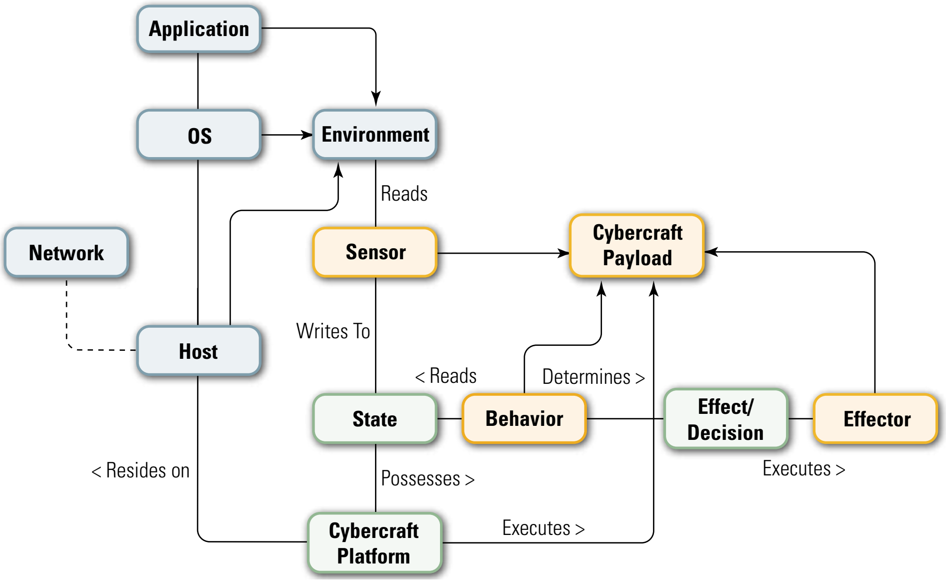


Figure 1 Conceptual Defensive Cyber-Network Domain Model

Much of the basic technology and defensive tools for our next-generation architecture exists in some form today, whether in commercial products or academic prototypes, and many research areas already give us considerations for platform design choices or specific payload configuration. Because basic functionalities—such as virus checkers, trust frameworks, trusted hardware components, self-repairing application environments, host-based integrity checkers, malware detection suites, and intrusion detection systems—all exist in some form currently (many immature but nonetheless demonstrable), one important question remains: what would a future defensive cyber platform most need so that we may rapidly, reliably, and securely integrate these technologies over time? In other words, what is the true revolutionary idea that our future defensive systems will need that our current tool suites do not provide?

We begin to answer these questions by considering that any single available technology is currently limited when used alone or used in a defensive vacuum. Namely, how do we measure the strength of any given technique when faced with an adversary that has subverted the OS or has taken control of the network infrastructure at some fundamental, administrator-privileged level? We can visualize this conundrum further by considering a modern virus checker that defends a system correctly as long as its signature-based algorithms detect and prevent *known* software threats from running maliciously. Unfortunately, "we do not know what we do not know," and this reflects more poignantly in our signature-based defensive mechanisms. In addition, these naive systems fail without having the full context of an attack. If the attack is "low and slow" or highly distributed and dynamic, we should focus on the context of the attack, not the manifestation of it.

Assuming an adversary does not successfully install an administration-level root kit with power to alter kernel-level operations and assuming we have a correct and current signature set to evaluate possible threats, a virus checker may provide defense along discrete attack vectors. Unfortunately, we cannot guarantee any of our base assumptions with any consistent measure. All application-level, virtual-level, and OS-level defensive techniques share this common weakness: they can be lied to by any of the underlying hardware components of a system or by the operating system that manages those components if an adversary gains root-level host access. As Figure 1 depicts, the notable difference of future cyber defensive platforms should be their independence from the rest of the environment in which they protect and operate. This concept depicts the only reliable way around a possible cycle of deception. We liken this concept to the military notion of taking the "high ground," and we call this cyber high ground the *root of trust*, which forms our number one priority in building the walls of the future Cybercastle.

Secure Root of Trust

Secure root of trust as a platform quality combines three notions: security (or self-defense), trust, and fundamental host context. The ability for platforms to be on the high ground in comparison to the possible level of attacks that assault them goes without saying; likewise, when we hold the high ground, we do not want to yield this ground to the adversary or open up separate attack vectors that put this privilege in a position of compromise. A self-defense guarantee centers on the ability to verify and validate that some hierarchy of platforms can keep and hold the *cyber high ground*, even if one (or a number) of the platforms is in physical or operational control of an adversary. This defense guarantee logically includes a wide range of mechanisms from hardware-based physical protection schemes to protocol-level proofs of security where distributed cryptographic voting schemes may be employed.

Trust is an elusive concept because its definition is rooted in social concepts as opposed to technical. Despite the overloading of the term itself, we can use trust to express a quality that military commanders make quite frequently: an objective dependability (whether by mathematical proof or demonstrated testing) that a system will perform according to its specifications, even though negative consequences can occur. For our look into the future, we also use trust to describe high confidence that an adversary (whether inside or outside) cannot subvert the operation of a fleet of cyber defensive platforms. Depending on the description and expression of our security parameters, we may have some varying degree of trust expressed in terms of achieved security levels. These two views of trust (a system will perform as expected and confidence in an adversary's cost of compromise is extremely high) provide some context to consider design tradeoffs for future defensive architectures in the cyber realm. We may also consider other agent-oriented aspects of trust more common to information quality and collaborative agent societies, but we believe the more fundamental concepts have greater dominance for long-term system design.

In another workshop paper, we provide a closer look at how we may overlap system requirements analysis, attack modeling, and trust model specification to concretize a trust analysis space for Cybercraft. [6] To achieve the cyber high ground in the future, we must marry trust and security at some fundamental host context level. In other words, we need to consider the highest level of trust and security relative to the degree of independence from the host in which a cyber defensive platform operates. We can visualize an application-level platform that relies completely on the integrity of an underlying OS versus a possibly virtualized-level platform that sits possibly at the same level as an OS in terms of privilege. The virtualized approach affords the possibility for greater independence, but it still does not offer the highest level of independence because it may be open to (undetected) subversion. To get below the level of the OS or below the level of any possible hypervisor/virtual OS that may execute on a host, we must position the cyber defense platform at some fundamentally lower level where physical access to the hardware remains unhindered or unobscured. We are investigating the tradeoff spaces and design possibilities for such a hardware-based root of trust that would support the addition of synergistic physical protection mechanisms while giving a generic payload execution environment for defensive C3 packages.

We note that a fixed, hardware-based root of trust is not a new concept—the Trusted Computing Group (TCG) has sought such expression for quite some time. [7] TCG specifications give an initial set of security-related building blocks to link trust with various computational components, such as storage, networking, software, and devices. In considering future defensive cyber platforms, the military environment provides the business case and operational bounds for feasibly acquiring and implementing hardware-based manifestations of Cybercraft. The movement toward implementable and procurable secure hardware solutions in the commercial market at least demonstrates the overlap with USAF and DoD goals to integrate such technology. A root of trust established in hardware for future defensive platforms will give us the basis for analyzing other trust-related concepts, such as collaborative and adaptive decisionmaking problems where agents must consider varying levels of trust over time with the information they gain from other agents.

Highly Scalable C3 Architecture

If defensive platforms are to be truly useful, they must integrate seamlessly into networks to share information and increase their level of autonomy. We see this relationship expressed in the domain model of Figure 2, where Cybercraft platforms are related to other platforms (for C3 purposes) with no explicit realization given. As Figure 3 depicts, a conceivable (real world) USAF platform deployment hierarchy may be focused on traditional organizational units located at bases and managed by higher level network operations security centers (Integrated Network Operations Security [INOSC]/ Air Force Network Operations Center [AFNOC]). To operate with DoD and USAF relevance, platforms must communicate and coordinate functionally in networks of extremely large size. With a goal of one million or more such platforms residing on the same network, issues of scale become a dominating factor.

Suffice to say, there are very few networks in existence today that accommodate large-scale *and* complex C3. Three examples of large-scale networks are the Internet, GNUTella, and KaZaA, and each has a different topology and performance. The Internet follows a topology governed by several power-law relationships, [8] GNUTella employs a Peer-to-Peer (P2P) architecture, [9] and KaZaA uses a Hierarchical Peer-to-Peer (HP2P) architecture [10]. Many existing multi-agent architectures in fact fall into one of five categories: power-law, P2P, hybrid (HP2P), multi-layer, and hub-based. We may see glimpses of the future Cybercastle by looking at multi-agent communication topologies that currently scale to hundreds of thousands of clients—and we briefly describe each one along with positive attributes for consideration.

Power-Law Functions—During 1997–1998, researchers performed experiments to observe the traffic patterns at various points of the Internet. [8] As the Internet grew by 45% during this period, the observations remained consistent through that growth. Scientists discovered that the structure of the Internet closely followed a power-law relationship among several graph topology metrics: the diameter of the graph, the out-degree of any node, and the average out-degree of the nodes of the graph. When displayed together on a log-log plot, these attributes formed linear relationships. This display yielded the notion of network topologies in large systems, in particular the Internet, as following a *power-law* relationship.

Because such power-laws have been shown to support large-scale networks, it makes sense to instantiate a network topology for defensive cyber platforms that follow these laws. However, building a network topology generator requires constructing a single node at a time, with the impact of each node on the overall structure remaining hidden until much later in the algorithm. Only once a sufficiently large number of nodes exists can the topology's macroscopic properties be measured—thus limiting the efficacy of such organization for future platform aggregations.

P2P Architecture—A P2P [9] network is one in which the nodes may establish multiple connections to other nodes. That is, the nodes are both client and server (or neither) and are free of the usual distinctions between the two. Rather, they communicate in a manner that best benefits the system objectives, without regard for the communication flow semantics of the client/server paradigm. A small P2P network is shown in Figure 3-a, illustrating that each node connects to as many other nodes in the network as deemed necessary. As we increase the number of connections per node, we reduce latency between source and destination nodes, but we also increase processing and communication burdens on all nodes along the source to the destination path.

The spanning form of a P2P network also nicely facilitates fault tolerance, where nodes may part and join the network dynamically and without warning. However, suboptimal network growth may adversely affect this fault tolerance. As a P2P network grows, it remains difficult to maintain a given set of performance metrics—and easily results in increased processing or bandwidth burdens on each node. Distinguishing between different types of nodes—*i.e., super* peer nodes—may offer a better hybrid approach.

Hierarchical P2P (HP2P) Architecture—An extension to the P2P structure includes super peers or super nodes. [10] These super peers act as regional hubs, absorbing additional burdens of the network traffic and processing load for distributed search and communications. These hubs may be interspersed across the network, as in the case of hubs

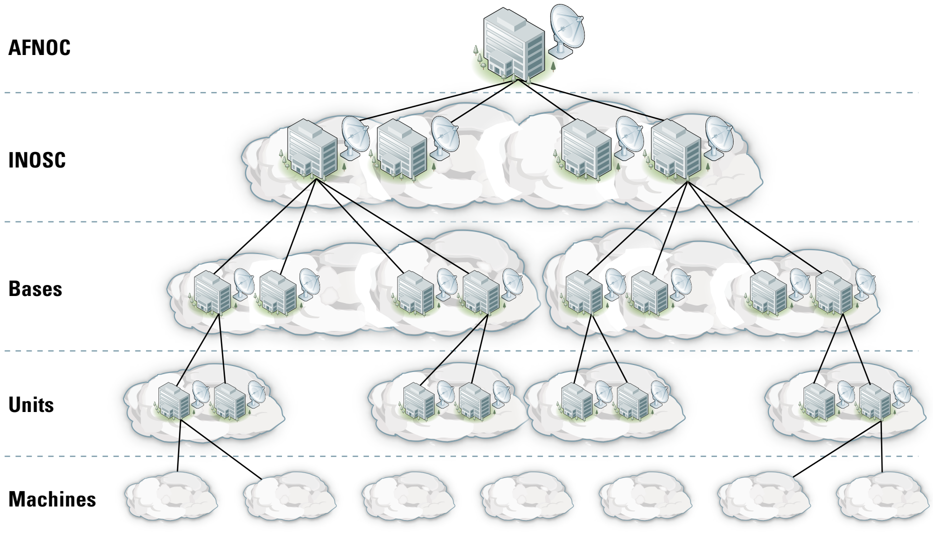
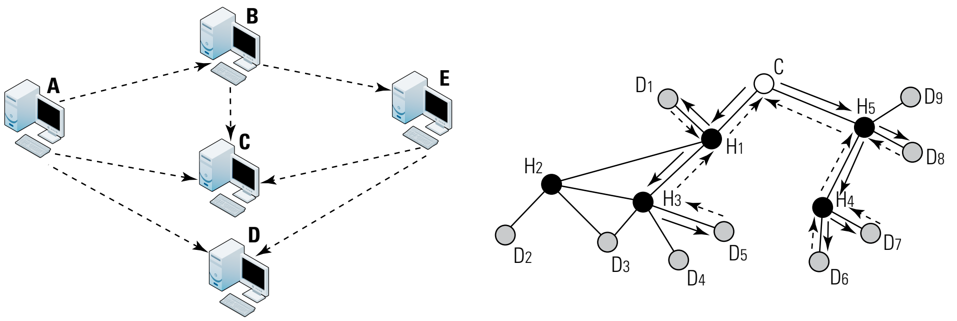


Figure 2 Conceptual Hierarchical Relationships Among Defensive Cyber Platforms



(a) Simple P2P

(b) Hierarchical P2P

Figure 3 Example P2P Network Configurations

controlling local clusters of regular nodes, or they may represent the bridges between layers of distinct P2P networks. The structure of the HP2P architecture improves on the P2P structure by incorporating clusters connected through super peers. These super peers provide additional bandwidth and processing capability similar to hubs, but because they are still multi-connected peers, they help retain the overall network structure. Such networks, as shown in Figure 3-b, are referred to as HP2P networks. P2P organizations scale to roughly tens of thousands of nodes, whereas HP2P has been shown to scale to approximately 50,000 nodes. [11] The benefit is that HP2P networks improve scalability by providing designated routes to other parts of the network. The clusters themselves are conveniently distinguished, allowing more intuitive segmenting of mission and information hiding from the rest of the network.

Botnet/Internet Relay Chat (IRC) Architectures—Often referred to as *botnets*, a *bot* (short for robot) *network* is a multi-agent distributed system that may be networked to other bots that possibly reside on a larger network (*e.g.,* the Internet) and is capable of being controlled by one or more users. One of the earliest uses of the term "bot" originates from IRC robots that were initially developed to run autonomously on IRC, allowing users to play games and performing simple authentication and chat channel protection functions. Botnets give us a small look at the power of organized agent societies that scale to large numbers of clients, and we can consider adaptations for cyber platform C3 topologies.

Though botnets were originally accessible only from IRC itself, they eventually offered the ability for users to log in and operate them with separate connections outside of IRC using proprietary protocols designed and tuned for C3. Thus, botnets were autonomous, operated in closed networks, and offered multiple interfaces for C3. They could also grow and shrink as bots became available or signed off. These features provide close corollaries to the desired features we wish to see in our future C3 defensive platform frameworks.

Hub-Based Botnet Architectures—Hub-based architectures focus on a central point of communication, called a hub, to which one or more leaf nodes connect. The hub is responsible for all of the routing and usually much of the processing in the network. This design is still very popular for network hardware because it is simple and efficient for small- to medium-sized networks. It suffers from scaling problems, complexity in dealing with interconnected hubs, and networks in which graph cycles exist.

Botnets operating with a hub-based C3 structure have been known to connect 7,000 or more infected machines to a single IRC network at once. The ability to command an entire botnet at once is a significant capability to the botmaster. IRC networks as a means of botnet C3 are useful because of their simplicity, availability, and cost (free to botnet authors). They have since faded in use for malicious botnets because all communications and IP addresses can be logged, leading to discovery of how the botnets work and what purpose they serve. In addition, the authors and users of these botnets become vulnerable because they too must connect to IRC to interact with their bots.

Fast-Flux Botnet Architectures—Fast-flux is a relatively new DMAS architecture, leveraged extensively by cybercriminals to support identity theft, spam email, and perform other types of computer-based crime. [12] It exploits the deliberate configurations of some networks to allow rapid changing of dynamic IP addresses, such as those that support cable and dial-up users. The goal is for a fully qualified domain name to have hundreds or even thousands of IP addresses assigned to it. The IP address to which the hostname resolves is then changed between the IP addresses frequently, with a very short time-to-live parameter. This prevents caching of the IP addresses by Domain Name Service (DNS) servers and forces DNS clients to continually recheck for the most recent IP address of the target hostname. Users attempting to connect to the destination host will connect to a different IP often—sometimes on a minute-by-minute basis. The fast flux domain name can then be used to maliciously build a reliable network of hosts that serve Web pages that may or may not be infected with viruses, Trojans, or other malware.

Fast-flux networks can be further extended to multiple layers, where infected hosts serve as redirectors to backend content servers, called *motherships*, which serve both Hyper-Text Transfer Protocol (HTTP) and DNS, providing virtual hosting for up to thousands of different Top-Level Domains (TLD). Fast-flux networks with more than 400,000 nodes are believed to exist, [13] presenting a possible real-world example of functioning large-scale systems and a possible glimpse of what C3 may look like for cyber defense platforms in the next decade.

Extensible Agent Architecture

Secure root of trust and C3 requirements create a large separation between normal business class networks on which a typical DMAS might operate and the military context in which cyber platforms of the future will operate. As we discussed in our workshop paper, the uniqueness of the agent architecture for the future Cybercastle may also create a rich area of research and associated challenges. [14] We consider the requirements for the Cybercraft *platform* (essentially a lightweight agent with a possible realized manifestation in hardware), which receives and executes generic *payloads* based on a common interface. We envision these payloads to be modules that execute persistently in agent process space, such as system and network sensors or communications modules.

There are countless possibilities for the further decomposition of the domain concepts seen in Figure 1 for cyber platform/payload interaction. We may in fact consider a broad range of possibilities for how a cyber defensive platform may be decomposed to achieve the goals of secure root of trust, scalable C3, and generic execution of payloads to support network defense goals. In our current research, we investigate the use of a classic design pattern from robotic control theory model known as Three-Layer Architecture. [15] This paradigm is designed to support multi-staged information flow for core decisionmaking activities and is built on a planning approach, where each of the three layers of the agent's planning process attempts to break a complex plan into one or more simpler plans.

We illustrate a possible component-based Three-Layer Architecture that meets the high-level goals for a defensive platform collective in Figure 4. The *coordinator*, *sequencer*, and *controller* structures provide the necessary interfaces to allow payloads to produce, share, and respond to state changes in the environment. The *coordinator* (often called the deliberator) deals with high-level goals, the *sequencer* splits a goal into actions, and the *controller* framework carries out the actions using a perceptual state and a primitive feedback loop. This process supports the implementation of more sophisticated and longer term goals, as well as machine learning, on which future defensive cyber collectives will likely rely.

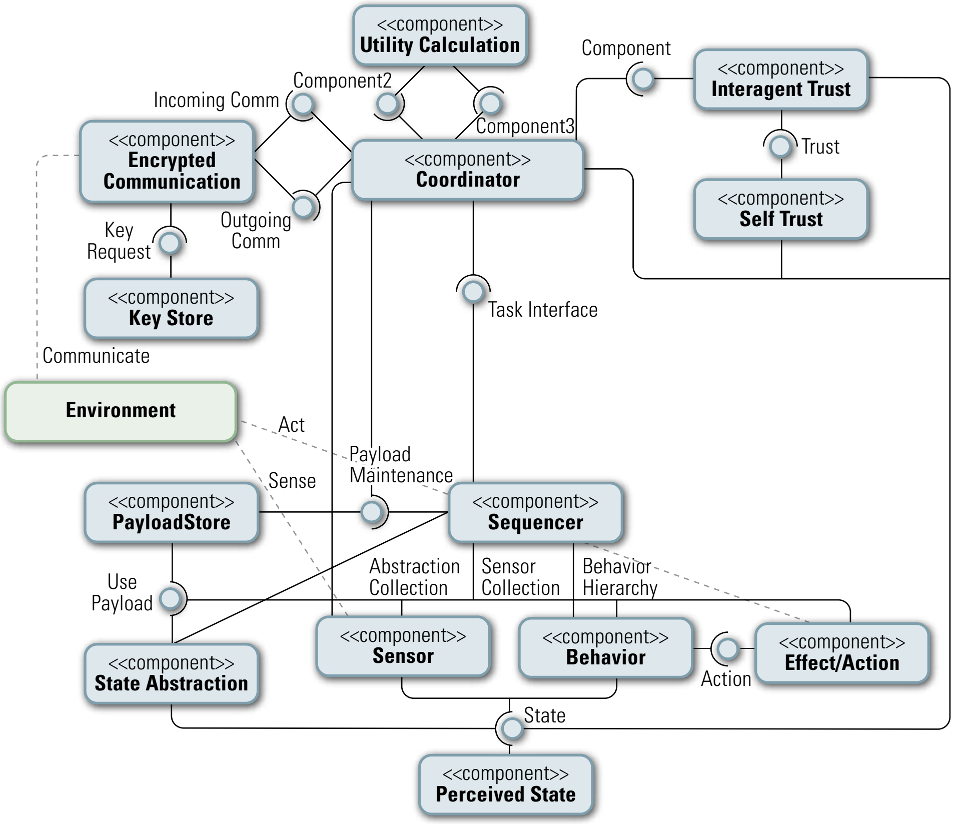


Figure 4 Conceptual Component Design for Defensive Cyber Architecture

This architecture illustrates a layered approach to payload integration and command functions. The first stage of information flow involves sensor modules that collect data about the agent's local or global environment. Other modules for the first stage include modules that provide secure and encrypted channels of communication. The second stage manages local perceptual state modules. Because the application demands a small agent, the state tracking is minimal, based on current mission and policy. Learning and decisionmaking occur in the third stage. For this application, modules in the third stage implement the Unified Behavior Framework (UBF). [16] This behavior framework supports simple and aggregate behaviors and is designed to be modified at runtime—illustrating one possibility for introducing flexibility and extensibility into the platform design.

In this design, the behavior (or controller) carries out sets of actions designated by the sequencer and provides quick response to unexpected states (much like a human's automated responses to tripping over a stone). The UBF maps behaviors together to form aggregate behaviors, and the mapping can be changed at runtime. Architectural design such as this may help better express a commander's intent or an operator's policy expression, which may change frequently and unexpectedly based on the needs of the mission and the agent's autonomy level.

The final stage in this model is an actuator stage. This stage usually consists of communicating alerts and status information back to human operators. However, if the system is under attack or detects threats to mission or resources,

Level	Synapsis
1	The computer offers no assistance; human must make all decisions and take actions
2	The computer offers a complete set of decision/action alternatives
3	The computer offers a selection of decision/action alternatives
4	The computer suggests one alternative and executes that suggestion if the human approves (management by consent)
5	The computer suggests one alternative and allows the human a restricted time to veto before automatic execution (management by exception)
6	The human is not involved in the decision making process; the computer decides and executes autonomously

Table 1 Levels of Computer System Automation

such as distributed denial of service, modules in this stage may restructure the network or enable additional security constraints. Secure communications is built into the core agent design, with a key store that will most likely be distributed. Models such as this express intent for agent architecture with flexible payload support and execution of defensive missions that may or may not exist when the fleet is deployed. The modules at each stage need only support the interface and communication requirements of the agent design. This provides a framework that may grow in capability as the system evolves and may be reconfigured at runtime.

We believe the strength of such robotic-based control design helps focus delineation of tasks in the platform and helps identify the possible interfaces between discrete components. This design also reflects the concept that our defensive platforms essentially function in the role of distributed and coordinated cyber sensor networks, with the added advantages of learning and large-scale communications. This particular design consideration offers us visibility into what the next generation of defensive agent collections will need to look like and helps us explore current technologies and design prototypes that may be integrated into a cohesive operational framework. Regardless of the particular agent design chosen, they will all need a high level of extensibility and flexibility to avoid monolithic platform/ payload realizations.

Autonomous Operations

Given flexibility in the agent architecture itself, questions remain about how, when, and where to permit human operators to observe and control the keys to the future Cybercastle defensive infrastructure. After all, attacks occur at the speed of electronic propagation, and detection/ response may need to execute at the same speed. Future systems will need some level of autonomy that our defensive cyber systems provide only in limited operational situations currently (*i.e.,* quarantining a known detected virus). Future defensive platforms must provide visibility and fusion of data *via* payloads so that the cognitive bandwidth of appropriate operators, administrators, and commanders remains low in the face of complex and coordinated network attacks. The effectiveness of the operator and the cyber platform network under consideration depends heavily on many factors, including the heterogeneity of the mission, the complexity of the interface to the cyber platforms, specific policies, the complexity of the payloads themselves, and the level of autonomy given to a cyber platform.

One common model for an operator control loop involves six levels of automation, as shown in Table 1. [17] Given a large network, a small number of centralized operators, the complexity of the USAF network defense mission, the number of varied activities in the future defensive landscape, and the speed of threats against this future integrated platform environment, it is unlikely that operators will be able to actively choose a course of action for each decision point (Level 1). Likewise, full software autonomy (Level 6) is unlikely due to the nature of the missions involved and trust evaluation of commanders at various levels. Research will continue to refine how we may effectively and efficiently inject operator monitoring and control into the fabric of the Cybercastle walls.

Redundancy/Fault Tolerance

Through normal incidents that create network outages in connectivity providers and lower reliability of some military systems (wireless, satellite), networks today and in the future will undoubtedly experience periodic connection problems. Whether our future defensive cyber platform hierarchies remain connected and operational in such environments is a question of great importance. Cyber defensive platforms in such network conditions must determine which missions are viable and how (if at all) the network must be reorganized. If a given set of missions cannot be completed, agent platforms must provide appropriate operator feedback or be programmed to execute autonomous decisionmaking evaluations. An operator may want to cease processing specific missions or provide for an autonomous halt so other missions remain unaffected. Cyber defensive platforms will be the workhorse to handle a wide variety of such network-related problems and will most likely act to provide correction, allow disconnected networks to rejoin, reevaluate traffic flows and patterns, and reorganize network configurations so missions can continue.

A plethora of research into distributed systems exists for determining when a network becomes disconnected and how to establish new leadership. The challenge for the Cybercastle of the future is determining if missions may continue based on more restricted resources. When loss of communication removes processing power, information, and assets that may be essential to completing the mission, we may allow our fleet of Cybercraft to know which resources are necessary to continue processing. How can we best represent these requirements? If the network is split evenly in two, each with the capability to continue the mission, should they both continue, should only one continue, or should both abandon processing? This provides yet another fertile ground of research, and the results will shape more precisely how defensive network suites function.

Looking Ahead

In the near term, there are several areas of achievable goals that we want to consider to develop and design the blueprints of the Cybercastle to meet the demands of the next few decades. Of interest will be the applicability of the HP2P design to military networks and our ability to formulate feasible options for hardware-based levels of trust in a host/system design context. Cyber platforms will need to be overlaid onto networks with varying underlying physical hierarchical topologies and possibly some independent communication networks for key super peer platforms of importance.

To integrate revolutionary concepts into the mainstream operational networks of interest (which our future C3 defensive systems will require to stay ahead of our adversaries), we must consider at some point how the transition from our current modes of operation might merge with newer technologies. We can start to gauge the design tradeoff space for the Cybercastle of the future by considering examples of current mission-critical systems, such as the Common Operational Picture (COP) and Air Operations Center (AOC). COP is a military system that distributes real-time information about a mission area (typically geography-centric) to personnel who use the information for mission planning. In a simplistic view, imagine a terrain map viewed on a desktop computer. The COP then feeds information about mission targets, friendly force locations (ground, air, and sea), points of interest, and other useful data overlaid onto the map. The information sent through the COP network typically consists of more than object position, and it allows operators to tie assets back to missions and vice versa. Such an integrated collection of missions may provide the perfect context for considering how revolutionary defensive changes may be integrated.

Regardless of what the Cybercastle physically looks like in 2020, we believe it must embody several of the principles we have discussed here: a secure root of trust that gives us the cyber high ground, a flexible C3 architecture that allows hierarchical and complex relationships among defensive nodes, and an extensible agent architectural design that supports tailored payload development and implementation with minimal changes to established platform interfaces. Of course, this collective must also provide some ease of use, support for autonomy, and resilience against network failure/attack. We believe some of the building blocks exist for this Cybercastle already: namely, our earnest expectation as researchers and cyber warriors to see them realized so the USAF may indeed fly, fight, and win in Cyberspace. ■

References

1. Gettle, M. "Air Force releases new mission statement." December 2005. *http://www.af.mil/news/story.asp?id=123013440.*
2. Lopez, C. T. "8th Air Force to become new cyber command." November 2006. *http://www.af.mil/news/story.asp?storyID=123030505.*
3. National Security Presidential Directive-54 / Homeland Security Presidential Directive-23 (Cybersecurity Policy).
4. Phister, D. P., Fayette, D., and Krzysiak, E. "Cybercraft: Concept linking NCW Principles with the Cyber Domain in an Urban Operational Environment." Technical Paper, Air Force Research Laboratory, 2006.
5. Kowalski, E. et al. "Insider Threat Study: Illicit Cyber Activity in the Government Sector." Technical Report, U.S. Secret Service and Software Engineering Institute, Carnegie Mellon University, January 2008. February 11, 2008. *http://secretservice.tpaq.treasury.gov/ntac/ final_government_sector2008_0109.pdf.*
6. McDonald, J. T., and Hunt, S. "Developing a Requirements Framework for Cybercraft Trust Evaluation." Proceedings of the 3rd International Conference on Information Warfare and Security, April 2008.
7. Trusted Computing Group, TCG Specification Architecture Overview. Revision 1.4 2 April 2007. *https://www.trustedcomputinggroup.org/groups/TCG_1_4_Architecture_Overview.pdf.*
8. Faloutsos, M., Faloutsos, P., and Faloutsos, C. "On power-law relationships of the internet topology." SIGCOMM, 1999, pp. 251–262.
9. Foster, I. T. and Iamnitchi, A. "On death, taxes, and the convergence of peer-to-peer and grid computing." IPTPS, 2003, pp. 118–128.
10. Yang, B. and Garcia-Molina, H. "Designing a super-peer network." ICDE, vol. 00, p. 49, 2003.
11. Ripeanu M., Iamnitchi, A., and Foster, I. "Mapping the gnutella network." IEEE Internet Computing, vol. 6, no. 1, 2002, pp. 50–57.
12. KYE, "Know your enemy: Fast-flux service networks." The Honeynet Project & Research Alliance, White Paper, July 2007.
13. "Kraken botarmy." [Online]. *http://www.damballa.com/research.*
14. Karrels, D. R. and Peterson, G. L. "CyberCraft: Protecting Air Force Electronic Systems with Lightweight Agents." Vir V. Phoha and S.S. Iyengar (editors), Proceedings of the Cyberspace Research Workshop, 58-62. United States Air Force, Shreveport, LA, November 2007.
15. Gat, E. Artificial Intelligence and Mobile Robots. Cambridge, MA, USA: MIT Press, 1998, ch. On Three Layer Architectures, pp. 195– 210.
16. Woolley, B. G. and Peterson, G. L. "Genetic evolution of hierarchical behavior structures." GECCO '07: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation. New York, NY, USA: ACM, 2007, pp. 1731–1738.

▷▷ *continued on page 19*

▷ continued from page 12

17. Sheridan, T. B. and Verplank, W. L. "Human and computer control of undersea teleoperators." MIT Man-Machine Systems Laboratory, Tech. Rep., 1978.

About the Authors

Lt Col J. Todd McDonald | is an Assistant Professor of Computer Science in the Department of Electrical and Computer Engineering at AFIT. Lt Col McDonald received a BS in computer science from the US Air Force Academy, an MS in computer engineering from AFIT, and a PhD in computer science from Florida State University. His research interests include software protection, obfuscation and anti-tamper applications, network and information security, and secure software engineering.

Dr. Gilbert "Bert" Peterson | is an Assistant Professor of Computer Engineering at the AFIT. Dr. Peterson holds a BS in architecture, an MS in computer science, and a PhD in computer science from the University of Texas at Arlington. He teaches and conducts research in digital forensics and artificial intelligence.

Capt Daniel R. Karrels | is a computer engineering PhD student at AFIT. Capt Karrels received his BS and MS in computer engineering from the University of Florida. His research interests include artificial intelligence, networking, large-scale systems, object-oriented design, and data structures.

Major Todd R. Andel | is an Assistant Professor of Computer Science in the Department of Electrical and Computer Engineering at AFIT. Major Andel received a BS in Computer Engineering from the University of Central Florida, an MS in Computer Engineering from AFIT, and a PhD in Computer Science from Florida State University. His research interests include formal methods, wireless routing, and network security protocols.

Dr. Richard "Rick" Raines | is the Director of the Center for Cyberspace Research (CCR) at AFIT. Dr. Raines received a BS in electrical engineering from Florida State University, an MS in computer engineering from AFIT, and a PhD in electrical engineering from Virginia Polytechnic Institute and State University. He teaches and conducts research in information security and global communications.

▷ continued from page 13

of the Virginia Alliance for Secure Computing and Networking Project (VASCAN), which was created to help secure Virginia universities and to work with the Virginia state government to aid in developing secure Virginia initiatives.

Mr. Berg has more than 35 years of experience in the intelligence community, private industry, and higher education. His experiences include developing security education programs for industry and the military on operations security and physical and personnel security. Mr. Berg has served as a consultant to universities and community colleges where he assisted in developing information security programs and federal grant proposals. His expertise includes distance education, security management and training, all-source intelligence collection and analysis, diplomatic activities, and special operations. Mr. Berg's work with universities and community colleges, the business community, researchers and faculty, and federal funding agencies has given him a deep appreciation of the importance of forging and supporting educational relationships between community colleges and universities and federal and state governments to meet the nation's commitment to securing our country and the protection, safety, and well-being of its citizens.

Mr. Berg is a member of the Government Security Conference (GOVSEC) board of advisors and served as the GOVSEC 2003 National Chairman. In addition, he is the Director of the Colloquium for Information Systems Security Education Secretariat, as well as the Treasurer and Conference Manager.

If you have technical questions for Dr. Maconachy, Mr. Berg, or another IATAC SME, please visit *http://iatac.dtic.mil/iatac*. The IATAC staff will assist you in reaching the SME best suited to helping you solve the challenge at hand. If you have any questions about the SME Program or are interested in joining the SME database and providing technical support to others in your domain of expertise, please email *iatac@dtic.mil* to have the URL for the SME application sent to you. ■

References

1. Web: *http://www.cisse.info/colloquia/cisse12/program/Vic%20Maconachy.htm*

About the Author

Angela Orebaugh | supports a variety of security engagements with the National Institute of Standards and Technology (NIST). She has 15 years experience in information technology and security and is the author of several technical security books including Nmap in the Enterprise and Wireshark & Ethereal Network Protocol Analyzer Toolkit.
Ms. Orebaugh is also an adjunct professor at George Mason University. She may be reached at *iatac@dtic.mil*