A Human-Agent Teamwork Command and Control Framework for Moving Target Defense (MTC2)

Marco M. Carvalho
Florida Institute of Technology.
Melbourne, FL
mcarvalho@fit.edu

Thomas C. Eskridge, Larry Bunch, Jeffrey M. Bradshaw, Adam Dalton, Paul Feltovich, and James Lott
Florida Institute for Human and Machine Cognition
{teskridge,lbunch,mbradshaw,adalton,pfeltovich,jlott}@ihmc.us

Daniel Kidwell
US Department of Defense
dlkidw2@tycho.ncsc.mil

ABSTRACT

In this paper we discuss the need for a command and control (C2) capability for moving target (MT) defenses. We describe some of the requirements and constraints associated with such a capability, and propose a human-agent teamwork approach for MTC2. We further discuss some specific concepts and technologies that could play an important role in the development of this capability, and conclude by describing some implementation details of a prototype being developed to demonstrate and study the proposed concepts for MTC2.

Categories and Subject Descriptors

K.6.5 [**Computing Milieux**]: Management of computing and information systems – *security and protection.*

General Terms

Management, Design, Reliability, Security

Keywords

Moving Target Defense, Command and Control, Human-Agent Teamwork, Coactive Emergence, Organic Resilience, Computer Network Defense

1. INTRODUCTION

The concept of Moving Target Defense (MTD) has been recently proposed as a game-changing capability for computer network defense with potentially broad applications in mission critical systems, enterprise networks, and national infrastructure security.

The basic idea in the concept of MTD is to create an environment that will enable the establishment and maintenance of a diverse and dynamic computational infrastructure.

The goal is to move to a new security paradigm where instead of protecting a fixed target, defenders will modify the target itself, increasing the cost for attacker to identify and effectively exploit specific vulnerabilities that exist on different states of the moving system.

MTDs can be both proactive and reactive. That is, they may be in place to proactively move the state of the protected system to vary the attack surface and thereby increase the costs of potential attacks, and may also respond to security events in order to quickly reconfigure in order to mitigate ongoing attacks, as well as subsequent attacks. These adaptive defense mechanisms (coupled with their proactive counterparts) allow systems to contextually change in a reactive mode, and potentially learn from experience.

In the last few years, a number of new ideas and techniques have been proposed in the context of moving target defenses [4][6][7]. These capabilities include defense monitoring and mobility components. Defense monitoring components include Intrusion Detection Systems, server and firewall log analysis, and traffic pattern monitors. Other capabilities are focused on creating the dynamic changes in the target system, which we refer to as the mobility space of the system. Some examples include, a) changes associated with the execution environment of services and applications, b) changes associated with the computation platform (i.e. operating systems and architecture), c) changes associated with the application or service itself, d) changes associated with the data used by services and applications, and e) changes associated with the network itself.

While encouraging results have been published for some of the proof-of-concept implementations of the proposed MTD concepts [6], there are still questions regarding their utility and practical use. There are important interdependencies between individual defense tools and the functionality of critical applications and services. Furthermore, different operational contexts are likely to require different configuration requirements for individual defense tools or groups of tools. This is especially important when taking into account the adaptation (or co-evolution) of the adversary. Therefore, from our perspective it is important to start addressing the coordination, or the command and control, aspects of moving target defense tools.

In this paper, our focus is not on specific MTD capabilities, but on the design requirements for a command and control framework that coordinates and controls one or more MTDs.

We believe that a MTD infrastructure must be able to combine, manage and optimize the use of multiple moving target defenses, under different operational, conditional and mission requirements. We also recognize that an effective coordination mechanism for these complex environments must account for both the high-level understanding and framing on operational settings, as well as the low level distributed monitoring and control enabled by intelligent software components.

CSIIRW '12, October 30 - November 2, Oak Ridge, Tennessee, USA
Copyright © 2012 ACM 978-1-4503-1687-3 ... $15.00

We propose, in this work, the need of a Human-Agent Teamwork approach to Moving Target Command and Control, and outline some initial ideas of a framework design for that purpose.

2. A MOVING TARGET DEFENSE INFRASTRUCTURE

Building from a literature review of moving target capabilities, there are three main components that constitute MTD Infrastructure: a) the monitoring tools and capabilities, b) the mobility tools, and c) the MTD command and control mechanisms.

Both for proactive and reactive mobility, a feedback-loop control analogy can be used to represent the interdependency among these components, as illustrated in Figure 1. The network defense and monitoring tools are responsible for observing the state of the different systems, services and networks. They include components such as network intrusion detection tools, performance monitors, host-base intrusion detection and others.

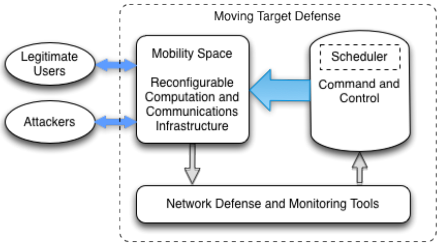
Moving Target Defense
Legitimate Users
Attackers
Mobility Space
Reconfigurable Computation and Communications Infrastructure
Scheduler
Command and Control
Network Defense and Monitoring Tools

Figure 1. Feedback-loop Control

The Command and Control component, in this case, uses feedback from the monitoring component and pre-defined proactive configurations (the scheduler) to maintain and control the mobility space (moving target defenses). This is a classical feedback control loop formulation of the problem.

One important difference from the classical feedback loop formulation lies in the sensing components. These components can, in theory, be configured and deployed at runtime, allowing the C2 to configure the sensors as well as the actuators. That is one of the reasons why more complex C2 capabilities are necessary. The control of the mobility space is influenced by the monitoring feedback, which can be configured to operate within specific contexts, or hypotheses.

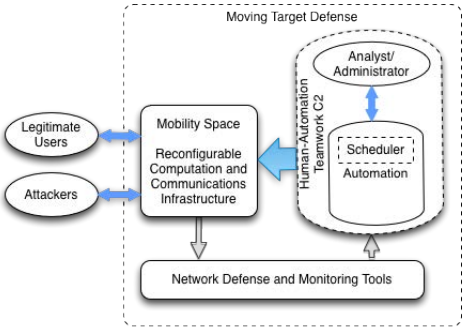
Moving Target Defense
Legitimate Users
Attackers
Mobility Space
Reconfigurable Computation and Communications Infrastructure
Human-Automation Teamwork C2
Analyst/ Administrator
Scheduler
Automation
Network Defense and Monitoring Tools

Figure 2. An Human-Agent Approach to MTC2

In our MTC2 design, we introduce human-agent teamwork to MT C2. As first introduced in [2], we propose two new important concepts and necessary capabilities for human-agent teamwork: Coactive Emergence [3] and Organic Resilience [5], both developed in the context of co-active design and emergent resilience, and are adapted to MTC2. In section 3 we briefly discuss our proposed design and some of the key capabilities for MTC2 and, in section 4, we introduce our prototype implementation where specific technologies are identified to support the new concepts and capabilities envisioned in our proposed design.

3. A HUMAN-AGENT TEAMWORK APPROACH FOR MOVING TARGET C2

To date, most approaches to MTD—and cyber operations in general—tend to focus on specific mechanisms, with the analyst being relegated to the role of compensating for various shortcomings in the opaquely-constructed automated control loop, rather than being part of the perception, decisions, and actions taking place within the loop itself. We believe that better, more resilient performance can be obtained by leveraging the joint capabilities of humans and automation – so long as the system is designed to support such teamwork. Humans can keep the technology aligned to richer situation contexts than what can be modeled within the system itself, verify ongoing progress and effectiveness, use their expertise to shape and reshape system actions, take corrective action as needed, and contribute the human powers of perception and decision-making to the work.

3.1 Coactive Emergence as an Approach to Human-Agent Teamwork

We characterize our approach to human-agent teamwork by the term *coactive emergence*. It describes a continuous iterative process whereby useful interpretations of data are developed, host and network configurations are adjusted, and effective responses to threats are undertaken through the interplay of joint sensemaking, decision-making, and task execution activities performed by analysts and software agents in tandem [3].

The word "coactive" emphasizes the joint, simultaneous, and interdependent nature of such collaboration among analysts and agents. Figure 3 illustrates how this applies to MTC2: 1) Analysts manage the work of software agents through policy constraints that direct their sensemaking and task execution activities; 2) Policy-governed agents work together to interpret real-time data and to manipulate host and network configurations, optionally enriching their capabilities through machine learning techniques; 3) Agents aggregate and present their findings to analysts as part of integrated graphical displays, and analysts interact with these displays in order monitor ongoing progress and effectiveness, and to explore and evaluate hypotheses and options; 4) Based on these results, analysts may redirect agent activities to increase system effectiveness or to take corrective action. Note that the sequential presentation of the cycle in the diagram is somewhat deceptive, because each of these activities occurs in parallel, and at individually varying rates.

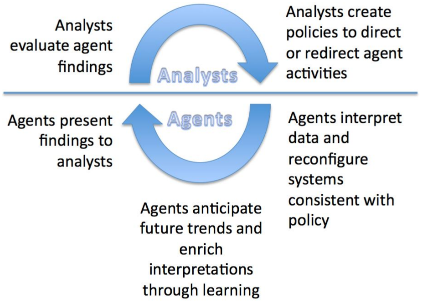
Analysts evaluate agent findings
Analysts
Analysts create policies to direct or redirect agent activities
Agents present findings to analysts
Agents
Agents interpret data and reconfigure systems consistent with policy
Agents anticipate future trends and enrich interpretations through learning

Figure 3. The Coactive Emergence Cycle

3.2 Organic Resilience

Our approach to resilience relies on software agents to assure graceful, robust, and adaptive performance in the face of stressors and surprise. Organic resilience relies heavily on biologically-inspired analogues and self-organizing strategies for the management and defense of distributed complex systems [5].

As with many biological systems, the goal of an organic resilience approach is to, as much as possible, avoid static and centralized single-point-of-failure solutions for organizing. Thus, although groups of agents within the system are collectively responsible for jointly executing various tasks, the specific responsibilities assigned to agents are not completely sorted out in advance. The goal is to allow the agents to self-organize within the constraints of their individual capabilities, the current applicable policies, and current availability of agents. Applied to organic resilience, policy-based collective obligations provide the regulatory mechanisms that enable effective and coactive coordination algorithms in the MTD mobility space.

4. THE MOVING TARGET C2

We are currently developing a prototype of the MTC2 framework to demonstrate and evaluate the proposed concepts. We have chosen to base the prototype in well-established services and capabilities that would support the principles of coactive emergence, organic resilience, and an easy integration with existing MTDs. In this section, we provide a brief description of some of the key components in our prototype.

4.1 KAoS Policy Services

Because agents can undertake action in a powerful and coordinated way, we use powerful policy management and enforcement frameworks to govern their actions. The KAoS Policy Services framework was the first to offer an ontology-based approach (based on the W3C standard, OWL 2) to policy representation and reasoning, and its core ontology has been adopted by the NSA-sponsored Digital Policy Management (DPM) Architecture Group as the basis for future standards efforts in DPM.

KAoS ensures that the software agents respect all security, privacy, sensemaking, and task execution policies, that they respond immediately to human redirection, and that they have the teamwork knowledge they need to collaboratively work with analysts and other agents. KAoS policies also ensure that the entire system adapts automatically to changes in context, environment, task reprioritization, or resources. New or modified policies can be made effective immediately and as widely as desired.

4.2 Luna Agent Framework

Software agents in the prototype are constructed using the Luna agent framework. Luna agents function both as interactive assistants to interpret data and carry out tasks [1]. They contain built-in capabilities, configurable through KAoS, which allow them to be proactive, collaborative, observable, and directable in human-agent teamwork interactions.

Luna also relies on KAoS for capabilities such as registration, discovery, self-description of actions and capabilities, communications transport, and messaging. One of the most important innovations in Luna is the ability to add custom agent actions to the policy ontology, based on their Java implementation. We provide a Java2OWL tool to automate this task.

4.3 Moving Target C2 APIs

The C2 API is designed to allow interaction with a wide variety of MTDs. It provides four main interfaces: the KnowledgeModel API, the Strategy API, the Registration API, and the Event API.

The KnowledgeModel API provides a general abstraction for the C2 system to gain awareness of concepts that are being protected by MTDs. The API distinguishes between types (or classes) and instances. It is designed to be compatible with ontology-based descriptions of these classes and instances (e.g. OWL), for integration with a semantic reasoner. However, the API itself does not place any requirements on the implementation; it just provides an abstraction for building a semantic description. This insulates the MTDs from being required to describe themselves in a particular semantic language. On top of the KnowledgeModel API, we have built convenience classes for registering the types of things that we know will be important, such as logical services and their execution environments (e.g., applications, operating systems, hardware, etc).

The Strategy API allows the C2 to observe and direct the strategies being used by the MTDs to protect the system. Strategy templates specify the parameters and ranges that can be used to define a strategy for a given defense. Strategy descriptions define a particular strategy by restricting the range of the parameters to a subset of the allowed types and/or instances. Strategy instances describe the possible instantiations that have been selected to satisfy a given strategy description. Lastly, the strategy state describes the current instantiation being used to satisfy the strategy.

The Registration API allows MTDs to register with the C2 service, and provides the C2 with a handle back to the defense. The C2 then uses the Strategy API to observe and direct changes to a defense's strategies. The Event API provides a generic abstraction for the C2 service to be notified about changes to the system state; such events may include Strategy events (strategy changed, strategy state changed, etc.) as well as Information events (e.g. intrusion alert received from IDS), and Progress events (e.g. moving service X to host Y is 90% complete). In addition to informing the C2 service, MT defenses may also register as listeners for events being published by other entities.

Combined, these APIs allow for the integration of standard defense mechanisms to the MTC2 framework. The APIs have been designed to require a specific defense to provide enough information for basic control and feedback of each defense, as well as the support of simple visualizations that facilitate human-agent interaction and teamwork. Due to the space limitation is this paper, we will not describe the visualization components of the MTC2, but the interested reader could find more details about the basic visualization principles utilized in [2].

5. ILLUSTRATIVE SCENARIO

While we continue to develop the complete prototype, early releases are available for test and demonstration in simple application scenarios. Our current implementation includes a first version of the MTD APIs, as well as the integrated C2 elements based on Luna and KAoS.

In order to illustrate the functionality of the system we have designed a simple scenario for the MTC2 using only two defense monitoring components (SNORT, and Server-Log monitors), and one defense components (application diversity). Functionally, the illustrative scenario developed for demonstration of the early prototype release is illustrated in Figure 4. The scenario is based on a simple client-server request/response model (a). In normal operation, a monitoring component (a host-based IDS in this case) detects events and notifies a user (or a syslog server). One moving target version of this simple scenario may implement, for example, an instance of application diversity, as illustrated in (b).

In this example, the functionality of the service is maintained (as perceived by the client), through the "static" logical service, but the actual implementation is moved across multiple (2, in this example) operating systems.

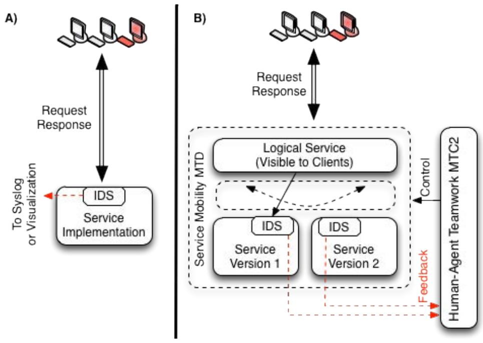

Figure 4. Functional description of the illustrative scenario

The implementation of the functional example shown in Figure 4, is illustrated in Figure 5. The target, in this example, is a web application that provides a front-end to a database server. Two configurations of the same service are provided, one base on MySQL (in one of the subnets), and one based on SQLite (on another subnet). One of the configurations is vulnerable to a SQL Injection attack that is periodically launched from some of the clients, while other clients continually make legitimate HTTP and database queries, creating background traffic.

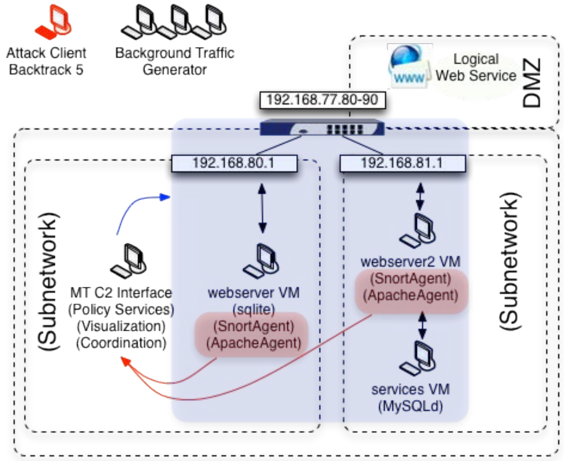

Figure 5. A simple scenario for prototype demonstration

Both the background traffic and the attacks are continuously submitted to the server (at random times, but following average rates). Attacks are designed as multi-stage sequences of queries, starting with a scan of the network to locate possible services, which is followed by a port-scan to identify potential HTTP servers. Once identified, HTTP server pages are then crawled in search of active pages.

For each active page, the attack then attempts a probe to try the SQL injection (on all active pages), and upon success, it initiates the exfiltration of a protected password table.

A tradeoff can be established between the level of defense and the level of service provided to legitimate users through the configuration of the defense. Humans are well positioned to evaluate that tradeoff, based on context and mission requirements. Agents are well suited to identify specific defense configurations that will satisfy user requirements, without violating system policies and other constraints. High-level trade off requirements are defined by users (through policies), and enforced by the software agents through a configuration to optimize the proposed trade-off. Humans and agents, in this context, collaborate to jointly control the specific moving target defense, at different levels of abstraction.

The MTC2 prototype has been implemented and is currently under tests in simulated scenarios like the one illustrated in this abstract. We are in the process of collecting results for the different scenarios to evaluate how the agents are reconfiguring the defenses with and without humans as team members. While this abstract refers to MTC2 managing a single defense, the current implementation of the prototype includes four defenses operating simultaneously as an example.

6. CONCLUSIONS

In this work we have discussed some of the requirements for a command and control framework for moving target defense. We have also proposed a human-agent teamwork approach for MT C2, and designed a prototype for demonstration and testing of the proposed concepts. We are currently on the final phases of the prototype and will soon provide an initial release that demonstrates the simple illustrative scenario discussed in this paper. Our focus in this work is on the discussion of specific capabilities for MTC2 support, as we progress with the prototype development new results will be made available to the research community.

7. REFERENCES

[1] Bunch, L., Carvalho, M., Bradshaw, J. M., Eskridge, T., Feltovich, P. J., Lott, J., and and Dan Kidwell, A. U. Policy-based governance within Luna: Why we developed yet another agent framework. In Software Agent Teamwork for the Semantic Web Workshop, in conjunction with 2012 IEEE/WICACM International Conference on Web Intelligence, Macau, China, December 2012.

[2] Carvalho, M., Bradshaw, J. M., Bunch, L., Eskridge, T., Feltovich, P., Hoffman, R., Lott, J., and Kidwell, D. A human-agent teamwork approach to moving target defense command and control. Poster presented at the Moving Target Research Workshop, Washington, D.C., June 2012. Online at: http://cps-vo.org/node/3850

[3] Bradshaw, M.J., Carvalho, M., Bunch, L., Eskridge, T., Feltovich, P., Forsythe, C., Hoffman, R., Johnson, M., Kidwell, D., Woods, D., Coactive Emergence as a Sensemaking Strategy for Cyber Defense, ICST Transactions (in print), 2012.

[4] Ghosh, A. K., Pendarakis, D., and Sanders, W. H. Moving target defense co-chair's report - National Cyber Leap Year Summit 2009. Tech. rep., Federal Networking and Information Technology Research and Development (NITRD) Program, 2009.

[5] Carvalho, M., Lamkin, T., and Perez, C. Organic resilience for tactical environments. In 5th International ICST Conference on Bio-Inspired Models of Network, Information, and Computing Systems (Bionetics), Boston, MA, December 2010.

[6] Jajodia, S., Ghosh, A. K., Swarup, V., Wang, C., and Wang, X. S. Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats, 1st ed. Springer Pub. Company, Incorporated, 2011

[7] Sheldon, F. T., and Vishik, C. Moving toward trustworthy systems: R&D Essentials. Computer 43, 9, pp. 31-40, September 2010