Available online at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Active cyber defense with denial and deception: A cyber-wargame experiment

Kristin E. Heckman*, Michael J. Walsh, Frank J. Stech, Todd A. O'Boyle, Stephen R. DiCato, Audra F. Herber

The MITRE Corporation, McLean, VA 22102, USA

ARTICLE INFO

Article history:
Received 27 November 2012
Received in revised form
6 March 2013
Accepted 31 March 2013

Keywords:
Computer security
Authentication
Denial
Deception
Counterdeception
Wargame

ABSTRACT

In January 2012, MITRE performed a real-time, red team/blue team cyber-wargame experiment. This presented the opportunity to blend cyber-warfare with traditional mission planning and execution, including denial and deception tradecraft. The cyber-wargame was designed to test a dynamic network defense cyber-security platform being researched in The MITRE Corporation's Innovation Program called Blackjack, and to investigate the utility of using denial and deception to enhance the defense of information in command and control systems.

The Blackjack tool failed to deny the adversary access to real information on the command and control mission system. The adversary had compromised a number of credentials without the computer network defenders' knowledge, and thereby observed both the real command and control mission system and the fake command and control mission system. However, traditional denial and deception techniques were effective in denying the adversary access to real information on the real command and control mission system, and instead provided the adversary with access to false information on a fake command and control mission system.

1. Introduction

Denial is the process of denying the adversary access to information, and deception is the process that creates misleading information through both facts and fictions (Bennett and Waltz, 2007). D&D are conventionally integrated because both processes are typically present and coordinated in a D&D plan to deceive an adversary. Deception is as old as history (Dulles, 1963) and it has evolved with the advent of technology. D&D is a key tool for secrecy, protection, evasion, and surprise (Bennett and Waltz, 2007; Whaley, 2007). It is a tool for both the defender and the attacker and can be used to target, among others, computer and communication systems, individual technology users, decision-makers, corporations, and national governments (Bennett and Waltz, 2007).

There are two general response approaches to unauthorized network access via compromised credentials: incident response and deceptive computer network defense (CND). In the first approach, security personnel lock and reissue credentials, preserve logs for post-mortem analysis, cleanup via scripts, restore services by reimaging affected systems, restore data from backups, and fix vulnerabilities (Brandt and

* Corresponding author. The MITRE Corporation, M/S N270 7515 Colshire Drive, McLean, VA 22102, United States. Tel.: +1 703 983 5547; fax: +1 703 983 3315.

E-mail addresses: kheckman@mitre.org (K.E. Heckman), mjwalsh@mitre.org (M.J. Walsh), stech@mitre.org (F.J. Stech), oboyle@mitre.org (T.A. O'Boyle), sdicato@mitre.org (S.R. DiCato), aherber@mitre.org (A.F. Herber).

0167-4048/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cose.2013.03.015

Wolff, 2010). In the second, security personnel select from a variety of computer security deception resources whose value lies in being probed, attacked, or compromised. Such resources include honeypots, "fake honeypots," "fake fake honeypots," honeynets, honeyclients, honeytokens, and tarpits. These resources can be included in a cyber-deception operation.

Honeypots are designed to be attacked to enable data collection about an attacker's activities and procedures, and to expose vulnerable services (Provos, 2004). Honeypots are typically categorized as high interaction or low interaction. Low interaction honeypots emulate services where the level of emulation built into the services determines the degree of intruder interaction with the honeypot (Spitzner, 2003). High interaction honeypots provide a real operating system designed to respond interactively to intruders (Spitzner, 2003). Several honeypot tools, their deceptive approaches, architectures, and methodologies are reviewed in Qassrawi and Hongli (2010). Fake honeypots are real machines with artifacts of false systems, such as virtualization and system monitoring tools, created to fool attackers into thinking they have compromised a valueless system, reducing the number of attacks on a real system (Rowe et al., 2007; Rowe et al., 2006). On the flip side, fake fake honeypots contain the same artifacts, but are in fact real honeypots which pretend to be overly obvious fake honeypots (Rowe et al., 2007; Rowe et al., 2006). A network of honeypot machines is a honeynet, which attempts to present a more plausible network environment to intruders (Spitzner, 2003).

Honeyclients were developed to simulate vulnerable client applications, to address a shift in tactics from server exploitation to client-side attacks using fraudulent emails and compromised web sites (Mansoori et al., 2012). These tools can also discover new browser vulnerabilities and proactively detect malicious webservers (Mansoori et al., 2012).

Honeytokens are another deceptive computer security approach. Honeytokens are crafted tidbits of information such as files containing false information, an email address associated with a non-existent person, or decoy traffic to detect eavesdropping in anonymous communication systems (Chakravarty et al., 2011). Although honeytokens are not intended to prevent attacks, they can provide effective early intrusion detection when accessed by an unauthorized user (Qassrawi and Hongli, 2010).

Tarpit software, such as Labrea Tarpit (Liston, n.d.), creates virtual servers that appear to reside on an organization's address space (Goh, 2007). These virtual servers send plausible but extremely inefficient replies to connection attempts. Tarpits make automated network scanning less effective, and they keep attackers busy to prevent them from attacking elsewhere.

A cyber-deception operation is the planned development and deployment of a set of actions taken to mislead attackers, and thereby cause them to take action, or inaction, which favors CND (Yuill, 2006). The purpose of a deception operation is to mislead the attacker into a predictable course of action or inaction that can be exploited. This is done by showing the attacker what is false and hiding what is real (Whaley, 1982; Bell and Whaley, 1982). Deception operations can utilize any or all of the deceptive CND responses previously described.

2. Background

In this paper we discuss an experiment that utilized a cyber-deception operation as a means of CND of information. The operation redirected an adversary using known compromised credentials to an adaptation of a traditional high-interaction honeypot. The honeypot was a fake command and control (C2) mission system wiki which was designed to look like a real C2 mission system wiki. A wiki is a web-based knowledge management system. The honeypot, or fake C2 mission system wiki, was dynamically populated with honeytokens, such as fake mission reports.

The real C2 mission system was operated and managed under the defender's practice of CND, whereby the defender focused on detecting the adversary's penetration by monitoring and analyzing its border boundary. The border boundary took the form of an Apache HTTP (HyperText Transfer Protocol) Server configured as a reverse proxy that would first receive and process every HTTP client request to the C2 mission system. The defender's monitoring consisted of analyzing the Apache server's access logs. Web browser clients would remain unaware of the Apache server's existence; thus, interaction with the proxy would lead either adversaries or approved users to believe they were directly accessing the real C2 mission system. The client would continue to make ordinary requests for content; the reverse proxy would then decide[1] where to send the user's request downstream to be processed; either directly to the real C2 mission system or to the Blackjack CND tool.[2] Requests were processed on the basis of the user's certificate and the proxy's configuration. That is, approved user requests continued on to the real C2 mission system, whereas adversary requests were redirected to Blackjack as shown in Fig. 1. After processing the request, the server would return the response of the downstream system as if it were the originator.

Blackjack used a rules engine to apply policy to each request in order to direct its response to user requests called Intellect,[3] a domain-specific language and rule engine written in Python. Intellect expresses policies to control Blackjack based on a rules engine that provides a form of artificial intelligence for objectively reasoning and understanding while utilizing a working memory. This artificial working memory retains knowledge relevant to the system, and a set of rules authored in the domain-specific language that describe a necessary behavior to achieve a goal. Each rule has an operational condition, and a suite of one or more actions. These actions either further direct the behavior of the system, and/or further inform the system. The engine starts with some facts,

[1] The reverse proxy was in place for all communication as a load balancer. Such a load balancing reverse proxy could make decisions about which backend content provider should be used based on several factors, such as geographic location, server load, network load, etc. In this experiment, the load balancing reverse proxy simply used userID as the decision criterion.

[2] The Blackjack CND tool is in development and actively being researched via a series of real-time red team/blue team cyberwargame experiments.

[3] Intellect is open sourced by The MITRE Corporation: https://github.com/nemonik/Intellect.