Available online at www.sciencedirect.com

ScienceDirect

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

Computers & Security

NIC displays to thwart malware attacks mounted from within the OS

Julian L. Rrushi *

Department of Computer Science, Western Washington University, Bellingham, Washington 98225-9038, USA

ARTICLE INFO

Article history:
Received 12 August 2015
Received in revised form 6 April 2016
Accepted 12 May 2016
Available online 26 May 2016

Keywords:
Malware
Insider threats
I/O interception
Defensive cyber deception
Computer decoys

ABSTRACT

This paper describes an OS-resident defensive deception approach, which can neutralize malware that has managed to infect a target machine. Such attacks account for most of the spying operations detected to date, and include malware, insider code, and Trojans that originate from compromises of the computer supply chain. The central idea that underpins this work is to display the existence of I/O devices in a computer system. While those I/O devices would not exist for real, their projection will make them appear as valid targets of interception and malicious modification, or as valid means of propagation to other target computers. We experiment with the implementation of a low-level network driver for the Windows operating system. The network driver emulates the operation of a network interface controller (NIC), and thus reports to higher-level drivers in the network stack as if the NIC were existent, fully functional, and with access to an existing computer network. We tested and evaluated NIC displays against a large sample of live malware, and thus discuss our findings in the paper.

1. Introduction

This paper describes the design, implementation, and evaluation of an applied defensive deception capability that can detect and neutralize attacks originating from within the operating system of a computer. Attacks of that kind are mounted by a malicious insider, or by an external attacker that has gained access to a computer through vulnerability exploitation. In both of those cases, the malware operates directly on the compromised machine. Interception code is the principal mechanism of gaining access to sensitive data that are stored on the compromised machine, or are processed or communicated by it. Technically savvy attackers can always develop their own interception tools. Otherwise powerful interception tools are available in the hackers' black market.

Cyber spying tools are often referred to as remote access tools or remote control systems. An example of practical significance is ComDarket (Kujawa). The malicious insider with administrator privileges on a target computer system represents an attack vector that could introduce the interception software into that computer. The malicious insider could install the interception software to intercept the webcam video traffic of any user of the computer system who uses the webcam. The attack vector is realistic, and has occurrences in the real world. There are public cases of installation of interception software by malicious insiders. The material in Prevelakis and Spinellis (2007) and Salem et al. (2008) describes the work of a malicious insider of the cell phone provider Vodafone in Greece. The malicious insider installed a rootkit on a key system of Vodafone, which intercepted the phone calls of select phone devices, including those of the prime minister and other high rank officers of the government.

Another attack vector stems from compromises of the supply chain that affect software and firmware installed on a computer (DARPA, 2013). That attack vector exploits the trust that

* *E-mail address:* julian.rrushi@wwu.edu.

http://dx.doi.org/10.1016/j.cose.2016.05.002
0167-4048/© 2016 Elsevier Ltd. All rights reserved.

a computer user places on software systems that are provided or sold as a desired utility. Some of those software systems are developed by malicious parties, who target the computer where such Trojan software is installed. The installation process of the Trojan software in question often requires administrator privileges, which the computer user provides voluntarily due to the trust granted to that software. Armed with a high level of privileges along with the opportunity to run on the target computer system, the Trojan software can easily penetrate the kernel of the operating system, and thus deploy its interception capabilities. We deem such a deception approach to be an insider exploit, given that the software provider and the software itself are granted insider status by being entrusted with accessing the computer system with a high level of privilege to provide a desired utility.

A hybrid form of various attack vectors could also be the root cause of insider attacks. For example, a commercial off-the-shelf software system may be originally free of Trojan functionality. A malicious insider in an organization, however, could inject Trojan code into that software prior to its installation on the organization's computers. There are circumstances in which the adversary has an opportunity of physical access to mobile computing devices. Transiting through the border of an unfriendly country, for example, would expose a laptop to fast interception software installation during the regular border crossing inspections. The contributions made by this article are the following:

- A technical description of the design and development of a phantom network interface controller (NIC) to intercept and locate malware and insider threats. The approach is described as applied to the Windows operating system.
- A practical evaluation of the defensive deception capability against a large set of malware samples, along with a practical assessment of its overhead.

The remainder of this paper is organized as follows. In Section II we discuss the threat model, architectural design, and practical implementation of the proposed defensive deception approach. In Section III we describe an empirical evaluation of the effectiveness and overhead of the approach. In Section IV we discuss the state of the art in cyber deception in terms of existing techniques and tools that function through deception concepts and mechanisms. In Section V we summarize our findings and conclude the paper.

2. Phantom NIC

2.1. Threat model

The threat model in this work establishes that there is a defender active in the system, who is genuinely interested in securing the computer. Without the existence of a defender, the machine is entirely lost. In those conditions, it makes no sense to rely on any computer security mechanisms, including phantom NICs, since there would be no one to install and operate those security mechanisms, and thus take action when alerts are raised or malware are identified. In enterprise networks of any size, the best role for the defender is that of system administrator. In a private computer, the defender is the owner. Administrator privileges are required to install and run a phantom NIC in the kernel of the operating system, and also respond to access by removing malware and investigating the intrusion. As of this writing, the phantom NIC security mechanism raises alerts for the defender to process, and hence does not remove malware by itself automatically.

The attack code is assumed to have access to the kernel of the operation system. The attack code can also have modules that run in user space. This is the most common form of system compromise caused by malware that lands on the system through one or more exploits, or through the other means described earlier in this paper. A human insider threat to the system is a user with a limited or administrative level of privileges on the machine. It is a strong requirement in this threat model that the human insider threat be other than the defender. This requirement is no different than those requested by other existing security mechanisms on a machine. For example, the user who writes an access control matrix to regulate access to files on a system needs to be different than any malicious users whose file access the matrix would enforce. If that requirement is not met, the access control matrix is useless.

Another strong requirement in this threat model is that the details of the phantom NIC, such as which NIC is real and which NIC is phantom, or what specific driver and data structures belong to the phantom NIC, are known only to the defender. Clearly if the malware or insider threat had that knowledge, the phantom NIC would be useless. The malware or insider threat is pulled into a catch-22 dilemma. To avoid generating any activity on the phantom NIC, such as sending network packets or probing its driver, the attacker or attack code needs to know which NIC is a decoy. In order to find out which NIC is a decoy, the attacker or attack code needs to generate activity on the phantom NIC. In regard to generating activity on a NIC, we are referring to capabilities of current malware based on our analysis of a large set of malware samples, and those of system administration tools.

A capability that is not present in current malware, and which we have experimented with, is that of probing the hardware of a target NIC directly through its hardware bus. To counter the next generation of malware that may have that capability, we are continuing with the work described in this paper to provide hardware support for a phantom NIC. We discuss some of the details of that research later on in the final section of this paper. The phantom NIC is positioned in a randomly selected order relative to the existing NICs on the machine. The order is changed at random after the machine reboots. An attempt to disable a phantom NIC results in detectable activity, and thus is subject to the catch-22 dilemma discussed earlier in this section. All external memory accesses to the driver stack of the phantom NIC are reported to the defender as an indication of attack.

2.2. Display projection

A prevalent structural characteristic of malware installed on a compromised computer system is the use of its network interfaces, i.e., connection to wired or wireless networks, to exploit other computers. The central idea that underpins the