UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CROWDSTRIKE, INC.,
Petitioner

v.

GOSECURE, INC.,
Patent Owner

Inter Partes Review Case No. IPR2025-00070
U.S. Patent No. 9,954,872

SUPPLEMENTAL DECLARATION OF DR. MARKUS JAKOBSSON

TABLE OF CONTENTS

I. INTRODUCTION 1

II. SUPPLEMENTAL OPINIONS 2

A. THE '872 PATENT AND CAPALIK MONITOR ATTACK ACTIVITY IN ONE OR MORE VIRTUAL MACHINES WITHIN A DECOY DEVICE 2

III. CONCLUSIONS 6

I. INTRODUCTION

1. My name is Markus Jakobsson, and I am over 21 years of age and otherwise competent to make this Declaration. I make this Declaration based on facts and matters within my own knowledge and on information provided to me by others.

2. I submitted a declaration (Ex. 1003) in support of Petitioner's Petition for *Inter Partes* Review of U.S. Patent Nos. 9,954,872 (the "Challenged Patent"). I understand the PTAB instituted the requested review. I have been asked to address a few additional issues raised by Patent Owner ("PO") in Patent Owner's Response ("POR") dated December 12, 2025, and the expert declaration of Dr. Angelos Stravrou (Ex. 2031). All opinions expressed in my original declaration (Ex. 1003) remain the same.

3. As part of my work and in forming my opinions in connection with this proceeding, in addition to the materials cited in my original declaration, I have reviewed portions of the following materials (i.e., the portions cited here and surrounding discussions for context):

- Patent Owner Response (Paper 47)
- Declaration of Dr. Angelos Stavrou (Ex. 2031)

II. SUPPLEMENTAL OPINIONS

A. The ’872 Patent and Capalik Monitor Attack Activity In One or More Virtual Machines Within a Decoy Device

4. I understand PO and Dr. Stavrou have argued that a key distinction between the ’872 Patent and Capalik is that the former monitors activities all the time, unrelated to identifying an incoming attack, and the latter monitors only incident to identifying an incoming attack. Patent Owner Response (“POR”), 43-52; Stavrou Dec. (Ex. 2031), ¶¶ 24-29, 67-70. I disagree. Both the ’872 Patent and Capalik describe monitoring activities in one or more virtual machines (“VM”) within a decoy device. Those VMs are intentionally left susceptible to attacks, so the attacks can be monitored. In both the ‘872 Patent and Capalik, the decoy device exists to be attacked, and until it has been attacked, there is no attack activity to monitor.

5. The ’872 Patent consistently describes monitoring and identifying activities performed in virtual machine(s) (“VMs”) within a decoy network device. *See, e.g*., Ex. 1001, 2:36-38 (“The method includes monitoring activity on the virtual machine. A plurality of activities performed at a virtual machine is identified.”); *id*., 2:55-58 (“The one or more programs include instructions for monitoring activity on the virtual machine, and instructions for identifying a plurality of activities being performed at the virtual machine.”); *id*., 6:46-52 (explaining that the introspection kernel “performs introspection (e.g., monitoring) into the physical memory

segments assigned to each of the [VM operating systems]."); Figs. 3A-3D, 4A-4D (depicting unauthorized activity within a VM); Figs. 6A-6C (illustrating monitoring and identifying unauthorized activities on a VM). These disclosures reflect a consistent design in which the VM serves as the monitored execution environment where attack activity is observed and analyzed.

6. Capalik describes the same approach. It monitors and identifies activities occurring in virtual machine(s) within the decoy computer network. *See e.g.,* Capalik, [0047] ("[I]ntrospection module 538 (also known as a virtual machine-based memory introspection analysis tool) that monitors and introspects into the virtualized decoy operating systems memory segments 512."), [0049] ("When an attacker connects to a network port and begins interacting with a virtualized decoy operating system 512, the introspection module 538 monitors and captures information from the connection, including port numbers, data streams, file uploads, keystrokes, ASCII or binary files, malicious payloads, memory manipulation attempts, and any other data transfers or malicious attempts."). In both Capalik and the '872 Patent, the monitoring is focused on activity being performed in the VM, where attacker interaction occurs. Neither reference describes monitoring activities in other components of the decoy system outside the VM.

7. Although the '872 Patent notes that similar methods of the alleged invention could be used in "non-decoy systems" or "non-virtualized computer

systems," it provides no examples or description of applying the alleged invention in such systems. Ex. 1001, 5:25-30 ("Although the methods described herein are described with respect to monitoring the decoy network device 106, persons having ordinary skill in the art would recognize that similar methods may be used for monitoring one or more non-decoy systems, such as the protected network device 136."), 11:18-22 ("Although FIGS. 3A-3D and 4A-4E illustrate unauthorized activities on the virtual machine 113-1, analogous methods can be used to represent and analyze unauthorized activities on non-virtualized computer systems."). While it is conceivable that certain monitoring functionalities described in the '872 Patent could be applied to a non-decoy computer system, the patent does not describe any such implementation.

8. Both Capalik and the '872 Patent describe decoy systems running VMs that are intentionally left susceptible to attacks, so attacks to the VMs can be monitored. Ex. 1001, 5:3-6 ("The purpose of the decoy network device 106 is to allow attackers to attack the decoy network device 106, so that the pattern of the attack can be monitored and analyzed to generate a fingerprint."); Capalik, [0045] ("The instigator of the attack 550 is thus lured into accessing the decoy 512, which is overseen by the hypervisor operating system[.]"). This is a well-understood design in the field in which instrumentation is triggered by attacker interaction with the decoy environment. Ex. 1003, Section V (discussing honeypots/decoy systems). In

both systems, monitoring occurs in response to attack activity within the VM. Ex. 1001, 9:57-10:7; Capalik, [0049]. Until such interaction occurs, there is no attack behavior to monitor, identify, record, or analyze. Accordingly, the decoy system is designed to monitor and identify activity once an attacker engages with the VM. It is not, as PO and Dr. Stavrou suggest, designed to monitor or identify non-attack activity in isolation from a detected attacker access.

9. Figs. 3A-3D and 4A-4E illustrate exemplary patterns of attack identified on a VM. Ex. 1001, 11:15-22. The network diagram of the pattern of attack depicted in Figs. 3A-3D starts with the identification of an unauthorized activity—Socket 306 receiving data Packet Stream 304. *Id*., 11:26-36. Once the first detected activity is deemed to be "affected by unauthorized activities," related unauthorized activities are identified, providing a detailed picture of the compromise. *Id.*, 11:45-47, Figs. 3B-3D. The attack pattern identified in Figs. 4A-4D is virtually identical to the one depicted in Figs. 3A-3D. However, the described process in Figs. 4A-4D starts with the identifying specific unauthorized activities, i.e., in File X 310 and File Y 312. Once the system detects these unauthorized activities, related unauthorized activities are identified, providing a picture of how an attacker broke into the system. *Id*., 11:66-12:26, Figs. 4A-4D.

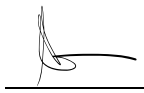
10. Identifying an attack pattern is a post hoc analysis that depends on previously monitored activity data. In both examples depicted in Figs. 3A-3D and

4A-4E, the system correlates events recorded during execution within the VM to reconstruct relationships between processes, files, and network activity. Ex. 1001, 11:15-22. This type of dependency tracing is well understood in intrusion analysis and requires that the underlying events have already been monitored and logged during runtime. *See e.g.*, Ex. 1006 (King), 24-29 (describing building backward and forward dependency graphs "[b]y logging objects and dependency-causing events during runtime"). Accordingly, the '872 Patent system's ability to trace activity from File X back to attacker 302, as in Figs. 4A-4D, presupposes that all relevant activity was monitored and captured after the attacker accessed the VM through socket 306. The system cannot reconstruct or analyze activity that was not first monitored. Thus, like Capalik, the '872 Patent monitors activity in response to attacker interaction with the VM, and subsequent identification of attack patterns is based on that monitored data.

III. CONCLUSIONS

11. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Date: March 19, 2026

By: 

Dr. Markus Jakobsson