Global Information Assurance Certification Paper

Gaining Visibility on the Network with Security Onion: A Cyber Threat Intelligence Based Approach

GIAC (GSEC) Gold Certification

Author: Alfredo Hickman, ahusmc@yahoo.com
Advisor: Rich Graves
Accepted: February 1st 2016

Abstract

Generating threat intelligence, detecting network intrusions, and preventing cyber threat actors from executing their objectives are critical measures for preserving cybersecurity. Network breaches of organizations such as the U.S. Office of Personnel Management, Target, Anthem, and many others, are proving that individuals and organizations of all sizes and backgrounds are targets of cyber threat actors. Another reality is that not everybody is equipped and funded to leverage threat intelligence to detect network intrusions and respond accordingly.

In response to this reality, this paper will address how small and medium sized organizations can leverage Security Onion, by Doug Burks, a Linux based Network Security Monitor, Intrusion Detection System, and Log Monitor, to generate and consume local threat intelligence to detect and mitigate the harms of network breaches. By performing a case study of deploying and operating Security Onion to detect and respond to a network breach, it is the intent of this paper to provide a roadmap for others to better secure their networks and facilitate the creation and consumption of threat intelligence.

1. Introduction

The goals behind Cyber Threat Intelligence are to provide strategic, operational, and tactical information that is both actionable and relevant to decision makers. While not new, Cyber Threat Intelligence (CTI) is now emerging as a preeminent discipline for detecting threat actors and mitigating cyber threats and attacks. By utilizing existing Tools, Techniques, and Procedures (TTPs), and leveraging advances in intelligence tradecraft and technologies such as cloud computing, machine learning, artificial intelligence, data science, analytics, and more, Cyber Threat Intelligence promises to be more than just a passing fad in the information and cybersecurity ecosystem. Furthermore, as Gilbert (2014) claims in his Stanford University honors thesis, the benefit of CTI is greatly enhanced by the sharing of CTI amongst government, industry, and academia.

While there is little doubt that Cyber Threat Intelligence and information sharing provide powerful and useful tools for detecting, preventing, and mitigating cyber threats and attacks, many individuals and organizations do not possess the resources required to develop or acquire CTI products, services, or talent. As such, it is the goal of this research to explore the creation, implementation, and operation of a component of a CTI and information-sharing platform, the network security monitor. As Nair and Puri (2015) attest in their research on Cyber Threat Intelligence, the creation of an analytic engine to facilitate the collection, processing, analysis, and dissemination of threat intelligence amongst organizations that have traditionally been unable to afford such systems is critical to a secure Internet. As such, this research will incorporate the work that Doug Burks has done in creating and making freely accessible, the Security Onion platform. Security Onion is a Linux based Network Security Monitor, Intrusion Detection System, and Log Monitor that can facilitate the collection and creation of the basic data and information components that are required to generate CTI. Furthermore, this research will build upon information published in the International Journal of Research and within the Intelligence and National Security Alliance to build a free and open source method for creating and consuming CTI.

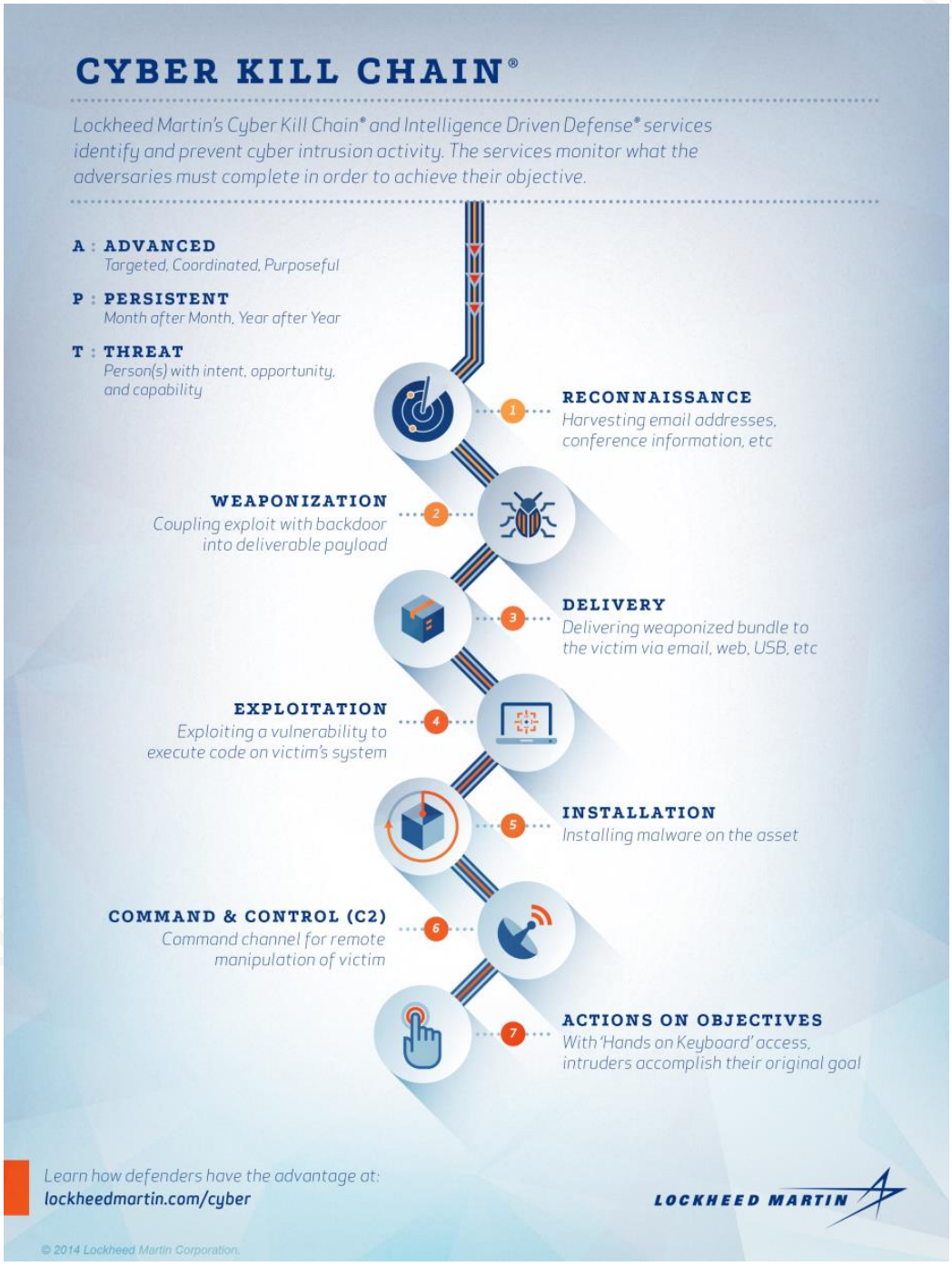
Alfredo Hickman, ahusmc@yahoo.com

2. Project Scope

The scope of the project is limited to the installation, configuration, and operation of Security Onion; and the analysis of data and information generated from the Security Onion sensor and server. The lab environment used in this project consists of one virtual Security Onion server and one Security Onion sensor installed according to the production deployment specifications found at the Security Onion Website. The traffic generated for the sensing interface of the Security Onion sensor will be fed to the sensor from a mirrored port on a managed switch. The switch will then forward to the sensor all traffic contained within a network consisting of various computers, printers, storage devices, and various other end-user devices. Furthermore, a laptop has been installed with Kali Linux 2.0 for the purpose of reconnoitering, enumerating, and attacking computers on the simulated network.

It is the intention of the project to have Security Onion detect and analyze malicious behavior on the network in order to generate data and information products that detail aspects of the Cyber Kill-Chain. This data can then be consumed and distributed within and amongst organization to detect and mitigate the effects of cyber-attacks. The Cyber Kill-Chain is a model developed by Lockheed Martin (2016) that describes the various stages of a cyber-attack, which include reconnaissance, malware weaponization, malware delivery, target exploitation, malware installation, system command and control, and actions on the objective. The Cyber Kill-Chain is depicted in Figure 1. In addition, I have added the process of cleanup and obfuscation to the kill-chain, as these are common actions in cyber-attacks. However, cleanup and obfuscation can be difficult to detect using Security Onion alone and would typically require a more robust, host-based approach to detection.

Figure 1. Lockheed Martin Cyber Kill Chain



Alfredo Hickman, ahusmc@yahoo.com

3. Lab Setup

The test lab setup consists of one Security Onion server and one sensor. The server and sensor are setup in a distributed production deployment configuration, as detailed by Burks (2015) on the Security Onion Website. The advanced Security Onion configuration is ideal in a production network, as the system is then able to collect and analyze data from a more complete network traffic perspective, such as from DMZs, Wi-Fi networks, and other network enclaves.

In addition, the test lab consists of one modem connected to the public Internet, a firewall and intrusion detection and prevention appliance with gateway anti-malware, a layer two managed switch, a network attached storage appliance, and one Windows PC. All the devices connect directly to the layer two switch, and the switch mirrors all incoming and outgoing traffic to a local SPAN port, which is directly connected to the Security Onion sensor interface. The Kali Linux laptop will attempt to conduct reconnaissance, enumeration, exploitation, and exfiltration from the inside local area network. As it is when an intruder is in the network that the detection of malicious behavior is no longer only desirable, but necessary for threat and attack mitigation.

4. Security Onion Setup and Configuration

The version of Security Onion used is Security Onion 12.04.5.3, based on the Xubuntu 12.04 LTS x64 Desktop distribution. Security Onion can be installed on any Ubuntu 12.04 LTS based distribution, and is done by installing the Security Onion Personal Package Archive (PPA) and going through the setup process. The two methods of installing and configuring Security Onion are the quick setup and the advanced setup. If a simple evaluation of Security Onion is required, then the quick setup will suffice. However, this research uses the advanced setup, since the intention is to provide a roadmap for implementing Security Onion in a production environment.

4.1 Security Onion Advanced Setup

Per the Security Onion installation instructions, download the Security Onion ISO and boot it on either a dedicated server or virtual machine. Once booted to the Security Onion Desktop GUI, run the Security Onion 12.04 installation program (sudo soup from the terminal). Once the installation screen appears, select the basic operating system configurations, and then proceed through the prompts. Now, select the installation type and select whether or not to erase and use the entire provisioned disk, or to partition the disk accordingly. Note: in production deployments, the contents of the intrusion detection system (IDS) database can grow to be quite large, and may require its own physical or logical disks or partitions. Furthermore, if utilizing a disk larger than two terabytes, it may be desirable to create a dedicated boot partition to prevent any boot loader issues. (Burks 2015).

Now, continue installing Security Onion and advance through the prompts after selecting the options appropriate to your location, preferences, and identity. Once the installation completes and the system reboots, login into the system and open the terminal. From the terminal, run the command “sudo soup,” which will update various components of the Security Onion installation. Once the “soup” operation completes and the system is rebooted, login to the system and execute the “Setup” program located on the desktop.

The Security Onion setup program will prompt to configure various components of the Security Onion server and sensor options depending on the system requirements. For the purpose of this research, the Security Onion server was configured in the following manner: The Security Onion Distribution was installed on a VirtualBox virtual machine, and it was provisioned with 2 CPU cores, 8 GBs of RAM, and 100 GBs of disk storage. During the Security Onion server installation, Suricata was selected as the IDS engine, Emerging Threats GPL was selected as the IDS ruleset, Bro was configured, and the access credentials for Sguil, Squert, ELSA, and Snorby were defined. Once the Security Onion server was setup and running, the sensor was configured next.

For this portion of the Security Onion deployment, the Security Onion Distribution was installed on the dedicated sensor appliance. Once the appliance booted to the live disk, the basic installation of the Security Onion was performed according to the instructions on the Security Onion website. During the sensor setup, the network interfaces were automatically optimized to perform NIC offloading, and the sensing and management interfaces were selected. Static IP addresses were configured for the sensor on the same subnet as the Security Onion server. This is done in order for network traffic to flow from the sensor to the server, and to allow for sensor administration from the server.

At this point, the Security Onion sensor reboots, and the Security Onion setup continues in advanced mode. The next steps are to select the sensor role and to configure SSH access back to the Security Onion Sguil server. Once the sensor connects back to the Security Onion Sguil server, the network interfaces that will monitor network traffic must be selected, and CPU cores must be assigned to the IDS processes. Once these steps are complete, Bro is configured. Then, the system will prompt for a decision on whether or not to enable ELSA on the sensor in order to manage distributed sensor logs from the central Security Onion server. Once both the server and sensors have been installed, configured, and updated, the system is ready to begin monitoring the network and identifying any potentially malicious behavior. The majority of the analysis done with Security Onion is done from the web interface and the Sguil application located on the Security Onion server.

Alfredo Hickman, ahusmc@yahoo.com

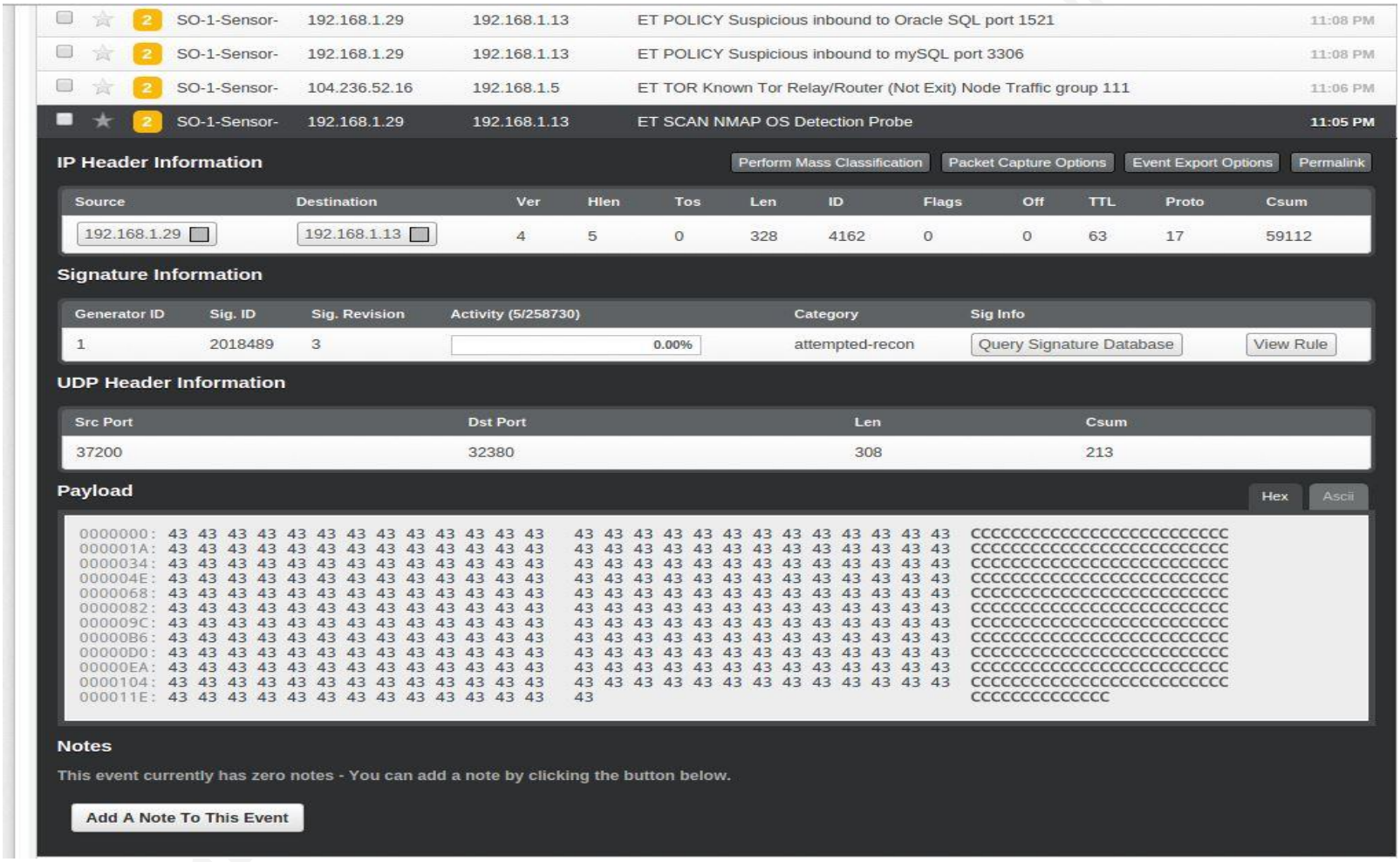
5. Data Breach Scenario and Detection of Malicious Activities

The test lab simulates a small business production network. An intruder has harvested valid IPsec VPN credentials for a standard; non-elevated user account, and has been assigned a DHCP IP address that is on the business's local area network. In addition, the business is using a flat IP network design that places all networked information systems on the same logical subnet: a standard 192.168.1.0/24 IP network. The business does not have many technical information security controls beyond an edge firewall and IDS, host-based anti-malware software, and a Security Onion network security monitor. The Security Onion NSM has its sensor interface placed on a SPAN port on their core Ethernet switch, and the server is installed on a virtual machine that is maintained by the business's system administrator.

What makes this scenario ideal is that there are many SMB networks designed in this manner. It is all too common to find SMBs that are not particularly versed in information security, and that do not have robust detective or preventative information security controls. Furthermore, it is common for many SMBs to operate their networks on flat topologies, and that have VPNs that employ single factor authentication (such as username and passwords). In addition, it is all too common for SMB VPNs to terminate on the LAN side of their networks, thus placing VPN users in the heart of the business network. As such, it is becoming commonplace for hackers to attempt to gain access to legitimate VPN credentials, and thus gain access to networks that way, rather than to exploit perimeter defenses or the VPN itself, which often more challenging. In this case, the stakes for detecting malicious behavior on the network grow even higher, as the intruder's initial breach of the network is through an otherwise authorized method that would not initially set off any "bells and whistles." It is in these scenarios where detection of malicious behavior is necessary.

At this point of the breach, the intruder's objectives are to find potentially valuable information like financial records, intellectual property, business plans, or incriminating or embarrassing information, and then to successfully exfiltrate and make use of the information. The first step the intruder takes after gaining access to the business's network is to scan the network to find vulnerable information systems that may contain valuable information, such as a domain controller, an administrative workstation, or file server. Now, the intruder performs a ping sweep of the network to identify available hosts for potential exploitation. At this point, Security Onion alerts to the fact that a host has scanned the network and attempted to determine the operating systems of various machines on the network as shown on Figure 2.

Figure 2. Security Onion NMAP OS Detection Probe

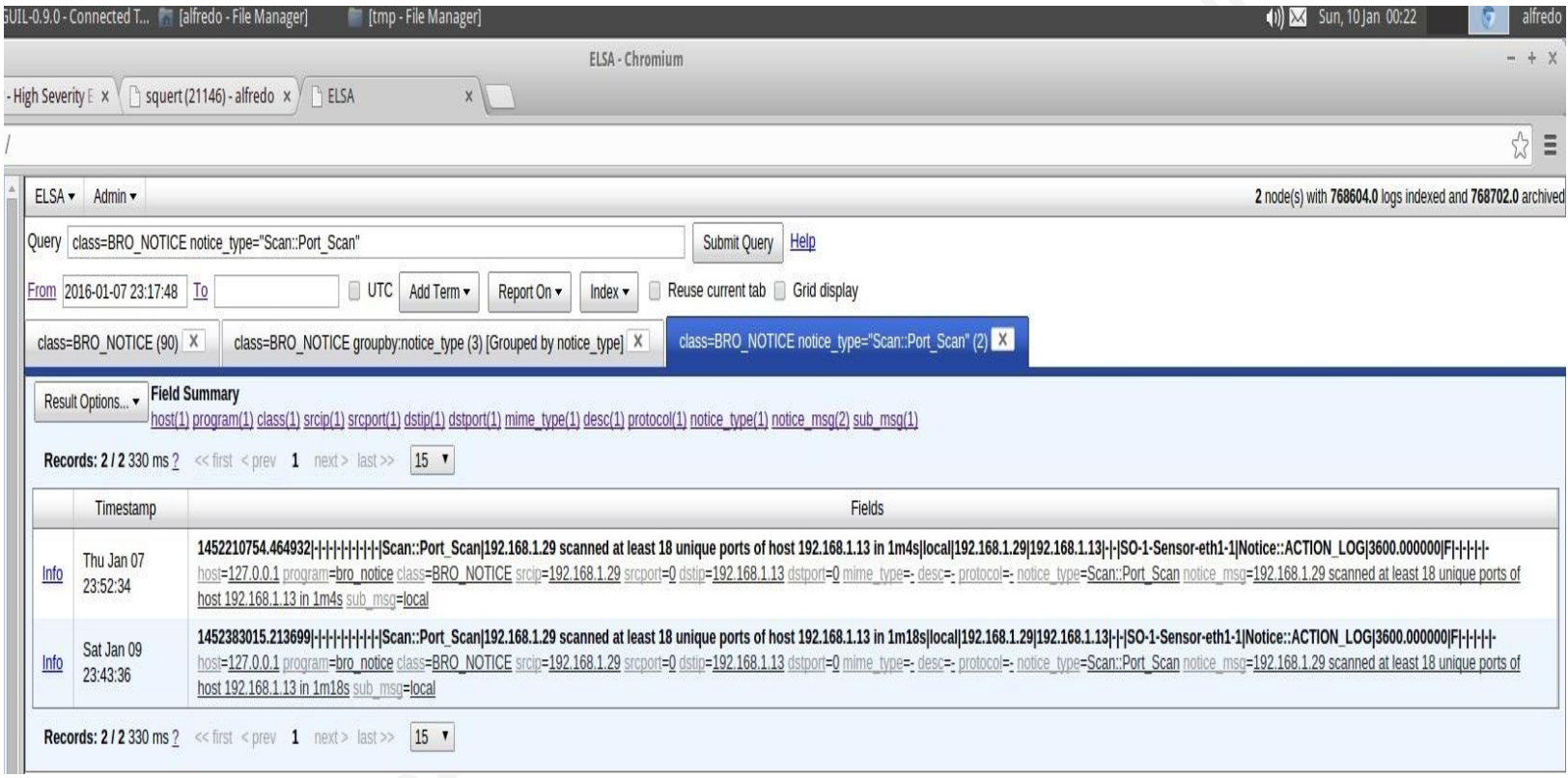


Security Onion has detected a possible reconnaissance attack, which indicates that a network user has scanned the network in an attempt to identify what type of operating systems are on the network.

After the initial ping sweep and operating system detection, the intruder identifies a Linux based server that may contain valuable information. The intruder scans the Linux server for open ports and identifies that the server has the FTP port open. At this step, Security Onion once again alerts to the port scan and provides useful information to security analysts that may have to respond to the breach as depicted in Figure 3.

Alfredo Hickman, ahusmc@yahoo.com

Figure 3. Bro Port Scan Detection Query

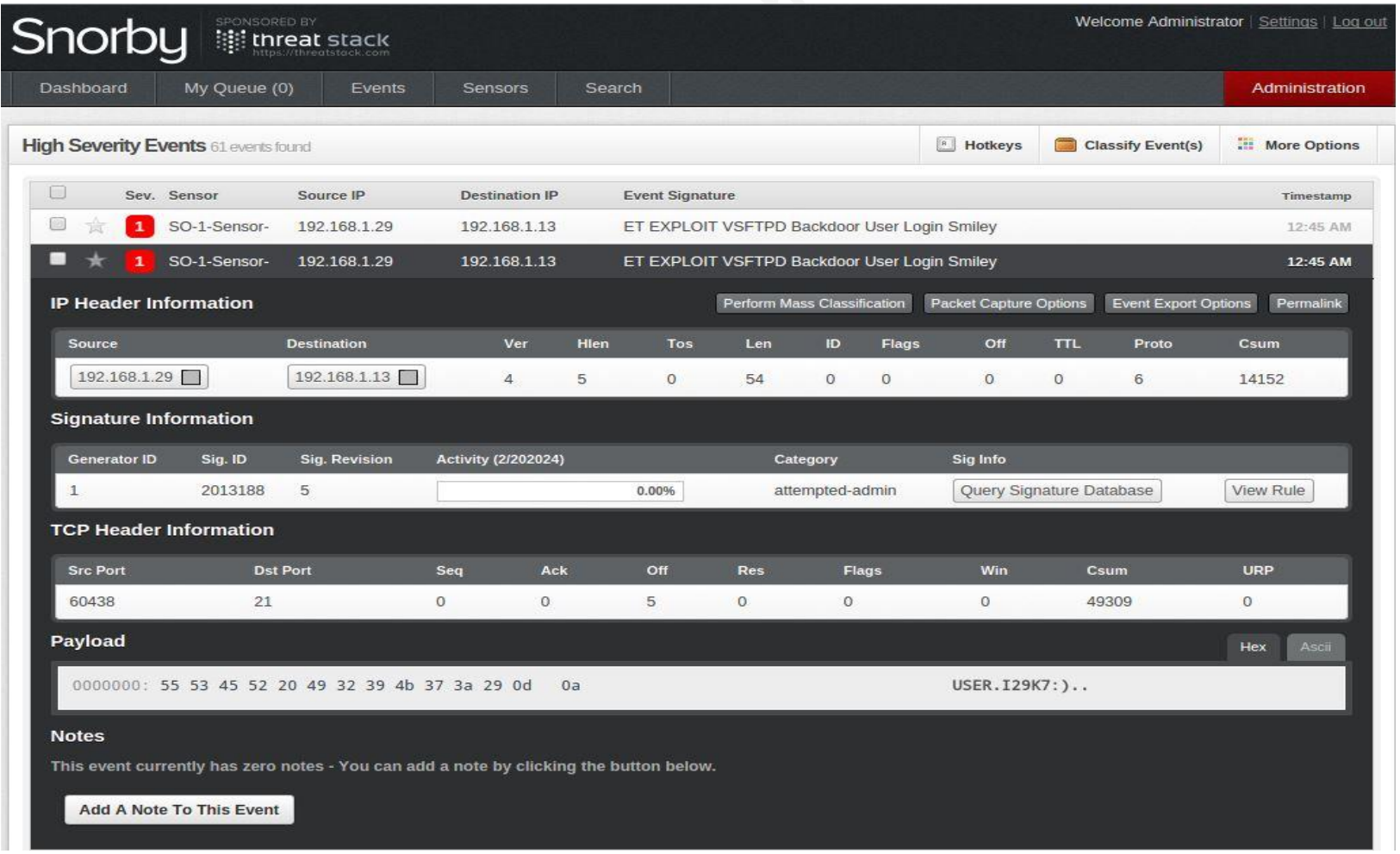


From this Bro query, we can see that Security Onion has detected that the intruder has conducted a port scan on the Linux server at 192.168.1.13, and has determined that there at least 18 unique ports open on the server. This is a potentially valuable finding, as it is common for servers on a LAN to not have a firewall, as many organizations place their trust on edge firewalls and intrusion prevention devices, and do not want a firewall preventing legitimate access to their production servers. Furthermore, from this Bro query, we can determine the IP address assigned to the intruder, which is useful for threat mitigation and incident response.

The intruder has now gained access to the network, scanned the network for available hosts, identified the operating systems on the network, and has now verified open ports on a targeted server. The next step for the intruder is to enumerate running services on the server, and to exploit a vulnerability in order to gain privileged access to the target system. In the next phase, the intruder has verified that the Linux server is

hosting VSFTP, a common file transfer application. And, the intruder has determined that the VSFTP server is vulnerable to a backdoor attack. The intruder successfully exploits the VSFTPD backdoor vulnerability, and gains root access to the Linux Server as depicted in Figure 4.

Figure 4. VSFTD Smiley Backdoor

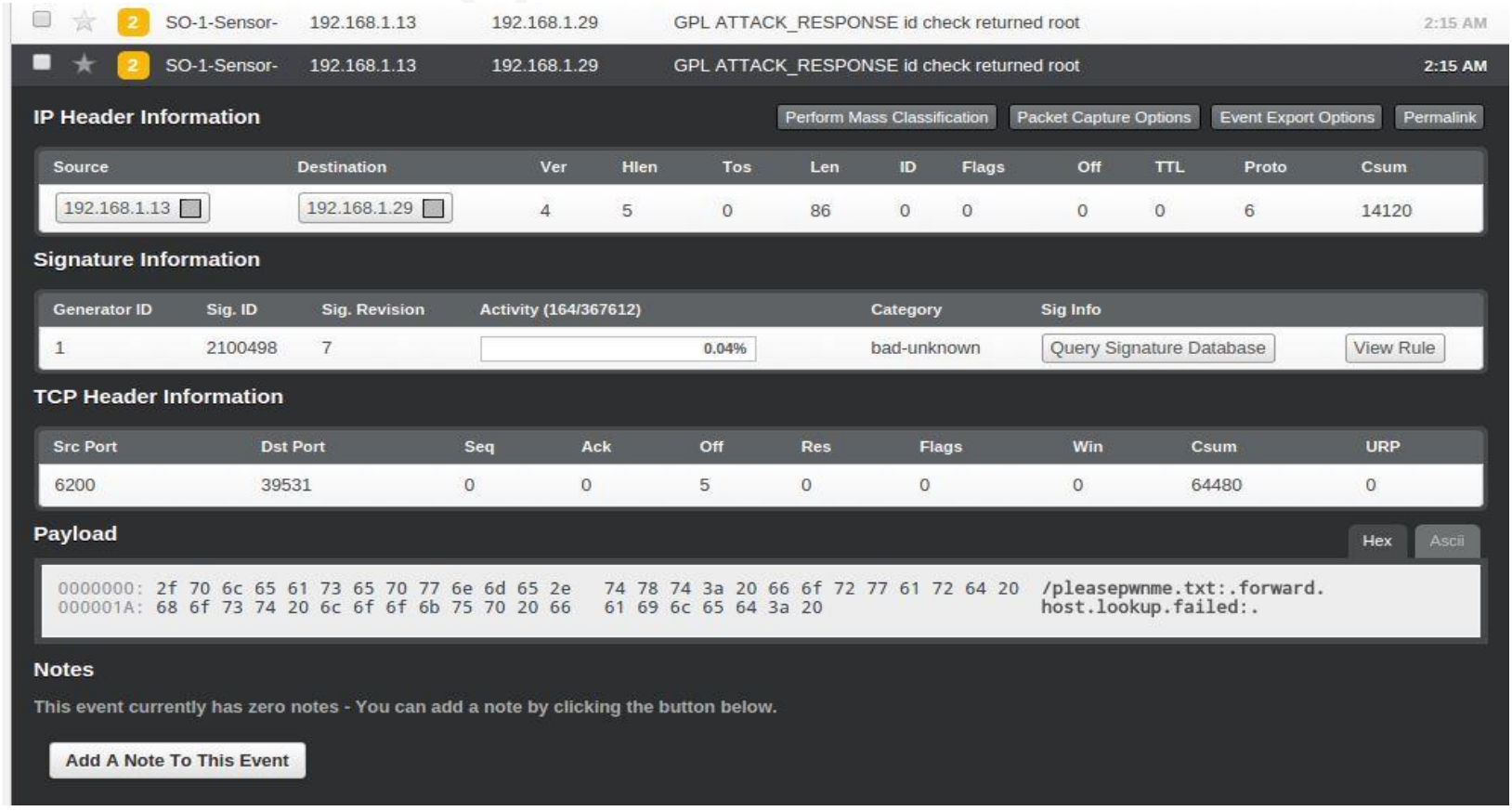


From this alert, we can determine that Security Onion has detected the unauthorized backdoor access to the vulnerable Linux VSFTP server. Furthermore, we can see that the source IP address of the intruder, the destination IP address of the target Linux VSFTP server, and the payload of the VSFTP backdoor attack – the smiley face.

Alfredo Hickman, ahusmc@yahoo.com

The next phase of the attack is for the intruder to identify and exfiltrate any potentially valuable information. In this phase, the intruder having gained root access to the FTP server is now able to draw on any number of resources to aid in their attack. With root access to a trusted system on the business LAN, the intruder can now conduct reconnaissance on the rest of the network, download and install tools from the Internet, and weaponize and deploy malware. Furthermore, the intruder could also gain persistence on the network, identify and pivot to other valuable and vulnerable systems, or simply crawl through the VSFTP server and identify and exfiltrate valuable information. In this case, the intruder navigates to the FTP root and finds a trove of valuable intellectual property, business plans, and personally identifying information. The intruder then collects the information and exfiltrates it as depicted in Figure 5.

Figure 5. GPL Attack Response Data Identification and Exfiltration



Alfredo Hickman, ahusmc@yahoo.com

At this point, Security Onion has alerted to the previous malicious activities, as detailed in the cyber kill-chain. Security Onion has also recorded the malicious behavior, with full packet capture, for further analysis, incident response, and post-breach forensics. However, the value of proactive security gained from the alerts generated by Security Onion is significantly diminished if the security analyst or system administrator is not aware of the malicious behavior on the network as it happens. While it would be ideal for all organizations to have dedicated information security personnel on hand to monitor and maintain the Security Onion deployment and generate relevant threat intelligence, many cannot afford to do so. It is with the small and medium sized organizations in mind, that security Onion implements proactive email alerts. From the Sguil application located on the Security Onion server, email alerts can be generated for wide variety of anomalous or malicious behavior. The Sguil alerts can be configured, as detailed by Deuble (2012), by modifying the Sguild.email file located on the server at: /etc/nsm/securityonion/sguild.email.

6. Conclusion

For too long, robust network security tools have been out of reach for many security professionals at small and medium sized organizations. As such, it is the intention of this research to provide a starting point and road map for security professionals to begin educating and equipping themselves to provide better security and threat intelligence to their organizations and the community. By educating and equipping themselves, security professionals and their organizations can become better and more secure net citizens and can improve the security posture of not only their organizations, but also the entire Internet. However, the discipline of Cyber Threat Intelligence requires more than just installing, deploying, and operating any one technology or tool, such as Security Onion.

Cyber Threat Intelligence, at its core, is much more than just a tool or a technology. Cyber Threat Intelligence is fundamentally a human-centric discipline that, while enabled by technology, is fundamentally driven by human motives, disciplines, and efforts. Furthermore, the benefits of Cyber Threat Intelligence can be exponentially multiplied by incorporating the TTPs of intelligence tradecraft into the operational methodologies of traditional technologies such as network security monitors, security information and event monitors, intrusion detection systems, DNS, cloud hosted platforms, and numerous other technologies.

The benefits of Cyber Threat Intelligence can be further expanded by collecting, processing, analyzing, consuming, and distributing the intelligence products generated amongst other collaborators within larger information networks. Sharing of Cyber Threat Intelligence is a field ripe for further studies, and is one that I hope this research will spurn others to develop. As such, this research has built upon the ideas of other researches that have dedicated themselves to studying and developing more nuanced aspects of Cyber Threat Intelligence and information sharing. Building upon these ideas, this research attempts to provide a high-level overview of a single component of a Cyber Threat Intelligence platform: the network security monitor. Furthermore, this research provides a high-level overview of how to install, configure, and operate Security Onion in a production environment. This research also provides a realistic, albeit simplistic,

scenario of how security professionals can use Security Onion to detect, analyze, and respond to malicious activities on their networks and at various stages of the kill-chain. From here, the path ahead to developing new and innovative approaches to Cyber Threat Intelligence and information security is in our hands.

Alfredo Hickman, ahusmc@yahoo.com

References

Bamford, G., Felker, J., & Mattern, T. (2013, September 1). Operational Levels of Cyber Intelligence (K. Dennesen, Ed.). Intelligence and National Security Alliance.

Bejtlich, R. (2013). The Practice of Network Security Monitoring: Understanding Incident Detection and Response. San Francisco: no starch press.

Burks, D. (2015). Security Onion: Production Deployment. Retrieved from https://github.com/Security-Onion-Solutions/security-onion/wiki/ProductionDeployment

Dennesen, K., Felker, J., Feyes, T., & Kern, S. (2014, March 1). Strategic Cyber Intelligence (R. Borum, Ed.). Intelligence and National Security Alliance.

Deuble, A. (2012). Detecting and Preventing Web Application Attacks with Security Onion. SANS Institute: GIAC Gold Paper.

Gilbert, C. (2014). Scalable Security: Cyber Threat Information Sharing in the Internet Age (Graduate Studies). Stanford University.

Hengel, S., Kern, S., & Little, A. (2014, October 1). Operational Cyber Intelligence (J. Cassidy, Ed.). Intelligence and National Security Alliance.

Kim, P. (2015) The Hacker Playbook 2: Practical Guide to Penetration Testing. North Charleston: Secure Planet

Lockheed Martin. (2016). Cyber Kill Chain. Retrieved from http://www.lockheedmartin.com/us/what-we-do/information-technology/cybersecurity/tradecraft/cyber-kill-chain.html

Nair, S., & Puri, P. (2015). Open Source Threat Intelligence System. International Journal of Research, 2(4), 360-364. Retrieved from http://edupediapublications.org/journals/index.php/ijr/article/view/1791

Regalado, D., Harris, S., Harper, A., Eagle, C., Ness, J., Spasojevic, B., Linn, R., & Sims, R. (2015). Gray Hat Hacking: The Ethical Hacker's Handbook. Ney York: McGraw Hill Education.

Alfredo Hickman, ahusmc@yahoo.com