NTS405: Incident Response; Final Project – SecurityOnion

Jackson Nestler

University of Advancing Technology

Abstract

SecurityOnion is a FOSS project available on GitHub[1] for the purpose of collecting forensic data that will aid in an incident response. Users are given an ISO file that contains everything they need for proper setup, similar to a utility like NightHawk. SecurityOnion is a valuable addition to any environment. Utilizing Elasticsearch, Logstash, Kibana, Snort, Suricata, Bro, OSSEC, Siquil, Squert, NetworkMiner, and an array of other tools, SecurityOnion is a one-stop shop for quick and easy SOC setup.

[1] https://github.com/Security-Onion-Solutions/security-onion

NTS405: Incident Response; Final Project – SecurityOnion

Synopsis: “Security Onion is a free and open source Linux distribution for intrusion detection, enterprise security monitoring, and log management. It includes Elasticsearch, Logstash, Kibana, Snort, Suricata, Bro, OSSEC, Sguil, Squert, NetworkMiner, and many other security tools. The easy-to-use Setup wizard allows you to build an army of distributed sensors for your enterprise in minutes!”

Included in SecurityOnion are an array of tools, including Sguil, Snorby, Squert, and ELSA. Sguil is where most people will spend a their time: it’s where Snort logs are funneled as well as Suricata and OSSEC alerts; Bro logs also end up here depending on your configuration. Sguil comes with a lot of powerful lookup tools (VirusTotal, Malware Domain List, Google Safe Browsing, etc) and can facilitate the capture and viewing of PCAPs in Network Miner and Wireshark. Squert is a web interface for Sguil, but it’s not a replacement entirely. ELSA is a “centralized syslog framework built in Syslog-NG, MySQL, and Sphinx full-text search.” ELSA also shows any host logs from OSSEC, a nifty little feature. In recent updates, SecurityOnion has replaced ELSA with an ELK stack.

To get OSSEC alerts, we need to download the forwarder; it’s a Host-based Intrusion Detection Tool. Doug, contributor to OSSEC, states “when you add the OSSEC agent to endpoints on your network, you gain invaluable visibility from endpoint to your network’s exit point. OSSEC performs log analysis, file integrity checking, policy monitoring, rootkit detection, real-time alerting, and active response.” OSSEC has varying levels of verbosity, with its default being “User Generated Errors” but we’ll crank that up to the most verbose: our SecurityOnion

setup can handle it and we want to demonstrate the full capabilities. Doug states that the "User Generated Errors" have "no security relevance."

Lab Environment

My lab environment is composed of a few different VMs running on the CWR cluster. All IP addresses fall within the 10.5.10.0/24 range and clients receive DHCP addresses from 10.5.10.100-10.5.10.200.

AD01

This is fairly self-explanatory: AD01 is the one and only domain controller in this testing environment. Strutting Windows Server 2016, it's pretty barebones and has a simple GPO for WSUS, along with a basic GPO that informs people that the system is private.

WSUS

A Windows Server Update Services (WSUS) box that simply serves to update any Windows clients on the domain.

Scripting (Windows 10 Client)

This is a VM I use for developing Powershell and Python scripts. It's not domain joined but is still present on the network. We can use this as a "DMZ" machine later on in the testing process.

Kali Linux

SecurityOnion can detect network-based attacks; I've deployed Kali Linux to generate some attack "noise"; in all likelihood the noise will simply be Metasploit attacks against the AD, maybe an EternalBlue attempt on a Windows 7 client. In my reading, I've determined that that these should cause SecurityOnion to alert.

Windows 7

This VM is our sacrificial lamb – it'll be infected with some malware samples I can find publicly on the internet. We'll throw a few well-known samples, as well as samples that US Cyber Command recently released as Russian APT malware: they have a ton of AV definitions, so they'd hopefully be caught by SecurityOnion.

SecurityOnion Setup

As previously mentioned, I've uploaded the provided ISO file to the CWR cluster. I've created a VM with 16GB of RAM, 2 processors with 4 cores each, 100gb of storage space, and a connection to the network that I operate on. The install runs smoothly: I booted, pressed the "Enter" key to begin installation, and the dependencies are being installed right away. Notably I see some incredibly important components like Python, system, dnsmasq, and other packages being installed. This is a typical Ubuntu installation, every single step of the process is exactly the same except where "Ubuntu" would typically be, it says "SecurityOnion."

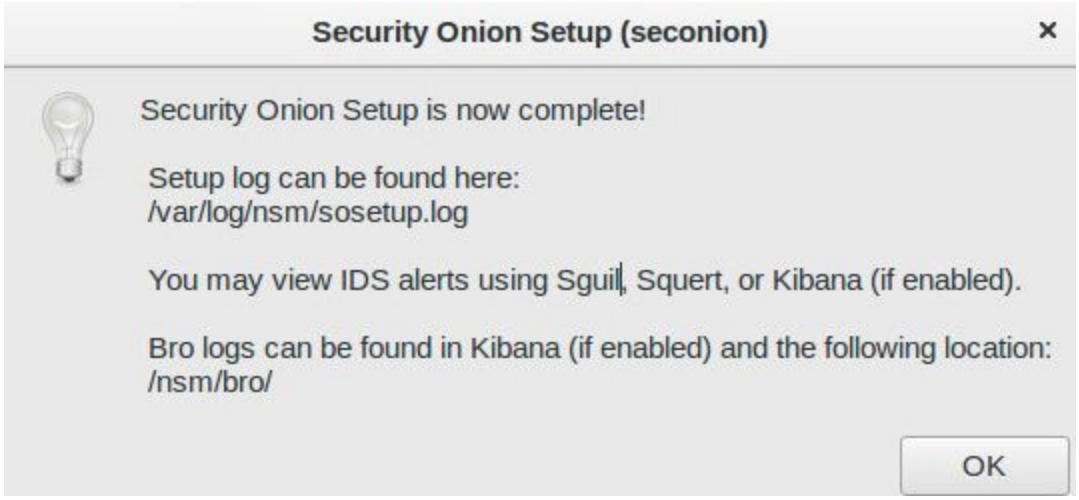
Upon logging in to the new installation, you're expected to run the "Setup" file on the desktop. In doing so, you'll be informed of various configurations that SecurityOnion will perform. If you have a network adapter connected (which is expected) you'll be prompted to set up a static or dynamic address system; I chose static for the sake of simplicity. The configuration wasn't difficult, just basic information about my IP and the network I'm on. A reboot later, the "Setup" file is on my desktop once again and I ran it to continue with the setup that had been aborted by the restart.

I'm then asked about a system type: production or evaluation. Since I want to incorporate this into my environment long-term, I went with production mode. I was asked to choose

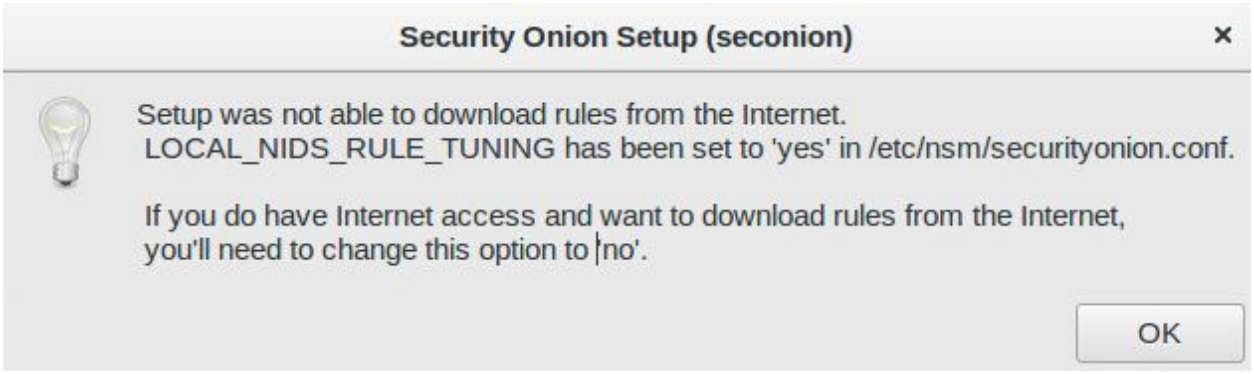
between creating a new SecurityOnion deployment or using an existing: since I don't have an existing, new it is. Next, a user account is setup (mine is admin/password). SecurityOnion can pull IDS reports from various sources, I went with the "Open" set of rules as to avoid paying for API keys.

I got a bit confused when asked about the "HOME_NET" settings, but I believe that's referencing the IP addressing scheme for the areas you'd like to monitor: I set mine to "10.5.10.0/24." Afterwards, the installation wrapped up! My settings can be viewed below:

After the completion of the installation, we get some information about log storage:



I got an error about setup of the rules, so I'll need to apply those configuration changes:



And we've got a note about the UFW being configured for SSH only – I'd like to reach this web panel over HTTP(S) so I'll have to do some firewall configuration in the future.

Utilizing SecurityOnion

SecurityOnion provides ease-of-use shortcuts on the desktop; these can work for our purposes, but in an enterprise scenario the authors highly recommend using Analyst VMs[2] or the use of Brian Kellogg's Ultimate Forensics VM[3]. These will provide a bunch of tools to us, and

[2] https://github.com/Security-Onion-Solutions/security-onion/wiki/Analyst-VM

[3] https://github.com/theflakes/Ultimate-Forensics-VM

the Analyst VM setup is quicker than the “sensor” setup. The “sensors” are grabbing live traffic; the Analyst VMs are not.

At this point, I took snapshots of every VM on the network and prepared to introduce some malware to the systems.

References

Miller, R. (2018, February 28). Security Onion Primer | Insecurity Matters Blog. Retrieved from https://ensurtec.com/security-onion-primer/

Miller, R. (2018, July 30). Security Onion Set Up Part 1: Planning | Insecurity Matters Blog. Retrieved from https://ensurtec.com/security-onion-set-up-part-1-planning/

Miller, R. (2018, June 29). Security Onion Set Up Part 3: Configuration of Version 14.04 | Insecurity Matters Blog. Retrieved from https://ensurtec.com/security-onion-set-up-part-3-configuration/

Miller, R. (2018, July 3). Security Onion Set Up Part 4: Tuning | Insecurity Matters Blog. Retrieved from https://ensurtec.com/security-onion-set-up-part-4-tuning/

Security Onion Solutions. (n.d.). Analyst VMs. Retrieved from https://github.com/Security-Onion-Solutions/security-onion/wiki/Analyst-VM

SecurityOnion Project. (n.d.). Security Onion - Home. Retrieved from https://securityonion.net/