An Integrated Approach for Cyberthreat Monitoring Using Open-source Software

As cyberthreats evolve each day, detecting these threats is becoming more important. Recent studies show that the time between a breach occurring and being detected is, on average, 229 days.1 Since 229 days is a long time, an average company will not respond to an attack in a timely manner and will not mitigate its effects if there is no extra effort used for detection. This number shows there is a lack of accurate cyberthreat monitoring for most companies, and it is mostly because necessary monitoring mechanisms are not placed correctly and/or do not work seamlessly. Additionally, most companies focus on prevention rather than detection. Since prevention methods for most advanced threats fail, the need for detection is becoming more important each day. There are also security investment cost concerns for most small and medium-sized businesses (SMBs). While a not-so-skilled attacker can easily hack a corporate IT infrastructure by using a US $500 exploit that is being sold in an underground market, the cost for preventing or detecting these attacks is not proportional with this low cost when a company chooses to buy and install commercial solutions.

For these types of needs, open-source software presents numerous possibilities since it has great community support and is cost-effective, especially for SMBs. With its advantages, a company may choose to build its security infrastructure using open-source solutions.

An average breach typically consists of seven main steps (figure 1), as modeled by Lockheed Martin and called the Cyber Kill Chain.2 If organizations want to adequately detect attacks, these steps are important starting points to address necessary monitoring needs.

### Figure 1—Seven Steps of the Cyber Kill Chain

<table>
<thead>
<tr>
<th>Steps</th>
<th>Example Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1: Reconnaissance</td>
<td>Harvesting emails, social networking, passive search, IP addresses, port scans, etc.</td>
</tr>
<tr>
<td>Step 2: Weaponization</td>
<td>Developing exploits with payloads, delivery system</td>
</tr>
<tr>
<td>Step 3: Delivery</td>
<td>Spear phishing, man-in-the-middle attacks (MitM), universal serial bus (USB), infected web sites, etc.</td>
</tr>
<tr>
<td>Step 4: Exploitation</td>
<td>Exploiting a vulnerability to execute a code on victim’s machine</td>
</tr>
<tr>
<td>Step 5: Installation</td>
<td>Installing malware on assets</td>
</tr>
<tr>
<td>Step 6: Command and Control</td>
<td>Command channel for remote manipulation of victim’s system</td>
</tr>
<tr>
<td>Step 7: Actions on Target</td>
<td>Data exfiltration, expand compromise, remote “hands-on keyboard” access</td>
</tr>
</tbody>
</table>

Source: Lockheed Martin. Reprinted with permission.

By using this Cyber Kill Chain abstraction, there is a chance to detect an adversary if necessary detection mechanisms are in place, executed and correlated correctly for each step. For example, if a network intrusion detection system (NIDS) is monitoring the active remote connecting IPs for possible command and control (C&C) activity using threat intelligence feeds, it can easily alert the security staff for needed blocking actions. Again, if a host-based intrusion detection system (HIDS) can monitor the host activities (e.g., integrity checking for critical system files), it can alert the security team when a malicious event occurs on the host.

**Network Intrusion Detection System**

An NIDS performs analysis of passing traffic on the entire subnet and matches the traffic that is passed on the subnet to the library of known attacks. By using it effectively, an NIDS can help an organization be alert for attack attempts at various steps of the Cyber Kill Chain model. For example, if there is malware using malicious URLs/IPs, the NIDS will
One of the most notable features of SO is its packet capture capability using the netsniff-ng tool. When choosing to configure the packet capture feature, whenever an intrusion detection system (IDS) alarm is generated, one can easily see and analyze the packet captures of the related event for detailed analysis. Since capturing all traffic consumes a large amount of hard disk capacity, organizations should plan carefully before installing their system. Network bandwidth value and log retention practices can be used as starting points for these plans.

**Host-based Intrusion Detection Systems**

HIDS is an intrusion detection system that monitors and analyzes the internals of a computing system. Different from NIDS, HIDS monitors for host-based activities. For example, it can monitor the integrity of critical files, network connections, system logs, local firewall status, rootkit detection, brute-force attempts to the system and more.

Using HIDS effectively can help an organization detect attack attempts in steps 5 and 7 in the Cyber Kill Chain. For example, step 5 uses the HIDS file integrity monitoring feature, which can detect whenever malware corrupts a system file or write itself to the registry and raise an alert.

One of the more well-known open-source HIDS projects is OSSEC (figure 3). It supports Windows, Linux, Mac, BSD, VMware ESX systems and more.

Its capabilities include centralized management, real-time and configurable alerts, agentless monitoring, and integration with commercial security information and event management (SIEM).

It is also easy to customize since it is open source. OSSEC can be customized for purposes such as USB device white-listing and software vulnerability scanning.
When used effectively, honeypots can help organizations detect attack attempts in step 1 of the Cyber Kill Chain.

There is a Linux distribution called HoneyDrive, which is a bundle of honeypot software and is easy to use to get started. Another well-known open-source honeypot is Dionaea. It is a malware-capturing honeypot initially developed under The Honeynet Project’s 2009 Google Summer of Code (GSoC). Dionaea aims to trap malware exploiting vulnerabilities exposed by services offered over a network and, ultimately, to obtain a copy of the malware. It captures exploits offered over a network and stores details of these harmful events such as source IP, attack type and downloaded binary for later analysis. While an attacker is mounting an attack within this honeypot, the organization can launch a proactive defense using this information. By default, Dionaea supports Server Message Block...
Therefore, using detection services effectively and in a combined manner is important for a well-protected IT infrastructure.

For central monitoring and dashboard purposes, ElasticSearch, Logstash and Kibana (ELK) stack are well-known open-source solutions. They consist of three major components. ElasticSearch is a Lucene-based search server and it provides a distributed full-text search engine. Logstash is an easy-to-use log collection framework that works well with ElasticSearch. Kibana is the ultimate monitoring web user interface and helps visualize all the logs that come from Logstash and are indexed by ElasticSearch.

Using this stack, HIDS, NIDS and honeypot systems can send their data to ELK, and an analyst can correlate these data, create a dashboard for central monitoring and start taking quick actions (e.g., blocking attacker IPs using honeypot data, correlating HIDS and NIDS data to increase accuracy of a detected attack according to kill-chain abstraction). Unless using security data effectively, all the logging efforts are useless.

**Conclusion**

With today’s fast-growing cybersecurity needs, building an effective cyberdefense infrastructure is a big challenge for many organizations. Building
a solid and accurate monitoring infrastructure will decrease the time to detect attacks since it will help gain the necessary insights from systems. A strong monitoring infrastructure will be able to correlate and use data accurately, enabling the security team to only work on important and accurate alarms.

This article provides an overview of open-source tools that can be used to deliver enhanced cyberthreat detection and defense to suit the resources of most cyberdefenders. In addition, these open-source software offerings provide significant flexibility and the benefit of a large support community. This can help to level the playing field for those tasked with guarding an organization and its “crown jewels.” On the other hand, to utilize flexibility and low-budget advantages of open-source security solutions, the security team in charge of installing these solutions should know what they are doing and enjoy the open source community and culture. But open source is also a risk for companies that have small security staffs. Especially in the long term, a product that is no longer supported must be managed by the organization, resulting in unique challenges.

### Endnotes