SANS

A SANS Whitepaper

Detecting Malicious Activity in Large Enterprises

Written by **Matt Bromiley**

September 2020

Sponsored by:

Chronicle

Ensuring information security and the cyber defense of an organization can often feel like an uphill battle with no end in sight. Each week, new data breaches potentially put your users and customers at risk. Meanwhile, the ever-looming threat that any attack could turn into a ransomware outbreak keeps analysts up at night. Then, in early-to-mid 2020, COVID-19 struck and forced global businesses to change their day-to-day operations plans. An unprecedented number of users were forced to work outside of the office, and thus outside of trusted corporate networks.

We're not here to discuss the downsides of remote workers; instead, we want to focus on how organizations can detect threats under these complex conditions. Modern enterprises are extremely diverse. With a geographically diverse user population and a mixture of virtual, physical and cloud-based systems, analysts receive data from many angles. They use a blend of legacy and currently popular technologies. But has security data collection, correlation and analysis kept up with today's environmental complexities? How are organizations effectively detecting suspicious/malicious activity across terabytes or petabytes of diverse data or when a threat actor may jump from on-premises to cloud to legacy technologies and back again?

Organizations are getting increasingly complex, with new technologies layered onto legacy ones. Detecting malicious activity shouldn't be hindered by the size of the organization or the amount of data collected. Your ability to scale detections should match the rate at which your organization collects more data.

In this paper, our mission is to explore **advanced threat detections at enterprise scale**. We focus on techniques to scale organizational growth as well as the explosion in data available to security analysts today. Many detection techniques are rooted in yesterday's logic, focused on single-source concepts or naively reduced to text searches. Think of an IP address, a web domain or a computed hash. These techniques worked—before. Today, however, attackers are quick to morph their malware, introduce new techniques and/or abuse organizations in ways previously unseen.

Before we get started, here are a few questions to consider as you work your way through this paper:

1. How much data is your security team currently generating compared to how much it is analyzing?
2. How does your security team detect threats to the environment? Is it done by you, or for you?
3. Does your team write its own detections, or do you rely on those provided by a vendor or service provider? Can these detections be applied to the entire data set you thought of in Question 1?

If those questions are somewhat uncomfortable to answer, you're not alone. Many organizations have fallen into a state of complacency and have a hard time evaluating how much data they ingest and whether the security team is effectively utilizing that data. They continue to apply old logic against new data and, as a result, fail to detect both basic and advanced threats. Additionally, if your security team has little to no involvement in writing detections, then the organization is only as strong as its security provider. It's time to rethink not only how your security teams write detections but also how they can harness data effectively.

Often, organizations consider themselves too small or insignificant for attacks. Nothing could be further from the truth. History shows us that organizations of all sizes and from all industries are vulnerable. Furthermore, as part of a supply chain attack, smaller organizations may be attacked to gain access to a larger one.

Enterprise Visibility

Before we begin looking at advanced detection techniques, let's focus on one question: **What does it mean to be *enterprise scale*?** Typically, the words *enterprise* and *scale* invoke ideas of a large organization with tens of thousands of assets and massive infrastructure, conjuring up visions of an organization the size of a Google, Apple or Microsoft. We often think, "We'll never be that big!" When it comes to security, however, we need to have the exact opposite frame of mind to ensure success.

When you think of enterprise scale from a visibility perspective, don't think about headcount or the size of your infrastructure. Enterprise scale means encompassing and utilizing all the relevant data points available for detections. Enterprise scale is:

- Recognizing that threat detection is nearly impossible when looking at only a small part of the organization, regardless of its size
- Making threat detections, customized to your environment, that work for *you* and *your team*
- Authoring detections that can scale with the business and the amount of data analysts are examining

Let's consider some of the more recent, ongoing, far-reaching attacker activity. Think about ransomware and extortion, credential harvesting schemes leading to financial fraud and massive spearphishing campaigns with malicious attachments. These types of attacks—which have earned attackers billions of dollars in recent years—are seldom limited to organizations with a large minimum headcount or asset size.[1,2] Even state-sponsored, advanced persistent threats will attack smaller organizations if they fit the target profile—or if the attack will provide access to the real target more easily.

Threat actors have figured out that a significant number of small businesses run the *exact same technology* as the larger organizations. Thus, their attack tools, techniques and procedures (TTPs) work at any organization, regardless of size. Attackers seldom make targeting decisions based on headcount or how many endpoints an environment has. A smaller data set doesn't mean different security problems. In fact, your small organization may be facing security issues very similar to those of a global conglomerate.

Many people assume that large organizations have the funds and the resources to build massive, highly capable security teams that are able to stop any threat in its tracks. But that is not a safe assumption. Facts prove that the larger the organization, the larger the amount of data the security team must sift through to identify threat actor activity. In larger organizations, more departments and resources compete for funds, and security often gets ignored. Some organizations may spend tens of millions of dollars a year on data ingestion but little on analysis.

Regardless of organizational size, the question remains the same: **Are you taking full advantage of all your security data to detect activity within the environment?**

Enterprise-scale detection doesn't mean detections apply only to large or complex organizations; it means that detections are scalable to flex and adjust, supporting the needs of the organization and the flexibility of an attacker.

Piecemeal Visibility Fails

Piecemeal security describes an organization's attempt to detect threats to the *whole* environment by monitoring only a *piece* of the environment. In its simplest form, environments are often thought of in terms of *network* and *endpoint* components. These two parts, though, are forever intertwined; host-based exploits or malware typically rely on network communication. Attackers are unable to maintain a foothold in an environment if they cannot re-enter that environment. If your security model relies on only one of these data sources, how can you ever craft efficient detections?

Or, perhaps worse, in some organizations data is collected from the entire environment but is correlated manually or only when an incident occurs. SIEMs are often treated as catch-all buckets for any and all data, but they are rarely finely tuned to provide value to the security team. Many security analysts spend most of their time correlating data to help provide context to activity within the environment.

There are two critical flaws here.

First, such practices beg the question: Does the "pipe it to a SIEM" model really work? Even if we spend the time to automate log and/or data correlation, we enter into a discussion of log source fidelity and all-around usefulness. If collecting and correlating 1TB of logs a day is the answer, has detection actually improved?

[1] "Business Email Compromise the $26 Billion Scam," September 2019, www.ic3.gov/media/2019/190910.aspx

[2] "South Korean Firm's 'Record' Ransom Payment," June 2017, www.bbc.co.uk/news/technology-40340820

Here's a quick test. Ask the following questions of your security team:

- How long does it take to find the true owner of a system with a DHCP-assigned address?
- How long does it take to trace a DNS request to the true source system?
- How quickly can we identify all the systems a user account has logged into within the past 24 hours?
- After analysts have performed the above correlations, how do we further enrich the data so analysts can make actionable decisions for the environment? Where does that data come from?
- Can you use the logs you collected seven months ago, given that much of the context information (system IPs, names and usernames) may have changed since then?

If any of the above timeframes are measured in hours, or worse, are unknown, then regardless of your organization's size, you still have basic problems to solve. If answering any of the previous questions required finding a metaphorical needle in a haystack, then you have an issue with enterprise scale.

Second, when a security incident occurs, various clocks begin ticking. Depending on data exposure, regulatory requirements and/or organizational risk, your security team may have a limited amount of time to report the incident. Even worse, the more time an attacker has in the environment before they are detected, the more time they have to execute their attack. This period of time—between breach and detection—is known as *dwell time* and was reported to be 56 days for most organizations in Mandiant's 2020 M-Trends report.[3]

Median *dwell time* (the time between initial breach and threat actor detection) was reported as 56 days in Mandiant's 2020 M-Trends report. **Can your organization afford to give an attacker nearly two months untraced in your environment?**

Organization size isn't always a critical factor in cyberattacks. What matters is how quickly an organization can wrangle its available data points to detect malicious activity. The clock starts once an attacker has gained a foothold in an organization, regardless of your organization's size.

Crafting Enterprise Detections

You may be thinking, "But wait—my organization already has detection technology in place! What more do we need to do?" To be fair, many organizations have invested heavily in security technology that provides many of the capabilities we've discussed. We respond to the question about what more an organization needs to do with these questions:

- What level of visibility are your current detection capabilities built on?
- Can your security analysts author their own detections and push them into the environment? If so, what format are your analysts using to write detections?
- Are you detecting based on behavior and fluid TTPs, or are you simply looking for indicators?

[3] TechDemand, "M-Trends 2020 Report for NAVSEA," www.techdemand.io/whitepaper/security/m-trends-2020-report-for-navsea/

Let's throw one more question in there: How far back in the history of your environment can you apply your detections? If you discover an incident at *its very inception*, then hopefully you have enough data to confirm that it's also the *first time* said incident has occurred. What happens if intelligence delivered in the future provides previously unknown context? Is it worth exploring the incident again?

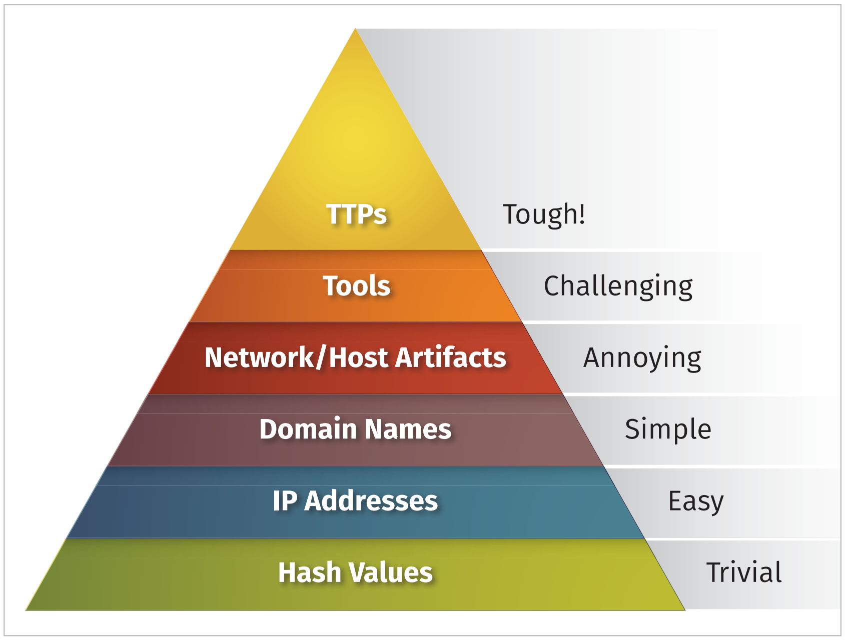
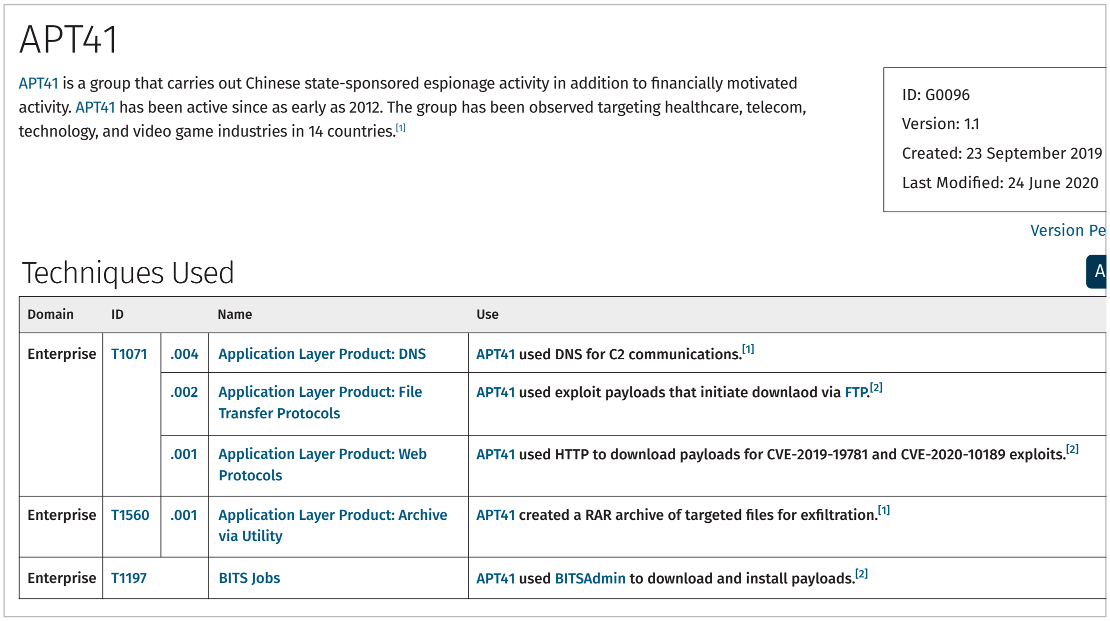


Figure 1. The Pyramid of Pain[4]

These questions, among many others, are designed to make you think about what capabilities your analysts currently have and whether those hinder or empower their daily duties. For example, consider an analyst who has access only to search endpoint data for hashes or IP addresses. These two types of indicators are so easily changed by an attacker that there's absolutely no guarantee you'll find them in time. Figure 1, the Pyramid of Pain (created by David Bianco in March 2013) illustrates the difficulty of finding threat actor indicators.

> You may already have some detection capabilities in your environment. How can your security team leverage those capabilities? Are they extensible to the team or is it a "black box" solution?

In contrast, when you have absolute visibility into your environment, you can begin to think of the way an attacker moves through an environment (behavior) and write rulesets based on those TTPs (the most difficult to find, according to Bianco's Pyramid of Pain). Security vendors may have already created rulesets you can use as inspiration. A myriad of free resources is available that identify attacker TTPs, such as MITRE's ATT&CK® Matrix.[5] Figure 2, for example, provides a snippet from MITRE's group description for APT41, a Chinese state-sponsored threat actor that is still active.



APT41

APT41 is a group that carries out Chinese state-sponsored espionage activity in addition to financially motivated activity. APT41 has been active since as early as 2012. The group has been observed targeting healthcare, telecom, technology, and video game industries in 14 countries.[1]

ID: G0096
Version: 1.1
Created: 23 September 2019
Last Modified: 24 June 2020

Version Pe

Techniques Used

Domain	ID		Name	Use
Enterprise	T1071	.004	Application Layer Product: DNS	APT41 used DNS for C2 communications.[1]
		.002	Application Layer Product: File Transfer Protocols	APT41 used exploit payloads that initiate downlaod via FTP.[2]
		.001	Application Layer Product: Web Protocols	APT41 used HTTP to download payloads for CVE-2019-19781 and CVE-2020-10189 exploits.[2]
Enterprise	T1560	.001	Application Layer Product: Archive via Utility	APT41 created a RAR archive of targeted files for exfiltration.[1]
Enterprise	T1197		BITS Jobs	APT41 used BITSAdmin to download and install payloads.[2]

Figure 2. Snippet of APT41 (Group 0096) from MITRE's ATT&CK® Framework[6]

Bringing together the nuances of your environment (such as subnets, permissions and user/system roles), you can begin to craft enterprise-specific detections sure to catch even the sneakiest of attackers. We'll look at some detection languages you should become familiar with in the "Writing Detections" section of this paper.

[4] "The Pyramid of Pain," updated January 2017, http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

[5] MITRE, https://attack.mitre.org

[6] MITRE ATT&CK® APT41, June 2020, https://attack.mitre.org/groups/G0096/

Understanding and authoring detections should be part of a security analyst's job. Security analysts at your organization should be comfortable at least reading detections, while perhaps more senior analysts should be in charge of authoring them. Enterprise-specific detections provide a variety of opportunities for the security team, as presented in the "Opportunities from Understanding and Authoring Detections" sidebar.

Along with the benefits described in the sidebar, understanding and/or authoring detections also allows the technology within the environment to complement the analyst, rather than the other way around. Without the power to influence what's being detected, analysts are left in a reactive state. Conversely, if they can contribute to what is being alerted on in the environment, they have more incentive to craft very finely tuned, true positive detections.

Opportunities from Understanding and Authoring Detections

Detections are a fantastic method to store institutional knowledge. An analyst authoring a detection *post-breach* is an example of capitalizing on lessons learned. An analyst who reads that detection in the future has absorbed that knowledge. This approach is only effective when you use a proper version control system for detection content that allows you to see your past detection approaches.

Understanding the logical flow of a detection provides insight into the attacker life cycle. There's no guarantee that each analyst will encounter every one of the advanced threat actors in the world. However, crafting detections provides an opportunity for an analyst to think about the attacker life cycle and to put that knowledge into action.

Writing detections reinforces knowledge of the environment. Writing a detection for any old data point is easy—but highly inefficient. Instead, analysts must think about *what* can be detected and *where* the detection needs to occur to be effective. For example, is it easier to detect outbound traffic on each workstation subnet or the firewall? What evidence is available at each point?

Detection life cycle and version control are vital. Your detections will change as your team learns more about a piece of malware or a threat actor—or as data points are enriched by threat intelligence. The same is true about threat actors, given that they modify their TTPs or cease operations due to some form of disruption. Controlling your own detections allows your team to respond to these changes and adjust accordingly.

Writing Detections

Learning to write detections often begins with choosing a language in which to write those detections. Of course, these are true, industry-standard, open source languages, formats that are highly integrated across the wide spectrum of open source and proprietary security tools. Interestingly enough, some formats for detection authoring also exist as indicator storage, depending on usage and applicability. Some examples of detection formats include:

- YARA (a pattern-matching tool to identify and classify malware)
- YARA-L (a language to express detections)
- Sigma
- OpenIOC
- Snort
- Suricata

Most detection formats are open source with immense amounts of documentation. We've chosen to explore two favorites: YARA and Sigma.

YARA

YARA,[7] which stands for Yet Another Ridiculous Acronym, is a tool that allows an analyst to craft rules to look for and identify malicious files on a system. YARA's syntax is very straightforward and allows analysts to create rules easily with a basic text editor.

YARA's original purpose was to help identify and classify malware samples based on common traits. Admittedly, it does very well with atomic or computed indicators that are typically referred to as *file metadata*. YARA has since been expanded with new capabilities, such as expanding compressed files.

[7] "VirusTotal/yara," https://github.com/VirusTotal/yara

Let’s look at the sample YARA rule in Figure 3.

```
rule AbsoluteVisibility {
      meta:
            author = “Matt Bromiley”
            sponsor = “Chronicle”
            version = “1.0”
            last_updated = “2020-07-23”
      strings:
            $string1 = “5up3r 53cr3t” xor
            $string2 = “SANS” base64
            $string3 = /(powershell|cmd.exe)/ nocase
            $string4 = “Q2hyb25pY2xlCg==” wide ascii
            $string5 = “VEhJU01TWUFSQQ==” wide ascii
      condition:
            any of them
}
```

Figure 3. Sample YARA Rule

YARA is relatively straightforward. There are three main fields in a YARA rule:

1. The **meta** field captures any metadata you want to store about the detection. This is a great place to store investigative relevance and implement version control. More on version control in a moment.
2. The **strings** field denotes which values you want to look for in a file. Note that these are not simply text strings. YARA includes modifiers such as XOR searching, base64 conversion and regular expressions.
3. The **condition** within a YARA rule denotes which condition(s) must be present for the rule to be a true positive. This is where the logic of a rule comes into play. Authors can write explicit detections for the values that must be present.

Looking at the YARA rule in Figure 3, one could argue that YARA is often structured around atomic/computed indicators. Admittedly, YARA’s primary usage has been searching for files based on similar rules (to help classify malware).

Another language inspired by YARA has been released called YARA-L, which has the ability to perform searches across vast data and log sets. YARA-L allows for artifact-specific rule crafting, detailed functions to define what activity to alert on and basic logic that tests various values before the rule fires.

Sigma

Sigma, a generic and open source signature format, allows analysts to write detections against any log file present with an accepting SIEM. Sigma rules are written in YAML, which means they are very easy to view and edit on the fly, if necessary. Let’s look at the sample Sigma rule in Figure 4.

```
title: S4 - Sample SANS Sigma Signature
id: a8fe43ed-482a-4cb7-902f-1749eb9e1746
description: A sample Sigma detection, written for this whitepaper!
author: Matt Bromiley
logsource:
      category: process_creation
      product: windows
detection:
      image_name:
            Image: ‘*\Public\*’
      file_name:
            OriginalFileName:
                  - ‘evil_binary1.exe’
                  - ‘evil_binary2.exe’
      condition: image_name or file_name
```

Figure 4. Sample Sigma Rule

Our Sigma rule, while very basic, includes some powerful logic:

- **Metadata** within the rule captures any details an analyst may need to pass on to future analysts and/or to explain why the rule exists.
- By declaring a **logsource**, we are telling the detection what type of data to focus on. Sigma recognizes a wide range of logs and can be used to craft detections against both endpoint and network artifacts.
- The **detection** portion defines what the detection should look for. There are many logic capabilities within this field, allowing for comparisons, base64 encoding and multiple character set possibilities. You'll notice in Figure 4 that we are looking for a process within a particular folder or with an original file name.

In this Sigma rule, we start to break out of atomic and computed indicators, instead focusing on known behavior. For example, in the detection **`image_name`**, we branch out of a single process name, instead looking for a process being run from a folder with the word **`Public`** in it. If we are tracking a threat actor that is known to utilize that folder, this search might provide us a true positive hit, regardless of what executable name the actor uses.

Detections as Code

Finally—with a team of security analysts that have the capability to author, interpret and model detections—you are on the path to creating a finely tuned environment with very little background noise. But, writing detections is only half the battle. As mentioned earlier, writing detections is a great way to capture lessons learned and institutional knowledge for the security team.

In previous examples, we looked at YARA and Sigma as two potential rule formats for crafting custom detections across products. These are simple text formats, but the rules you generate with them will likely change as your security team develops its environmental awareness. It's highly likely, for example, that your first Sigma or YARA rule will generate a significant number of false positives. But, as you learn more about an attacker or a technique, your detections will increase in fidelity and result in fewer false positives. You'll also avoid getting locked into a vendor's proprietary approaches and languages.

As your team continues honing its detections, keeping these files in a common location (such as a **`Git`** repository) will give the entire team access to all detections at any given point in time. Furthermore, inherent change tracking and versioning control ensures that the team is referencing the most up-to-date detections, rather than legacy or incorrect code.

A common repository of detections also allows you to build out additional programmatic steps to automatically integrate detections into the environment upon a code change. Figure 5 provides an example of how this can be orchestrated with some basic continuous integration (CI) concepts.

When the flow depicted in Figure 5 is in place, detections and rulesets are treated by your security team as code instead of loose files. Code repositories add benefits such as version control, capability to revert and view changes, as well as track which member of a team made changes. Your team now can observe attacker TTPs as they change over time.

Visual Anomaly Detection

Absolute visibility into your environment is only the beginning of advanced detection techniques. With enriched and correlated data in your arsenal, you can also begin to look for the "unknown" within your network. The human eye is adept at detecting anomalies in a pattern, also known as *visual anomaly detection*. Many analysts prefer to view data in a graphical form, because the eye is capable of identifying patterns that technology may not easily be able to.

Take Figure 6, for example, which represents DNS requests from a single system over a 10-minute period.

Without much background knowledge, the human analyst can easily observe that there were three spikes in requests during the applicable time frame. There were also minor upticks in between, with little time between the latter two periods of spiked activity. Determining what happened during that timeframe is where your analysts will need to rely on enriched, contextual data.

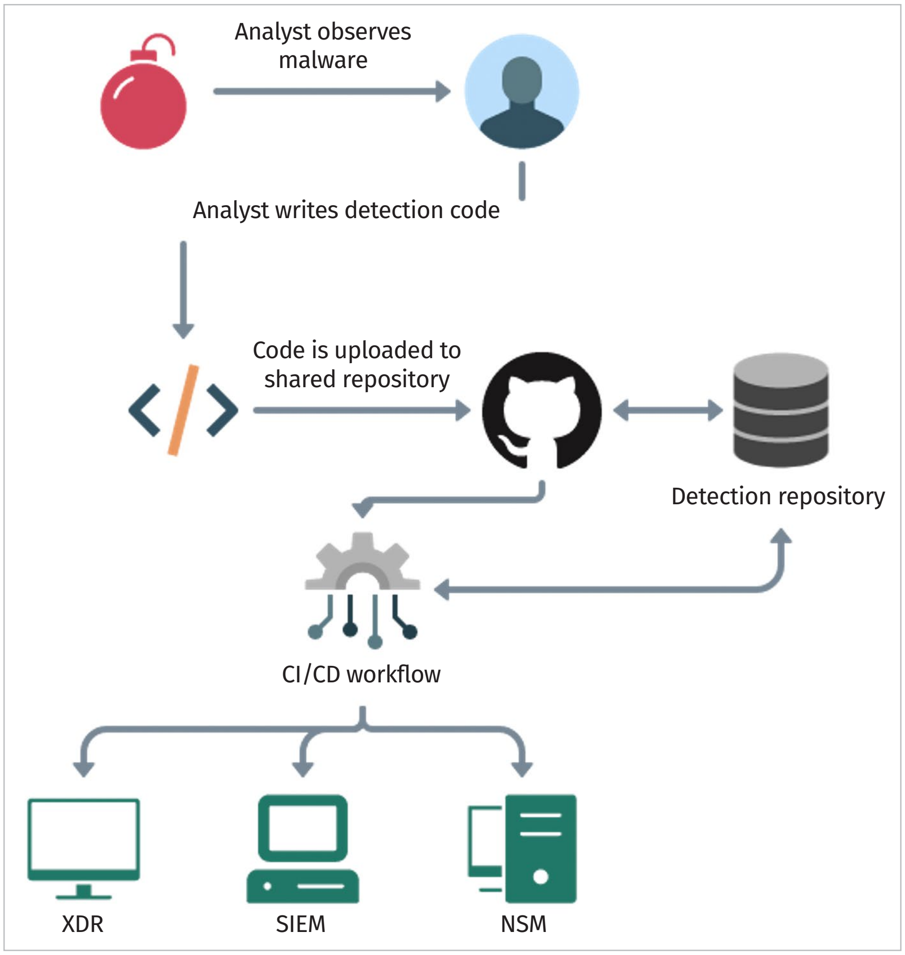


Figure 5. Flow of Detections as Code

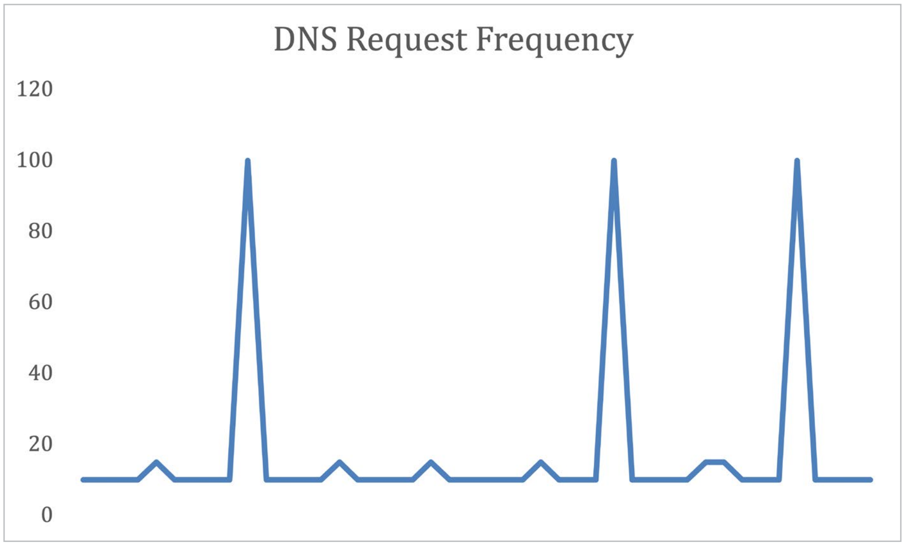


Figure 6. DNS Requests from a Single System

Closing Thoughts

In this paper, we took a strong stance on detecting malicious activity within an organization. After watching data breach after data breach make headlines, we believe it's time for some changes in how malicious activity is detected within an environment. We believe the following:

- Analysts are best empowered when they have absolute visibility into the environment.
- Analysts should be the caretakers of environment-specific detections built to harness absolute visibility and multiple data points.
- Detections are most effective when they are tuned to the environment and the experience of the analysts—and when they are treated as code, ready to be operationalized instantaneously.

Of course, these are not goals that can be achieved overnight or with the push of a button. Increasing your visibility and shifting your detection capabilities require significant time and process investment from the security team. However, it is an effort that will yield benefits to the entire organization for years to come. Your security posture will be so improved that malicious activity will stand little chance of success.

Visibility and empowered analysts also allow an organization to shift from a reactive to a proactive stance. With programmatic data enrichment and correlation, analysts will need to spend less time correlating data. With well-written detections, analysts will waste less time investigating false positives. That time savings can be directly converted into finding even more techniques to keep the organization safe and secure.

About the Author

Matt Bromiley is a SANS Digital Forensics and Incident Response instructor, teaching FOR508: Advanced Incident Response, Threat Hunting, and Digital Forensics and FOR572: Advanced Network Forensics: Threat Hunting, Analysis, and Incident Response. He is also an IR consultant at a global incident response and forensic analysis company, combining his experience in digital forensics, log analytics, and incident response and management. His skills include disk, database, memory, and network forensics, incident management, threat intelligence, and network security monitoring. Matt has worked with organizations of all shapes and sizes, from multinational conglomerates to small, regional shops. He is passionate about learning, teaching and working on open source tools.

Sponsor

SANS would like to thank this paper's sponsor:

Chronicle
now part of Google Cloud