BINARY DEFENSE

WHITEPAPER

A Threat Hunter's Checklist

What A Security Team Needs To Know To Get Started

Table of Contents

If you've been around the information security community, you've probably heard the term "Threat Hunting" and considered how you can apply these techniques to enhancing the security of your organization's network and computer systems. In this whitepaper, we're going to describe what threat hunting means, how you can get started, and what you're going to need along the way. The assumption in this whitepaper is that you are an IT professional, and have some familiarity with the basics of security.

What is Threat Hunting?

The threat hunting that we'll be discussing is a proactive, regularly-repeated security exercise to find attacks and computer intrusions that have been evading detection on your company's computer systems. This is done by searching across many sources of event data (network traffic, server logs, process trees and behaviors from endpoints, etc.) looking for patterns of unusual behavior or looking for uses of attacker techniques. This is different from searching for specific IP addresses, domain names or file hashes that are known to be used for attacks. That type of searching should be automated, freeing you to focus on finding the patterns that uncover attackers hiding on your systems.

Get Ready: This is Going to Be Fun!

It is helpful when learning any complex new skill to take it one step at a time and focus on having fun implementing each small piece before moving on to the next. To get started in threat hunting, it isn't necessary to set up a complex system of database servers or implement an expensive system right away. You don't need to handle all the possible sources of event data and tackle everything at once. Instead of being overwhelmed or frustrated by too much data, start with just one or two inputs and treat each new source of data as a challenge and opportunity to learn and explore what really happens on your network. There are sure to be surprises along the way, but if you treat each one as an adventure, it can be fun ot learn!

Threat Hunting in Context

When you're designing a security program for your company, it's helpful to start with an attacker's mindset: how would you go about breaking in if you were going to attack your company's computer systems without being caught? This should quickly lead you to realize that any defensive system you put in place is simply an obstacle to be overcome. As a defender, your job is to make it as hard as possible for anyone to break in, force them to slow down to confront multiple layers of security obstacles, and set up sensors and traps (honeypots, honeytokens and other deception technologies) all along the way so that you can detect attacks as early as possible in the cyber kill-chain and stop them before they get very far. It is important to focus on some basic and critical security controls as a first priority. Threat hunting is most effective when you are already blocking the most common and pervasive threats automatically, because you'll be able to focus on detecting targeted attacks.

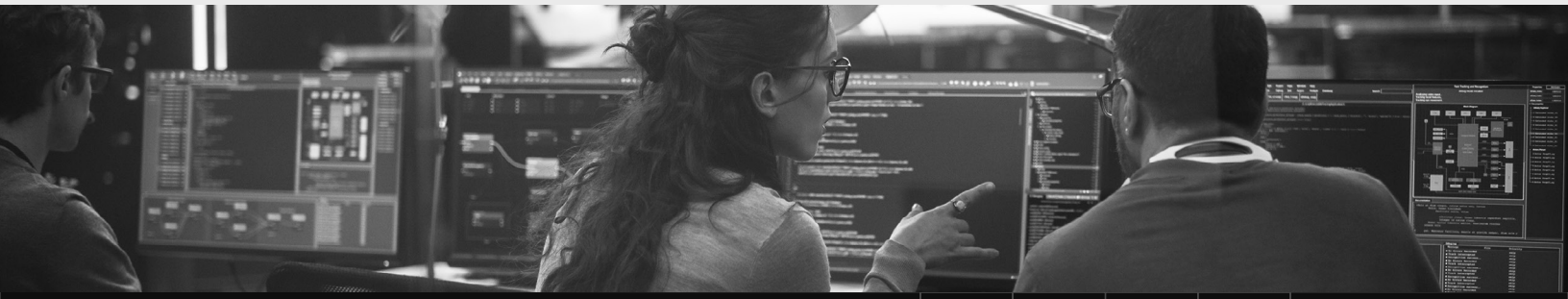
Cover the Basics

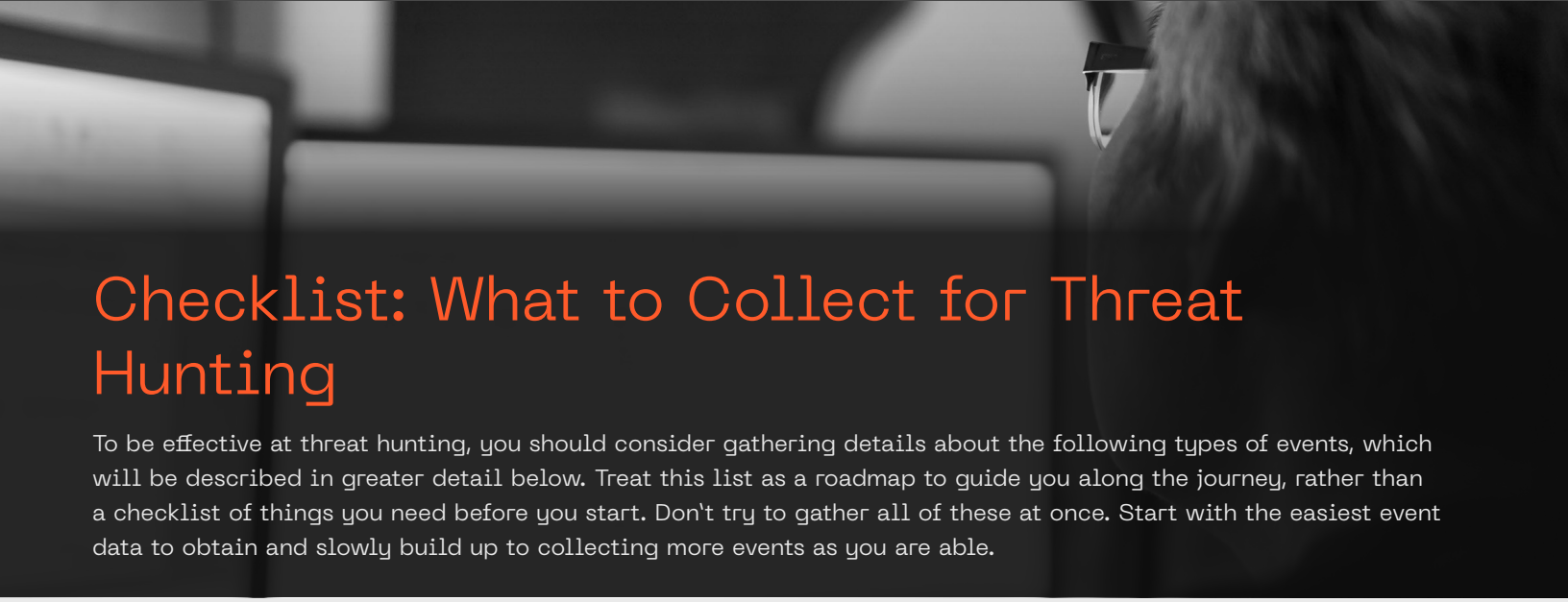
An in-depth discussion of the other layers of defense you should have in place is too broad of a topic for this whitepaper, but you should definitely consider the following suggestions.

- Because attackers often abuse passwords, implement Multi-Factor Authentication (MFA) everywhere that you can.
- Design your network to be segmented, with critical servers and databases separated from employee workstations, different departments segmented from each other, and everything protected from public Internet-facing servers.
- Keep in mind that the most likely point of initial intrusion will be from an employee workstation or a server responding to requests from the Internet, so treat those as potential threats at all times.
- Strong email scanning and threat filtering is a must, because so many threats arrive via email.
- Forcing all internal computers to use the company's DNS server and go through a web proxy that checks domain reputations for all HTTP requests is also critical to success. This is important once you realize how many malicious backdoor RATs rely on DNS and HTTP requests to communicate with their command and control server.
- You should identify special requirements for the few employees who need to use SSH or SFTP to connect to outside servers and block all others from doing so. Block all other outbound or inbound network connections that aren't required.
- Never assume that the ports will always be used by the protocol that is "supposed to" be on that port—malware often tries to send HTTP or a custom protocol over ports 53, 80 or 443.
- Keep antivirus updated on all endpoints but accept the fact that its purpose is just to detect well-known threats—it is trivial for any targeted attack to be completely undetected by antivirus solutions.
- Do your best to keep all software up to date with security patches, with higher priority on patching critical-severity vulnerabilities that allow remote code execution or authentication bypass on Internet-facing computers.
- Turn on Windows Firewall for all workstations to make it more difficult for an attacker who lands on one workstation to be able to move laterally to others.

Start Expanding Detection Capabilities

Once you have the most basic controls implemented to automatically block and minimize the chaos from all the commodity malware that blindly tries to infect everything it can reach, you are in a position to make your security team more effective and able to start hunting for targeted attacks. If you don't take care of the basics first, it will be far too easy for attackers to hide in the noise of the constant barrage of attacks and your team will be too busy putting out fires to pay attention to the subtle signs of an intrusion. However, don't let that stop you from starting to set up for threat hunting all along the journey of setting up security controls. Even if you find many non-targeted attacks in the early stagesxa, that can help you and your team understand where your security controls are weakest and prioritize which controls you should implement next.





Checklist: What to Collect for Threat Hunting

To be effective at threat hunting, you should consider gathering details about the following types of events, which will be described in greater detail below. Treat this list as a roadmap to guide you along the journey, rather than a checklist of things you need before you start. Don't try to gather all of these at once. Start with the easiest event data to obtain and slowly build up to collecting more events as you are able.

1. Endpoint Events

- [ ] Process start and stop events
- [ ] Command-line arguments supplied to processes
- [ ] Process hierarchy (parent process of each process)
- [ ] Details of modules (DLLs) loaded by processes
- [ ] File hashes (including import hashes) of every executable program run and every module loaded
- [ ] Digital signatures (or lack thereof) and version metadata for each process
- [ ] Files written by processes
- [ ] Selected registry keys and values created and written to by processes
- [ ] DNS lookups and network communication by processes
- [ ] Security events including logon, logoff and identification of remote hosts
- [ ] Web browser history

2. Network Events

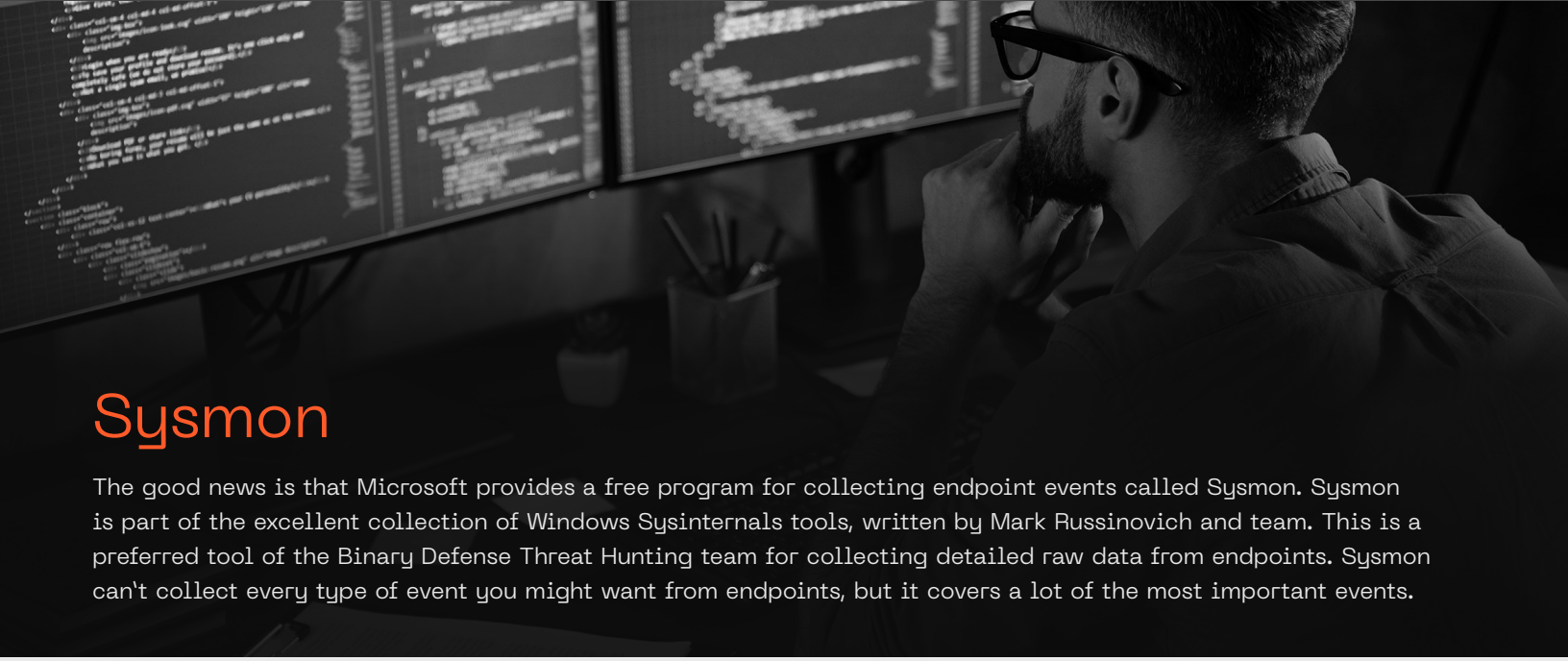
- [ ] Netflow data (metadata about all network connections, not the content)
- [ ] Identification of network protocols detected
- [ ] Extraction of executable files sent across the network

3. Server Log Events

- [ ] DNS log of queries and responses (all types, including TXT records)
- [ ] HTTP proxy records of requests and metadata about responses
- [ ] Firewall logs (allowed and blocked connections)
- [ ] DHCP logs to correlate LAN IP address with hosts
- [ ] Domain Controller authentication logs
- [ ] Email server logs, including metadata for email messages sent/received

Endpoint Events

The most valuable data to collect is the record of what happens on workstations and servers, which gives you the best visibility of not only how attackers are accessing systems, but also potentially what they are trying to steal. That's because no matter what the attacker tries to do to disguise their network traffic by proxying through Google servers or encrypting traffic that blends in to web browsing, they still have to do something on the workstations or servers to achieve their goals.



Sysmon

The good news is that Microsoft provides a free program for collecting endpoint events called Sysmon. Sysmon is part of the excellent collection of Windows Sysinternals tools, written by Mark Russinovich and team. This is a preferred tool of the Binary Defense Threat Hunting team for collecting detailed raw data from endpoints. Sysmon can't collect every type of event you might want from endpoints, but it covers a lot of the most important events.

Download Sysmon here: https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon

Sysmon version 11 was released in April 2020 and includes some great new features, including improvements to DNS event collection and the ability to save copies of files that are deleted or overwritten. Most of the malware dropper programs that we analyze quickly delete files that are part of the malware installation process—being able to recover those files for examination can be very helpful when investigating a security incident.

Configuring Sysmon

The first thing you need to know about Sysmon is that you need to create a configuration file to specify the events you want it to collect and report on and exclude the events that you don't need. You might think, "I'll just start with the defaults, or by collecting everything and ignore what I don't need," but that would be a mistake! The sheer volume of detail that is available from Sysmon can be overwhelming to sift through if you enable everything. It is much better to start small with a deliberately crafted configuration file to collect just a few types of events, understand how to search through those events effectively, and then add additional collection one at a time. Get started with example Sysmon configuration files with these community guides:

- Swift On Security configuration example: https://github.com/SwiftOnSecurity/sysmon-config
- Olaf Hartong Sysmon Configuration Modules: https://github.com/olafhartong/sysmon-modular

Review the Sysmon configuration from Swift On Security. You will quickly appreciate why it is important to set filters to exclude common events that are not useful for threat hunting, but which generate a lot of background "noise" that you'll have to sift through.

TrustedSec, Binary Defense's sister company, also has a comprehensive community guide to configuring and using Sysmon which is an excellent resource: https://github.com/trustedsec/SysmonCommunityGuide

Sysmon saves all of the event data that it collects to the Windows event logs, which you can view using the Event Viewer (look under Applications and Services Logs/Microsoft/Windows/Sysmon/Operational). In order to search through the event records across multiple endpoints, you'll need to forward the logs from each endpoint to a central server to collect all the events in one location. There are many ways to manage this, depending on what log aggregation solution you decide to use. To start, read the Microsoft guide to Windows Event Forwarding (WEF) here: https://aka.ms/WEF

For more comprehensive solutions, you may wish to look into Graylog, Splunk, and ELK (Elasticsearch-Logstash- Kibana), or read about an open source project called HELK that is built specifically for threat hunting: https://www.confluent.io/blog/sysmon-security-event-processing-real-time-ksql-helk/

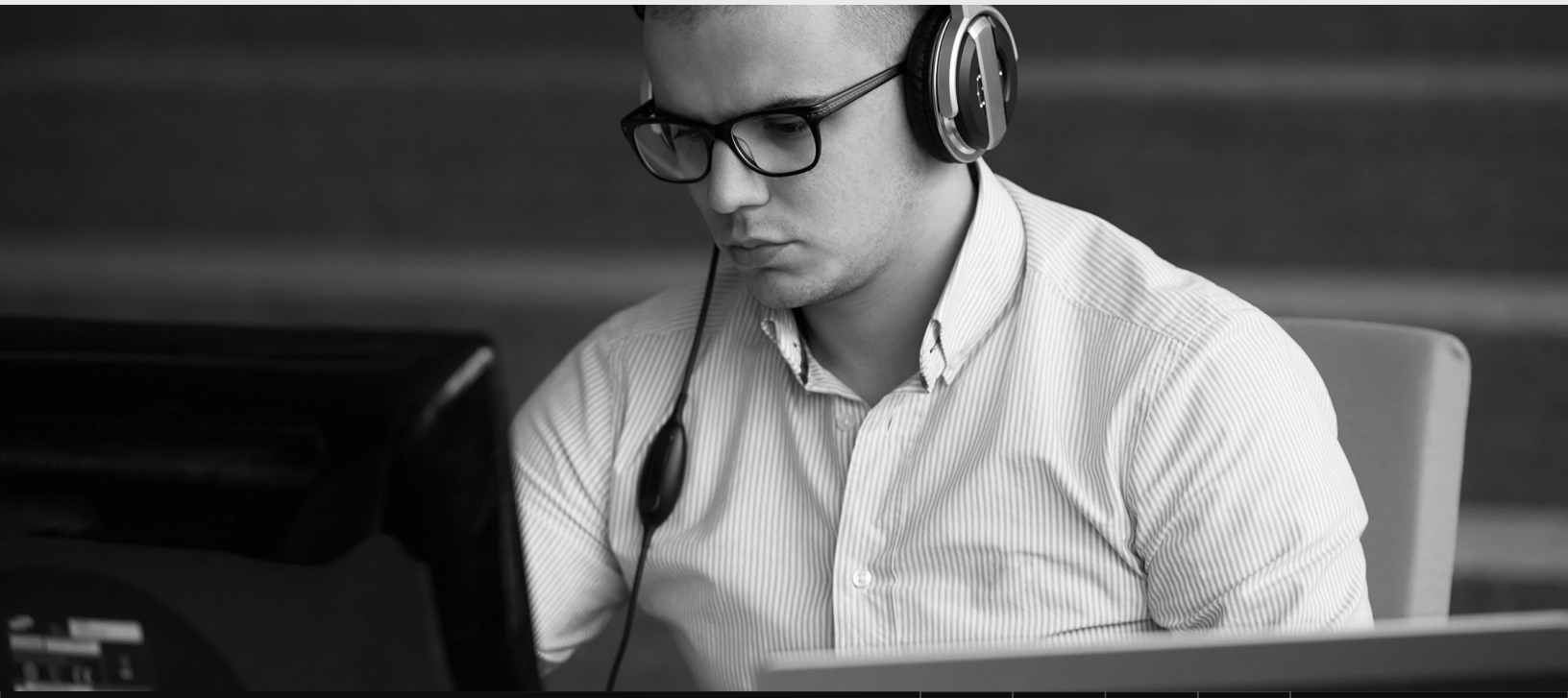
What do the endpoint events tell you?

Once we have information about processes on endpoints stored in some sort of database and a method for querying those events, we can now answer important questions to help us hunt for threats.

The types of questions that we query for evolve over time as threat actors change techniques, but some examples of queries that have been useful for a long time include:

- Are there any instances of an Office product (EXCEL.EXE, WINWORD.EXE) as the parent process spawning an unusual child process, including regsvr32.exe, rundll32.exe, powershell.exe, wscript.exe, cscript.exe, etc.?
- Did any program load a DLL with the same filename as a known system library, but from an unusual directory, or from an unsigned DLL file?
- Has regsvr32.exe spawned a PowerShell process?
- Have any of the system tools been copied to a new folder or renamed before being executed, especially to folders under the user profile?
- Has wscript.exe been used to run JScript files?
- What new scheduled tasks have been created?

The answers to these questions (and many other similar queries) will lead to discovery of completely normal, legitimate processes and scripts as well as possible malicious activity. The key is to first collect the data so that you can run queries, then understand an attacker technique well enough to write a query to detect it, and finally understand (or learn) what is normal in your environment so you can filter out those false positives and then closely inspect whatever is left over to determine whether it is an attack.



Registry Key Creation/Writing Values

A myriad of malware artifacts can be found in the registry. Monitoring all registry activity would be impossible, even on a single Windows machine, due to the sheer volume of normal interactions with the registry that happen all the time. To appreciate the scope of the challenge, download and run the Process Monitor (ProcMon) program from Microsoft System Internals (https://docs.microsoft.com/en-us/sysinternals/downloads/procmon) and observe how many events fly through the user interface every second, most of which are registry events. However, if you have a specific technique to look for, you can filter the registry events that you're monitoring (see the Swift on Security Sysmon template above for some examples).

Command Line Auditing

Capturing the command-line arguments to programs that are executed on endpoints tells a much more complete story about what happened during an intrusion. Many of the attacks that antivirus misses, but we catch, make use of built-in system tools such as wscript, cscript, PowerShell, certutil, rundll32, and others that are not suspicious on their own. The only indication that they were used for an attack comes from analyzing the command-line arguments that were used to run those programs. Windows has a Group Policy setting that allows the details of command-line arguments to be captured in event logs, as part of event ID 4688 (process creation). The setting is disabled by default, so you have to be proactive about turning it on, which requires enabling two policy settings:

1. Enable Audit Process Creation to see event ID 4688. Edit the policy located in: Computer Configuration > Policies > Window Settings > Security Settings > Advanced Audit > Configuration > Detailed Tracking
2. Enable the policy setting "Include command line in process creation events" located in: Administrative Templates\System\Audit Process Creation. Change the setting to "Enabled"

More detailed instructions and examples can be found in the Microsoft documentation: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/component-updates/command-line-process-auditing

Example Registry Hunt

2020-1048, a Windows Print Spooler Elevation of Privilege vulnerability. Exploiting this vulnerability to gain persistence for malware on a system is as simple as running one command in PowerShell as an unprivileged user and then "printing" an executable file to the new port. The command is: Add-PrinterPort -Name c:\ windows\system32\ ualapi.dll

To detect attackers using this technique, you need to search for file-based printer ports set up in the registry: HKEY_ LOCAL_MACHINE\ SOFTWARE\Microsoft\Windows NT\ CurrentVersion\Ports. Examine the registry values to find any references to executable files, especially filenames ending in .exe or .dll, and investigate anything found.

Many other malware techniques for persistence can be found in the registry. Sometimes even the malware binary executable file itself, encoded or encrypted, is stored in registry values. Being able to search for keys and values in registry hives is a crucial capability to support threat hunting.

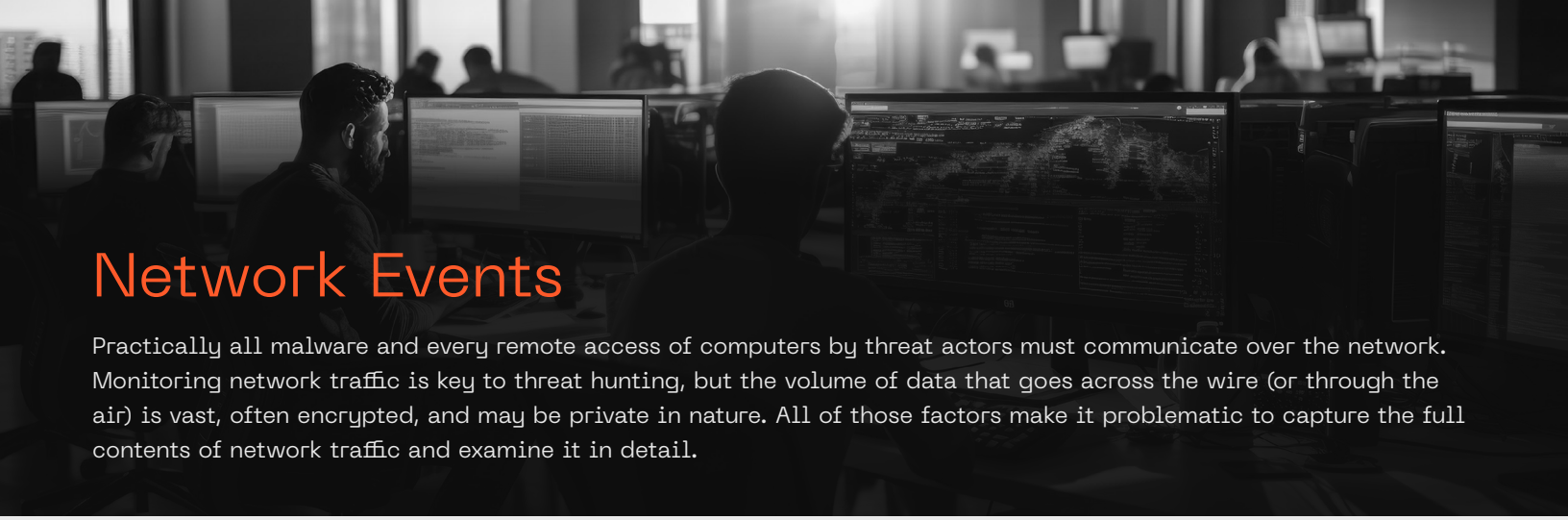
An important point to note is that there are both system registry hives and user-specific registry hives. The user hives, stored in "ntuser.dat" files in the user profile directories, are only loaded when the associated user account logs on. Therefore, it is important to collect event information about registry write events when they happen and store those in a central location or database to search later, rather than relying on live searches of the registry on endpoints

Logon/Logoff Events

You can capture local machine account logon and logoff events from the Windows Security Event Log without any special installation or setup. You should forward these events to a centralized log collection server and not leave them stored on just the endpoint. If the events are only on the endpoint, they are easy for an attacker to clear and remove evidence of their remote logon. Domain authentication events are recorded on the Domain Controllers, which will be discussed later.

Browser History

If collecting web browser history is allowed within your company privacy policy, you may wish to consider using a tool that can export browser history to XML files and set up a procedure to collect those XML files to a central logging server. Searching for downloads of zip files, exe files, vbs, jse, ps1 or other script files (particularly from WordPress sites or other unusual sites) can yield useful results that may indicate the second stage of malware installation. One such tool for exporting browser history is available from Nirsoft: https://www.nirsoft.net/utils/browsing_history_view.html



Network Events

Practically all malware and every remote access of computers by threat actors must communicate over the network. Monitoring network traffic is key to threat hunting, but the volume of data that goes across the wire (or through the air) is vast, often encrypted, and may be private in nature. All of those factors make it problematic to capture the full contents of network traffic and examine it in detail.

Netflow For the Win!

Instead of capturing all of the contents of network communication, start with just the metadata about the connections themselves. In other words, which IP addresses connected with other IP addresses at what times, on what ports, and how much data did they transfer? Which side initiated the connection? How long did the connection last? This data is called netflow. Network engineers are familiar with the idea of netflow as a diagnostic tool by taking representative samples of network traffic metadata. For threat hunting, we don't just want samples— we want information about every network data flow that can be captured.

The metadata may not seem like valuable information without any content, but by looking at the patterns of network communication you can identify many useful things. For example, you can detect if an internal workstation or server is making outgoing connections at regular intervals over a long period of time to an external IP address. This type of regular "check-in" or "beacon" behavior might be normal for some software, or it might indicate a remote access trojan or backdoor program checking in with its Command and Control (C2) server for instructions.

For example, try this experiment:

1. Create a list of the top ten internal IP addresses to external IP address pairs that had the highest number of outbound connections.
2. Create another list of the top ten internal IP address to external IP address pairs that had the highest ratio of outbound bytes sent to bytes received. Most normal connections from work stations receive about 10 times the amount of data that they send (small web requests result in large amounts of HTML, graphics and scripts downloaded). We're looking for the opposite pattern—small inbound commands and large outbound transfers, indicating that files are being sent out.
3. Create another list of the top ten longest-duration connections.

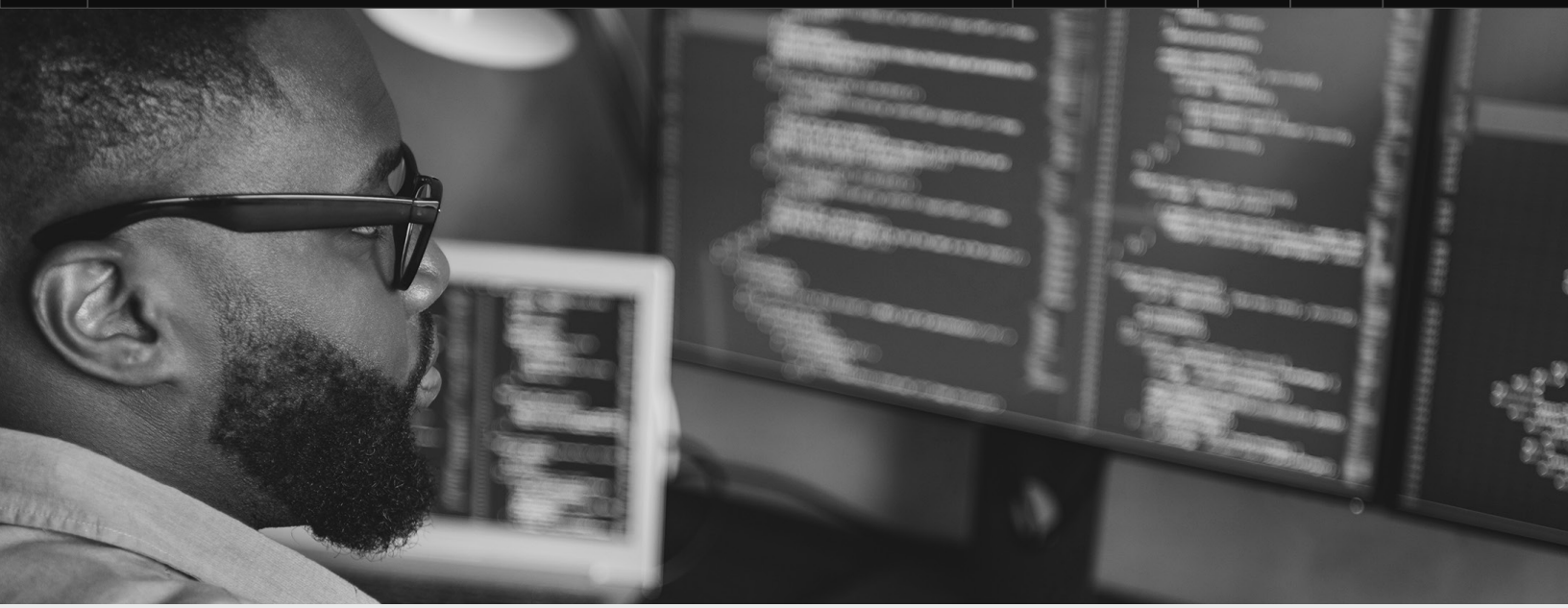
Now look at the IP addresses that appear in at least two of your three lists and dig deeper into those connections to determine if there is anything suspicious about those connections. You'll probably identify some legitimate use cases that are exceptions to the normal patterns, but once you understand those exceptions and add them to an "ignore" list, the remaining connections that are unusual are worth looking into. Look in the netflow records for any workstation making connections directly to other workstations. Unless employees are hosting file shares on their workstations (which is probably a bad idea) there shouldn't be any SMB connections from workstation to workstation. However, attackers quite often try to move laterally from computer to computer using stolen credentials once they are inside.

It's also worth querying for the connections going to external hosts on unusual ports.

For example:

- Look for outbound connections to external hosts (or anything other than your company's DNS server) on port 53. If you find anything, ask yourself why you aren't blocking that and funneling all DNS requests through your own server.
- Look for outbound connections to external hosts on port 445 and ask hard questions about whether it is really necessary to allow outbound SMB connections, when attackers could use that to steal authentication hashes.
- If it is unusual for some of the employee groups in your company (for example, the Accounting group) to connect to SSH servers, search for connections on port 22.
- Look for any port that you don't recognize and figure out what it is. Attackers quite often send connections to the "wrong" port for the protocol.

If you have netflow data, you can also identify patterns that indicate an internal computer in your network has been set up as a proxy or relay, by noting the pattern that every time some data is sent to that machine, it immediately sends the same amount of data to another computer (likely outside your network). This type of internal proxy is often used by attackers to "pivot" through your network to reach computers that aren't normally accessible directly to the Internet (for example, servers in a card data environment processing payment transactions).



Detecting Threats in Contents of Network Traffic

It may be too much work (or too invasive of privacy) to attempt to have an analyst examine all of the contents of network traffic flowing through your company's network, but that doesn't mean you can't deploy some software to peek into the traffic and find patterns that are useful to recognizing threats. The open source software Zeek[1] (formerly called Bro) is capable of detecting network protocols being used over unexpected ports, parsing DNS requests and responses, detecting executable files being sent over the network, pointing out "weird" network traffic that stands out from other traffic, and recognizing many other extremely useful patterns. Instead of storing all the network traffic, Zeek produces logs of results that can be searched or ingested into databases. Zeek detects and categorizes network traffic—it is not an intrusion detection or intrusion prevention tool.

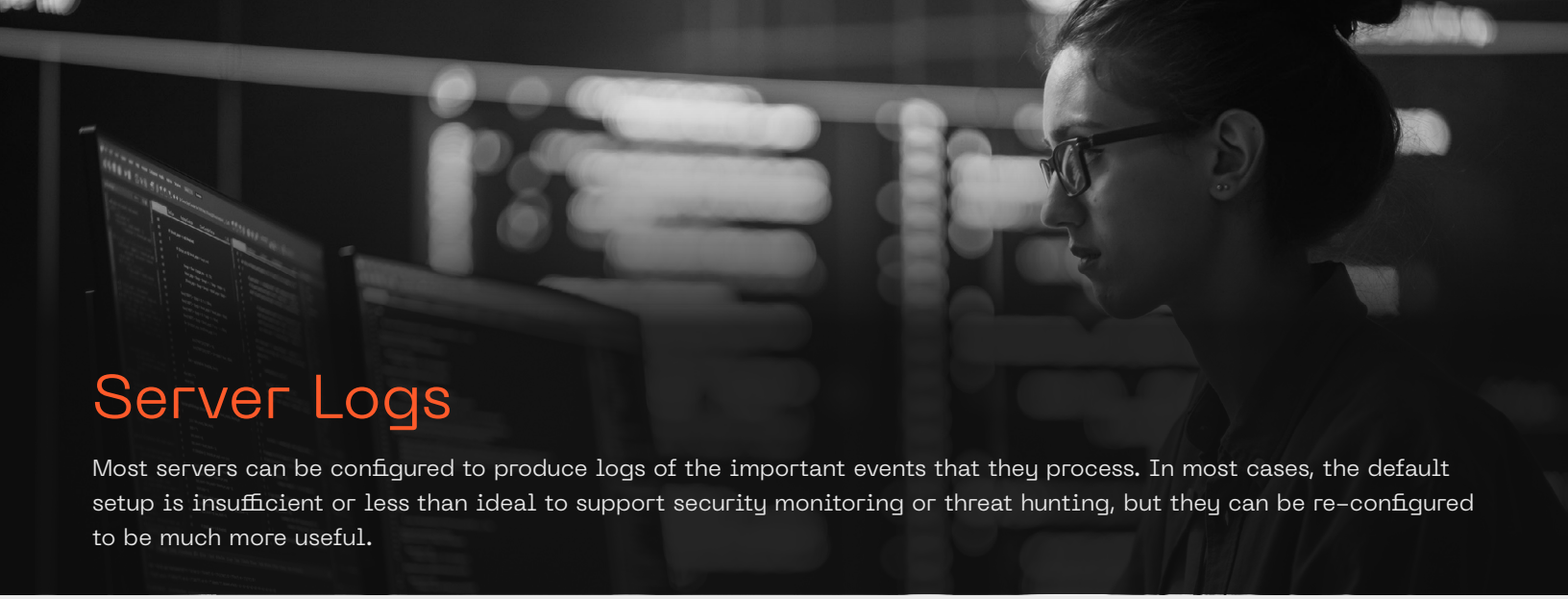
Another open source project, RITA[2], uses the output logs from Zeek to detect patterns of network communication that are similar to malware "beaconing" behavior. If you have Zeek set up, RITA is a great tool to help you quickly identify patterns that are worth a closer look.

Suricata[3] is another open source tool that examines network traffic and can alert on, or even block detected threats (if you set it to "intrusion prevention" mode). You can write your own rules to detect patterns of network communication that you want to be alerted to. For example, here is a Suricata rule that detects possible attempts to exploit CVE-2020-11651, the SaltStack authentication bypass vulnerability:

alert tcp any any -> any 4506 (msg:"ET EXPLOIT Possible Saltstack Authentication Bypass CVE-2020-11651 M1"; flow:established,to_server; content:"_prep_auth_info"; reference:url,labs.f-secure.com/ advisories/saltstack-authorization-bypass; reference:cve,2020-11651; classtype:attempted- admin; sid:2030071; rev:1; metadata:affected_product Linux, attack_target Server, deployment Perimeter, signature_severity Major, created_at 2020_05_01, updated_at 2020_05_01;)

Some open source and commercially-available threat intelligence feeds integrate well with Suricata and can alert you to patterns of malware communication, not just specific indicators of compromise such as IP addresses that can change rapidly.

1 Zeek: https://zeek.org/
2 RITA: https://www.blackhillsinfosec.com/projects/rita/
3 Surricata: https://suricata-ids.org/
4 Open Source Suricata rules: https://rules.emergingthreats.net/open/suricata/rules/



Server Logs

Most servers can be configured to produce logs of the important events that they process. In most cases, the default setup is insufficient or less than ideal to support security monitoring or threat hunting, but they can be re-configured to be much more useful.

There are too many different types of server logs to cover them all, but some that you should strongly consider include:

1. DNS log of queries and responses (all types, including TXT records)
2. HTTP proxy records of requests and metadata about responses
3. Firewall logs (allowed and blocked connections)
4. DHCP logs to correlate LAN IP address with hosts
5. Domain Controller authentication logs
6. Email server logs, including metadata for email messages sent/received

DNS Logs

Assuming your network is set up to only allow DNS queries to your company's managed DNS server and you are blocking, to the best of your ability, DNS over TLS and DNS over HTTPS as well as standard DNS queries to external servers, you can learn a lot by hunting through the DNS logs. These logs can seem overwhelming at first, because practically every single connection across your network involves some sort of DNS lookup to support it.

Enriching DNS query records with domain registration information (WHOIS records) allows you to search for domain names that were recently registered, domains that are not in the Alexa top one million website list, domains that consist of random, non-word sequences of letters, or domains that are registered through suspicious registrars such as domains4bitcoins.com (hint: look for domains whose name server/NS records match the pattern *.bitcoin-dns.hosting).

Some malware actually uses DNS requests and responses as a communication channel to retrieve commands (or even encoded, second stage malware downloads). Typically, these use DNS TXT type queries, which allow the server's response to contain hundreds of bytes of text. Look for unusual patterns, such as one internal host making DNS requests to hundreds or thousands of different "subdomains" for one suspicious domain, such as:
TXT Query: aaa.sysprofsvc.com
TXT Query: aab.sysprofsvc.com
TXT Query: aac.sysprofsvc.com ...and so on for hundreds of subdomains, with responses consisting of very long strings of text that could be Base64 encoded (or use some other encoding).

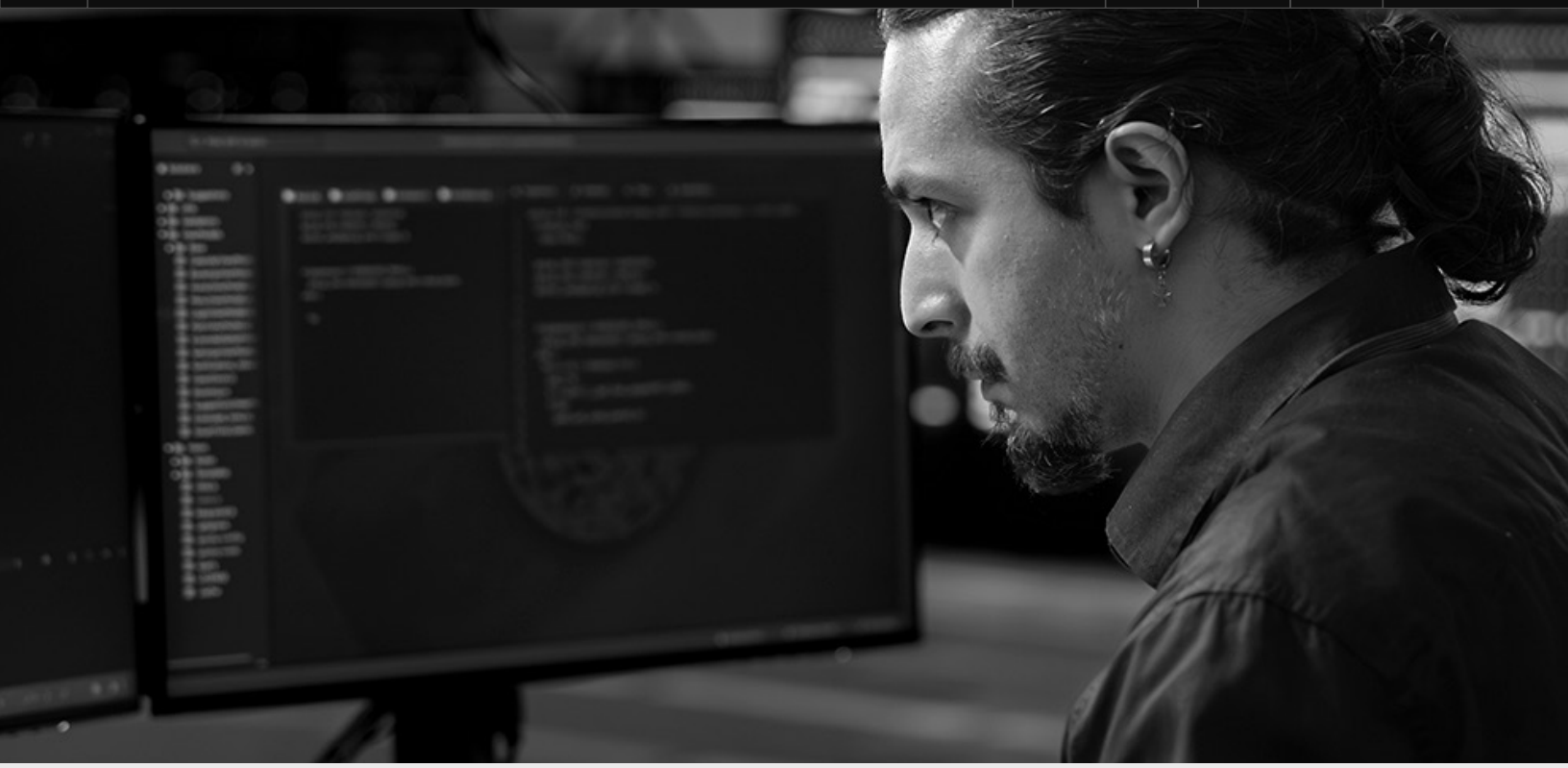
As you can see, it is important to have visibility into DNS requests and responses in order to get detailed answers to your threat hunting searches.

HTTP Web Proxy Logs

Many malware variants communicate with command and control servers over HTTP to blend in with normal web browsing traffic. Quite often, the pattern of requests for beacon check-ins looks fairly normal, but you can sometimes spot unusual patterns by looking for POST requests that upload large amounts of data. Unless an end user is uploading a lot of photos to a legitimate photo-sharing site, or sending files to a cloud storage provider, the pattern of frequent, ongoing large uploads through HTTP POST is worth investigating further. Once you have identified a particular pattern of malware check-in HTTP requests, you can search for that pattern in your proxy logs. Be sure that your search capability allows you to search requests using regular expressions to take full advantage of pattern matching without depending on exact string matches. Malware often incorporates some randomness to each request, but even the random parts follow some sort of pattern that a regular expression can match.

Firewall Logs

Events from your firewall can show you patterns of connections that were blocked and connections that were allowed. It is interesting to note if any IP address made a lot of attempts to connect that were blocked, followed by a few that were allowed through. Overall, the amount of noise from constant port scanning on the Internet can fill up your firewall log with a lot of excess records. Using an external source of threat intelligence to filter out the background noise of the Internet can be helpful to finding more useful patterns. Firewall logs are most useful once you've identified a suspicious IP address of interest and you just want to know exactly when it communicated and with which internal hosts and ports. exact string matches. Malware often incorporates some randomness to each request, but even the random parts follow some sort of pattern that a regular expression can match.



DHCP Logs

Dynamic Host Configuration Protocol (DHCP) logs are not as useful for finding patterns, but they can be very helpful once you've found suspicious traffic (especially if it was from long ago) to tie internal IP addresses to particular machines and be able to investigate the hosts.

Domain Controller Authentication Logs

In addition to the local authentication logs from endpoint security events, it is useful to search for unusual patterns in user authentication for domain accounts. Some useful patterns to detect include authentication at unusual times of day, weekends or holidays if an employee was not expected to be working, or remote authentication from IP addresses in parts of the world that the employee was not in. This can be very difficult in a large company with a global workforce working hard around the clock, but even then, it may be possible to detect an employee logging in from two different parts of the world if it would be impossible to travel between those locations in the time between logins.

Email Server Logs

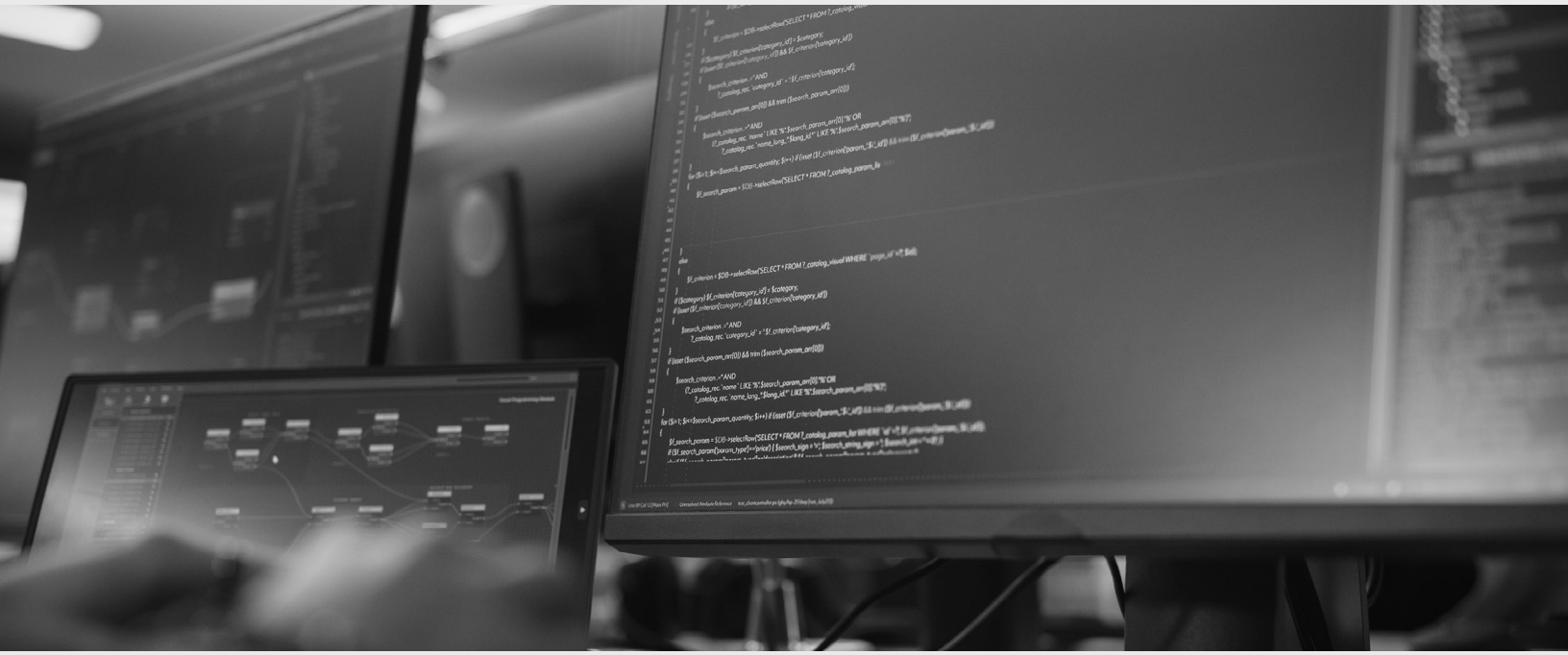
Since most threats arrive via email, it can be helpful to be able to search records of incoming email to find messages from suspicious external senders, containing links to suspicious websites, or with unusual attachments such as zip files containing scripting files (.vbs, .jse, etc.) or executable files. In most cases, the best solution is to use email threat scanning, threat intelligence and filtering to block dangerous messages from even getting through to end users. But if one of your employees reports an attempted phish, it is very helpful to be able to quickly identify which other employees received the same or similar message.

Other Logs

Every organization is different and will have other types of logs available. If you have a web server, definitely include HTTP access and error logs in your collection. If you have a VPN, that's a great source of logs, too. If there are custom-developed applications, see if there's a way to tell the developers what kind of information you want them to capture and send to a log.

Honeypots

In addition to all the passive logging of events, one of the best sources of information that helps you threat hunt comes from honeypots deployed inside your network. These consist of servers listening for incoming connections on your network that don't serve any business use but appear to be important to attackers as they explore and learn your network "by feel" through port scanning and discovery. Any interaction with the honeypot at all should trigger alerts and let you quickly investigate to determine who is exploring the network. A discussion of all the different types of honeypot servers, honey ports, honey tokens and honey files (on endpoints) is beyond the scope of this discussion, but the book " Offensive Countermeasures: The Art of Active Defense" by John Strand and Paul Asadoorian is an excellent introduction to this topic and practical guide for deploying active defenses in your environment.



Conclusion

We've covered a lot of ground, and yet barely scratched the surface of what is possible with threat hunting. It's a fun topic to dive into, because there is so much to learn and great opportunities for innovation. The best approach is to start small by collecting just a few logs from a small subset of sources, build your query capability and threat hunting skills, and then start adding to the number and variety of sources of raw information that you can search. Before you know it, you'll be so familiar with what "normal" looks like in your environment that any attacker will have a hard time blending in and evading your watchful eyes.

Ready to take the next step?

Our threat hunters operate with hypotheses based on how adversaries think, from initial access to lateral movement to exfiltration paths.

Start Hunting

BINARY DEFENSE

Binary Defense is a trusted leader in Managed Detection and Response (MDR), helping organizations across industries stay ahead of modern cyber threats. Our team of SOC analysts, threat hunters, detection engineers, and researchers works around the clock to deliver outcomes that matter: earlier threat detection, faster response, and stronger long-term security.

We take an attacker's mindset to defense, combining deep expertise with proactive tradecraft to protect what matters most.

Learn more at binarydefense.com, explore our blog for the latest insights, or follow us on LinkedIn.

600 Alpha Parkway, Stow OH 44224
sales@binarydefense.com