aws marketplace

SANS

Enabling a Threat Hunting Capability in AWS

Learn how to conduct effective threat hunting in your Amazon Web Services (AWS) environment.

AWS Marketplace Introduction

Threat hunting offers proactive ways to detect anomalous behavior in your environment, but it is a journey with many considerations to make along the way. In this whitepaper, SANS analyst, Shaun McCullough walks through the threat hunting process and how it should fit into an organization's overall security strategy. He also discusses what data to gather, options for analyzing it, and the kinds of tools threat hunters can use in their cloud environment.

Building on McCullough's perspective, AWS Marketplace will share how you can begin this process to your AWS environment. They will provide an introduction to relevant software seller solutions that can enable your threat hunting journey through efficiencies and enhancements. Finally, Sumo Logic will be featured as an available option that can facilitate your threat hunting program.

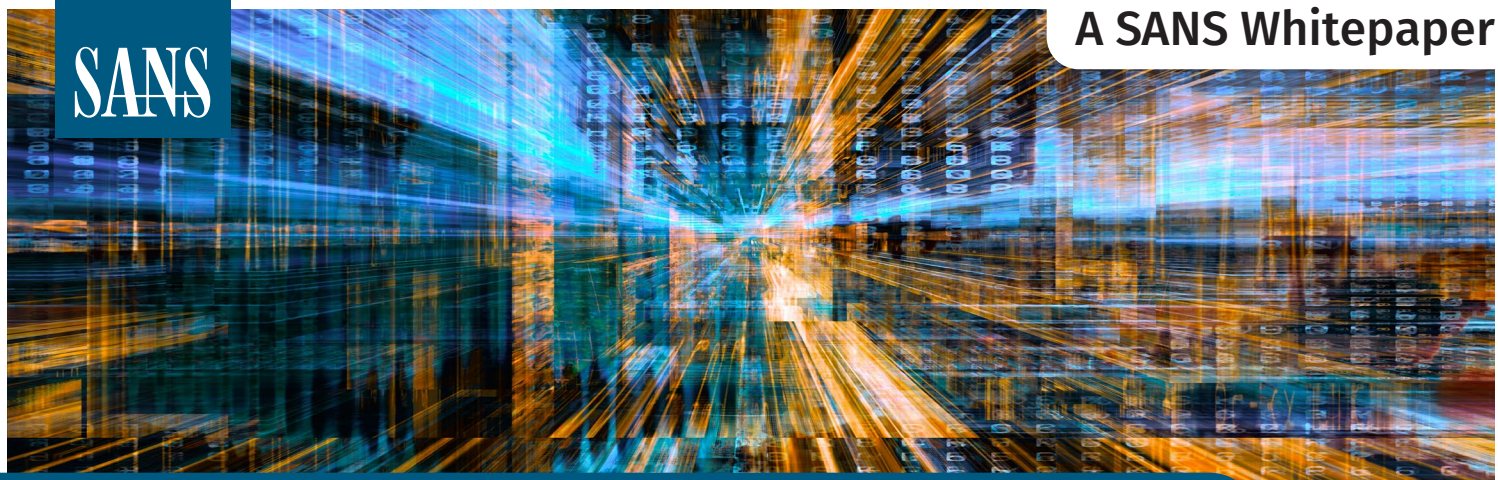
The featured Sumo Logic solution for this use case can be leveraged in AWS Marketplace:

sumo logic

Sumo Logic Cloud-Native Machine Data Analytics Service (Annually)
Continuous intelligence across your entire application lifecycle and stack

SANS

A SANS Whitepaper



How to Build a Threat Hunting Capability in AWS

Written by **Shaun McCullough**

November 2019

Sponsored by:

AWS Marketplace

Introduction

The infrastructure is built, a patching plan is in place, firewalls are locked down and monitored, assets are managed, and the SOC team is responding to alerts from the security sensors. When basic security hygiene is implemented, the threat hunting team needs to start evaluating infrastructure for any threats and undetected breaches.

Because infrastructures are complex, with many moving parts, teams need a plan to manage all the data from all the various operating systems, networking tools and custom applications. They also need to know which threats to look for, how to prioritize them and where to start hunting.

Cloud environments bring their own set of complexity and peculiarities for threat hunting. Customers realizing the benefits of elastic environments may find that systems that had a threat on Friday are terminated on Sunday. Reliance on cloud services likely means relying on the data they offer in a platform-specific format. In addition to the cloud, the management plane is now a new threat vector that teams have to consider, along with web apps, virtual machines and databases.

In this paper, we walk through the threat hunting process and how it should fit into an organization's overall security strategy. We discuss how to determine what data to gather, options for analyzing it and the kinds of tools threat hunters can use in cloud environments.

Threat hunting

The proactive evaluation of the infrastructure operations to detect a threat beyond the deployed security tools

Threat Hunting on Premises vs. in the Cloud

It is vital to understand the process of threat hunting and how to approach it differently than standard security operations. Let's look at this process in the context of a web application. To enhance understanding, this paper references a common use case found in cloud architecture: managing a web application.

Web Application Use Case

A database-based web application is running and is internet-facing. The virtual machine (VM) is running a critical business application and would be considered a potential target. Although the methods of attack against web applications in the cloud are similar to those on premises, threat hunters must adjust their approach and adopt a new set of tools for detection and remediation.

The cloud management plane is an attack vector that threat hunters must evaluate. If attackers were to gain a foothold in a web application, could they leverage it to get further into the cloud infrastructure? Could they make changes, set up persistence and spin up a cryptocurrency mining rig that will run at great expense to the victim? The damage can be financially and legally impactful. The web application is running on an Amazon Elastic Compute Cloud (EC2),[1] a VM, that reaches out to an Amazon S3 bucket to retrieve configuration files every time the server starts up. This use case, illustrated in Figure 1, is simplified by design to help tell the threat hunting story. A properly architected web application would include additional protections.

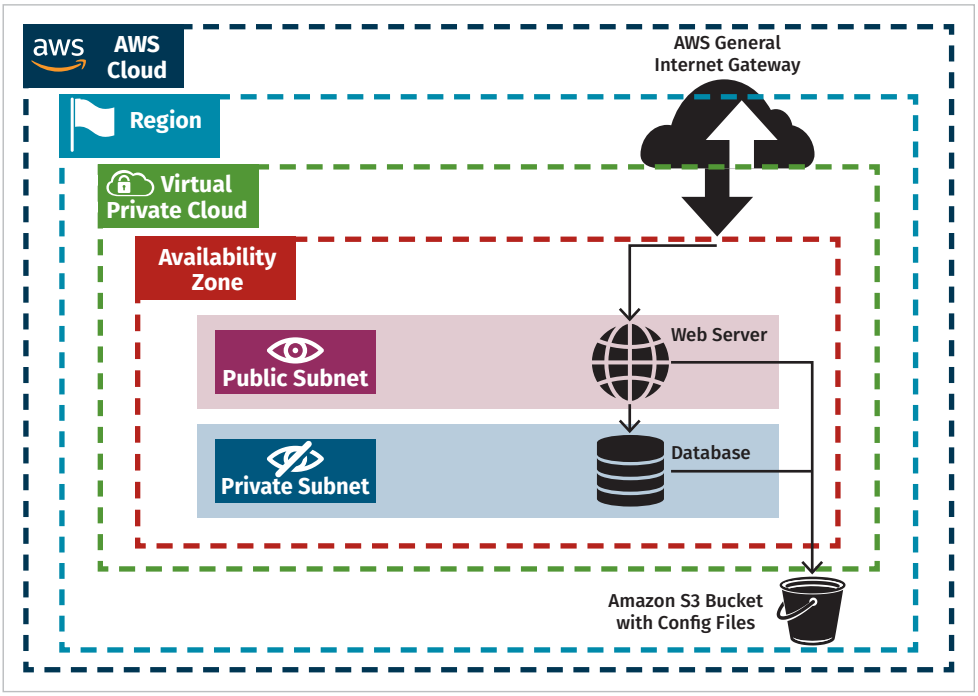


Figure 1. Web Application Use Case

How to Approach Threat Hunting

Threat hunting is more of an art than a science, in that its approach and implementation can differ substantially among various organizations and still be right. Every organization builds and operates its infrastructure in its own way; their teams have varied compositions of skill sets, talents and goals, and they face different threat risks.

[1] This paper mentions product names to provide real-life examples of how threat hunting tools can be used. The use of these examples is not an endorsement of any product.

Threat hunting is about approaching security from a different angle. For instance, the security operations center (SOC) has a collection of alerts from various security products, such as antivirus scans, email security solutions, vulnerability scans, firewall alerts, IDS/IPS, and login failures. If a scan shows that a production server is vulnerable with a critical alert, a SOC member creates a ticket for the server administration teams to plan for an update. The driver of that interaction is a security product alerting on a strong indicator. Thus a workload needs to be patched.

Threat hunting starts with the premise of, "Our main web application is facing the internet and may be the victim of a web attack. Let's see how we can determine that." Or maybe a weak indicator sparks suspicion: "Multiple failed SQL injection attacks in a row. The web server performance is slower. Let's look for potential intrusions." There are multiple scenarios in between that can all be considered threat hunting.

With a strong indicator from a security service, there is a process in place to remedy the situation. With threat hunting, the team is looking for anomalous behaviors without strong indicators. The outcome is likely unknown, the investigation is murky, and the process is research-intensive. It is essential to build a threat hunting process and environment to maximize the effectiveness of the team.

CIS Critical Controls Are Vital to Threat Hunting

The Center for Internet Security (CIS) identifies 20 essential security controls, the first six of which are basic controls. Table 1 lists these basics controls and describes their importance to creating an effective threat hunting program.

Table 1. CIS Critical Controls and Threat Hunting[2]

CIS Control	Description
Control 1: Inventory and Control of Hardware Assets Control 2: Inventory and Control of Software Assets	Threat hunters need to know and manage hardware and software assets, so they can identify which infrastructure services to evaluate and what software is approved.
Control 3: Continuous Vulnerability Management Control 5: Secure Configuration for Hardware and Software on Mobile Devices, Laptops, Workstations and Servers	By eliminating software vulnerabilities, threat hunters can save time and resources.
Control 4: Controlled Use of Administrative Privileges	Organizations should limit the use of admin privileges so threat hunters can better determine what is legitimate use.
Control 6: Maintenance, Monitoring and Analysis of Audit Logs	The core of threat hunting relies on proper managing, monitoring and analysis of logs.

Threat Hunting Loop

Building a threat hunting process from scratch takes time, resources and the ability to reach out to experts inside and outside the organization. The Threat Hunting Loop,[3] shown in Figure 2, describes the process for determining what threat to hunt for, evaluating it and then automating the further investigation.

The threat hunting process is all about deciding what potential threat activity to look for, using tools to analyze the available data and teasing out patterns that could indicate a likely event. Each of these steps of the loop is unique to your organization, its infrastructure, the data available to the team and the tools at its disposal.

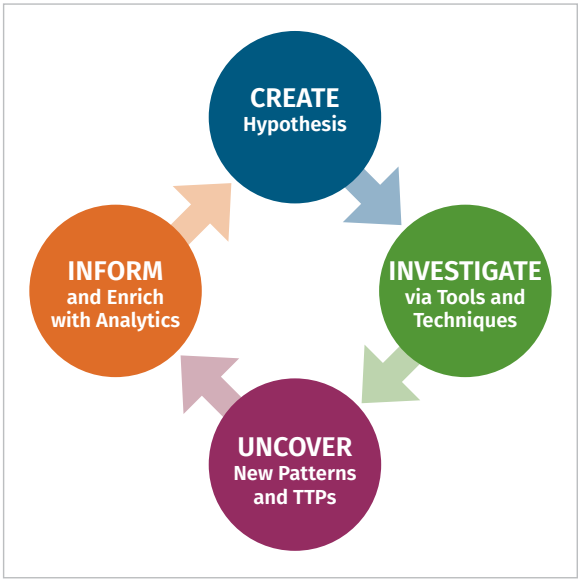


Figure 2. Threat Hunting Loop

[2] www.cisecurity.org/controls/cis-controls-list/

[3] www.threathunting.net/sqrrl-archive

Create Hypothesis

Step one is to create the hypothesis. Did the attacker gain a foothold in the production web application? Could credentials be accidentally embedded in the packaged software? Is there an unknown, CPU-intensive process running on an important server? The sheer scope of potential hypotheses could grind any team progress to a halt.

Identifying and prioritizing the most at-risk infrastructure components requires an understanding of which systems are most vulnerable and their values to the business.[4] By starting with a threat modeling process, an organization has an outline of priority systems that have a risk and are vulnerable to some set of attacks.

The threat hunting team needs to build a set of techniques to investigate and create a hypothesis of how those attacks would work and what artifacts are in the logs that need to be analyzed. Organizations with an offense-focused team, like a pen-test group or red team, have in-house experts who research and practice attacker techniques.

Others may need to rely on researching published materials on attack techniques to create new hypotheses. For example, the MITRE ATT&CK™ Framework is growing in popularity among researchers and security companies (see Figure 3). Although not cloud-specific, the ATT&CK Framework provides a detailed explanation of the hows and whys of specific attacker techniques.

At-risk infrastructure has one of four possible responses: attempt to mitigate the threat, eliminate the threat through infrastructure architecture, transfer the risk to a third party or just accept the risk.

MITRE Enterprise ATTACK™ Framework

Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Execution	Collection	Exfiltration	Command and Control
Image File Execution Options Injection			Forced Authentication	Network Share Discovery	AppleScript		Man in the Browser	Exfiltration Over Physical Medium	Multi-hop Proxy
Plist Modification			Hooking	System Time Discovery	Third-party Software		Browser Extensions		Domain Fronting
Valid Accounts			Password Filter DLL	Peripheral Device Discovery	Windows Remote Management		Video Capture	Exfiltration Over Command and Control Channel	Data Encoding
DLL Search Order Hijacking			LLMNR/NBT-NS Poisoning	Account Discovery	SSH Hijacking	LSASS Driver	Audio Capture		Remote File Copy
AppCert DLLs		Process Doppelgänging	Securityd Memory	File and Directory Discovery	Distributed Component Object Model	Dynamic Data Exchange	Automated Collection	Scheduled Transfer	Multi-Stage Channels
Hooking		Mshta	Private Keys	System Information Discovery	Pass the Ticket	Mshta	Clipboard Data	Data Encrypted	Web Service
Startup Items		Hidden Files and Directories	Keychain		Replication Through Removable Media	Local Job Scheduling	Email Collection	Automated Exfiltration	Standard Non-Application Layer Protocol
Launch Daemon		Launchctl	Input Prompt	Security Software Discovery	Windows Admin Shares	Trap	Screen Capture	Exfiltration Over Other Network Medium	Communication Through Removable Media
Dylib Hijacking		Space after Filename	Bash History		Remote Desktop Protocol	Source	Data Staged		Multilayer Encryption
Application Shimming		LC_MAIN Hijacking	Two-Factor Authentication Interception	System Network Connections Discovery	Pass the Hash	Launchctl	Input Capture	Exfiltration Over Alternative Protocol	Standard Application Layer Protocol
AppInit DLLs		HISTCONTROL	Account Manipulation		Exploitation of Vulnerability	Space after Filename	Data from Network Shared Drive	Data Transfer Size Limits	Commonly Used Port
Web Shell		Hidden Users	Replication Through Removable Media	System Owner/User Discovery	Shared Webroot	Execution through Module Load	Data from Local System	Data Compressed	Standard Cryptographic Protocol
Service Registry Permissions Weakness		Clear Command History	Input Capture	System Network Configuration Discovery	Logon Scripts	Regsvcs/Regasm	Data from Removable Media		Custom Cryptographic Protocol
Scheduled Task		Gatekeeper Bypass	Network Sniffing	Application Window Discovery	Remote Services	InstallUtil			Data Obfuscation
New Service		Hidden Window	Credential Dumping		Application Deployment Software	Regsvr32			Custom Command and Control Protocol
File System Permissions Weakness		Deobfuscate/Decode Files or Information	Brute Force	Network Service Scanning	Remote File Copy	Execution through API			Connection Proxy
Path Interception		Trusted Developer Utilities	Credentials in Files	Query Registry	Taint Shared Content	PowerShell			Uncommonly Used Port
Accessibility Features		Regsvcs/Regasm		Remote System Discovery		Rundll32			Multiband Communication
Port Monitors		Exploitation of Vulnerability		Permission Groups Discovery		Scripting			Fallback Channels
Screensaver		Extra Window Memory Injection		Process Discovery		Graphical User Interface			
LSASS Driver		Access Token Manipulation		System Service Discovery		Command-Line Interface			
Browser Extensions		Bypass User Account Control				Scheduled Task			
Local Job Scheduling		Process Injection				Windows Management Instrumentation			
Re-opened Applications		Component Object Model Hijacking				Trusted Developer Utilities			
Rc.common	SID-History Injection	InstallUtil				Service Execution			
Login Item	Sudo	Regsvr32							
LC_LOAD_DYLIB Addition	Setuid and Setgid	Code Signing							
Launch Agent		Modify Registry							
Hidden Files and Directories		Component Firmware							
.bash_profile and .bashrc		Redundant Access							
Trap									
Launchctl									

Figure 3. MITRE ATT&CK Framework[5]

Specifically, the technique of gaining initial access by exploiting public-facing apps is relevant to the web app use case. ATT&CK describes the purpose of the technique, the types of platforms, potential mitigations and references to online reports. The information provided on this technique does not give us enough details to start hunting,

[4] Learn more about the threat modeling process in "How to Protect a Modern Web Application in AWS," www.sans.org/reading-room/whitepapers/analyst/protect-modern-web-application-aws-38955, [Registration required.]

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

but it does point to the Open Web Application Security Project (OWASP) Top 10, which is more relevant to the use case. More detail is noted in Figure 4.

When identifying the potential attacks against a web application, one of the best sources is the OWASP Top 10. The OWASP Top 10 is a documented explanation of the top security threats to web applications, detailing the attacker techniques, examples and potential ways to mitigate.

The top threat in the OWASP Top 10 is an injection attack, or getting untrusted data sent to the interpreter and executed as part of a command or query. (See Figure 5.) In a SQL injection attack on a web server, the attacker provides unexpected values for the username or password to thwart the interpreter from retrieving the expected SQL values.

The Cloud Security Alliance (CSA) publishes a report on top threats[8] that focuses specifically on cloud services. The CSA also publishes an in-depth case study[9] that walks through how those threats are carried out. Rhino Security is a pen-test company, but it publishes blogs and free tooling for cloud and containerization threats.

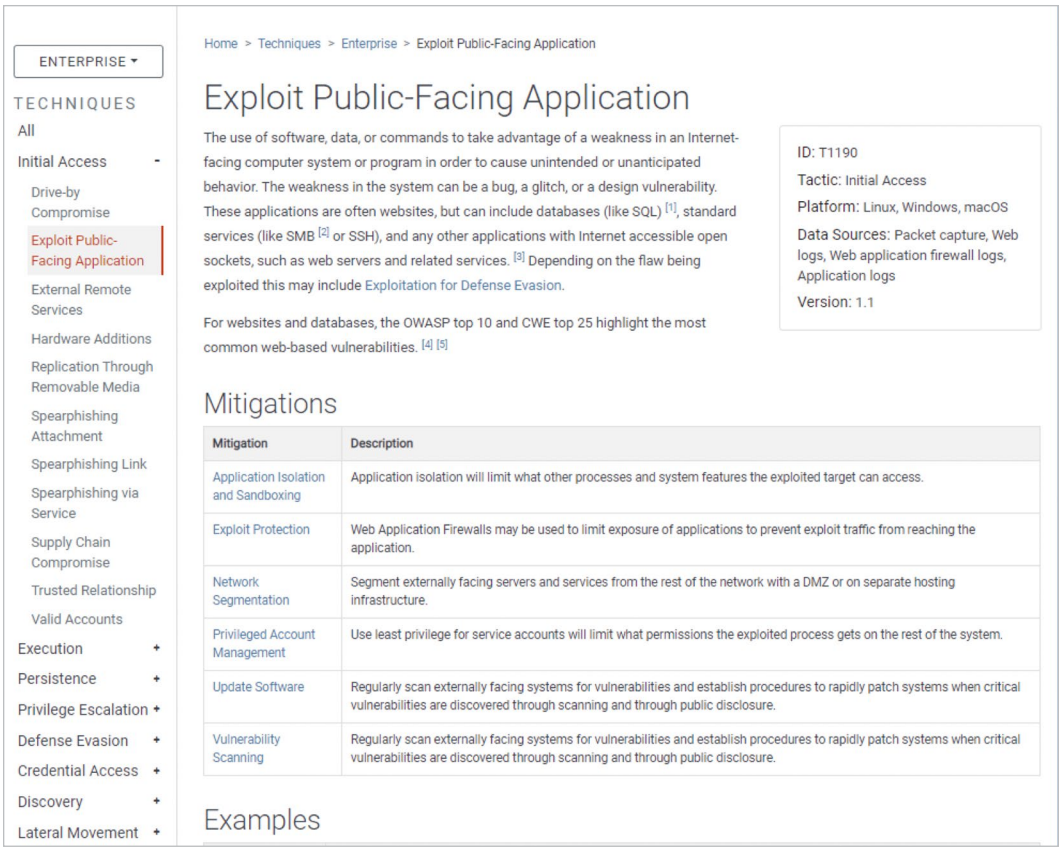


Figure 4. The Exploit Public-Facing Application Technique[6]

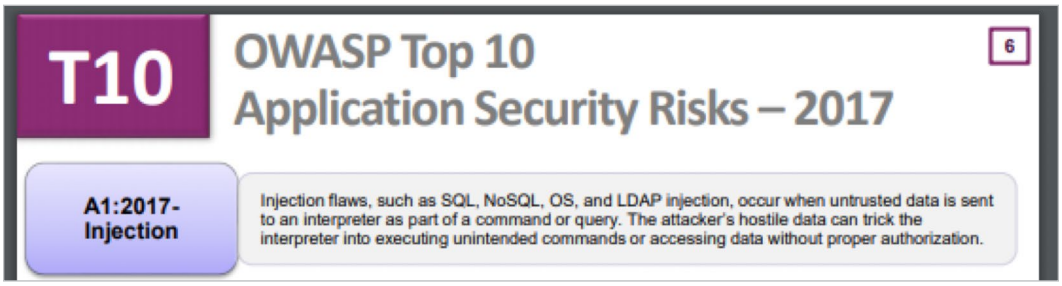


Figure 5. Number One Threat in the OWASP Top 10[7]

Other publications and researchers who track and describe attacker techniques include:

- Threat Post
- Threat Hunting Project
- AWS Security Bulletin
- (ISC)² Cloud Security Report
- Summit Route
- Toni de la Fuente's running list of AWS Security Tools

---

[6] "Exploit Public-Facing Application," https://attack.mitre.org/techniques/T1190/

[7] OWASP Top Ten Project, www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

[8] Cloud Security Alliance, Top Threats to Cloud Computing: Egregious Eleven, https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-egregious-eleven/

[9] Cloud Security Alliance, Top Threats to Cloud Computing: Deep Dive, https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-deep-dive/

Investigate Via Tools and Techniques

Threat hunters go beyond the automated alerts from security products, past the strong indicators and into the squishy unknown. To do this, data must be collected, understood, analyzed and viewed comprehensively. Threat hunters must also pivot through different types of logs and explore unstructured or partially structured data.

The first hurdle can be the infrastructure itself. If the organization has dozens of unique operating system configurations, manually managed deployment or shared remote management, then logs and operational data will be highly variant, allowing real attacks to blend in. Let's look at another use case.

Use Case: Gathering SSH Connections

Leveraging infrastructure as code, it is possible to deploy production systems without administrators SSH'ing, except in cases of troubleshooting. Teams can easily pull logs from any system and into Amazon CloudWatch. See Figure 6.

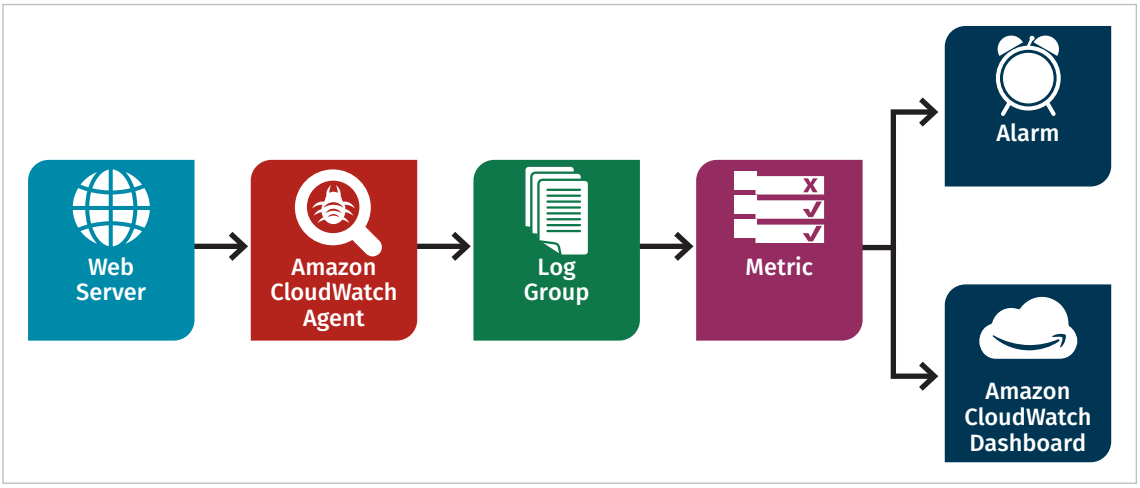


Figure 6. Overview of Amazon CloudWatch Log Collection

To use the Amazon CloudWatch agent to pull SSH connection logs from Amazon EC2s and into the Amazon CloudWatch logging service, follow these steps:

1. Install the Amazon CloudWatch agent on an EC2.
2. Configure the Amazon CloudWatch agent to send SSH connections to a specific log group.
3. Set up Amazon CloudWatch alarms to monitor for invalid user attempts and repeated SSH disconnects.

The Ever-Changing Cloud Infrastructure

Cloud service elasticity can make it difficult to directly interrogate systems when the environment is continually growing and shrinking throughout a day. For example, let's say the web application is attacked at 10 p.m. with a SQL injection attack that triggers logs from the web application firewall (WAF). The next day at 9 a.m., the threat hunting team investigates to determine if the attack was successful. Unfortunately, the VM has already been terminated by the cloud autoscaling engine. The threat hunting team needs to decide what data to collect from the elastic system, whether that data is readily available or needs to be pulled or pushed by additional systems, and how long to keep the data before aging it off. The threat hunter needs to account for the risk of those systems, the amount of data that might need to be stored and how quickly a team will evaluate the data. The following demonstrates an example.

Use Case: Post-Exploitation Detection

In a cloud environment of automation, once attackers gain access to the web application VM, they will want to use the MITRE ATT&CK tactic called Discover to find other services of interest, such as an accessible Amazon S3 bucket with the command **`ListBuckets`**. The web application we built has access to Amazon S3 buckets for configuration, but the IAM role does not allow listing of buckets. Automated systems likely already know the resources they need to interact with, so listing potential names is unnecessary. From the Amazon EC2 instance, listing buckets results in an error, as shown in Figure 7.

```
[ec2-user@ip-10-0-25-212 ~]$
[ec2-user@ip-10-0-25-212 ~]$ aws s3api list-buckets

An error occurred (AccessDenied) when calling the ListBuckets operation: Access Denied
[ec2-user@ip-10-0-25-212 ~]$
```

Figure 7. A **`ListBuckets`** *Error*

AWS CloudTrail gathers and allows an analysis of Amazon Web Services (AWS) API requests. AWS CloudTrail, using the Amazon EC2 ID as the username, looks at the **`ListBuckets`** as an indicator. There is an **`AccessDenied`** error code, as shown in Figure 8.

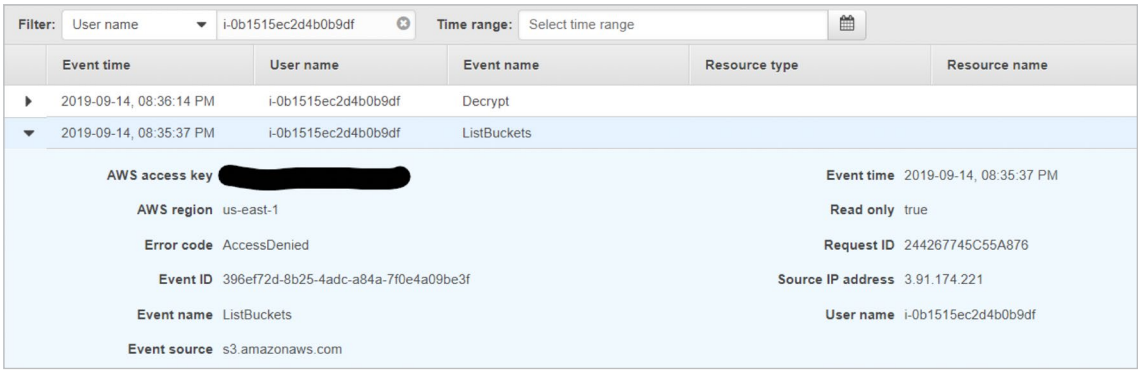


Figure 8. **`AccessDenied`** *Error Code*

Another option is to use the AWS Command Line Interface (CLI) to look for all commands from the Amazon EC2 in question:

```
aws cloudtrail lookup-events --lookup-attributes
AttributeKey=Username, AttributeValue=i-0b1515ec2d4b0b9df --query
'Events[].{username:Username,time:EventTime,event:EventName,eventid:
EventId,resource:(Resources[0].ResourceName)}' --output table --
region us-east-1
```

Figure 9 shows sample results of AWS CloudTrail **`lookup-events`**.

```
-------------------------------------------------------------------------------------------------
|                                          LookupEvents                                         |
+-------------+--------------------------------------+-----------+--------------+---------------------+
|    event    |               eventid                | resource  |     time     |       username      |
+-------------+--------------------------------------+-----------+--------------+---------------------+
|  Decrypt    |  27bce37b-7db0-4567-8367-ee4f4f02ef39 |  None     |  1568507774.0|  i-0b1515ec2d4b0b9df|
|  ListBuckets|  396ef72d-8b25-4adc-a84a-7f0e4a09be3f |  None     |  1568507737.0|  i-0b1515ec2d4b0b9df|
|  ListBuckets|  2579d4a9-e0b1-4cf0-b7a8-7f6edcab28ed |  None     |  1568507736.0|  i-0b1515ec2d4b0b9df|
|  ListBuckets|  aa9628dc-de9c-4818-8a40-dc22bc9dc846 |  None     |  1568507736.0|  i-0b1515ec2d4b0b9df|
|  ListBuckets|  0b3c1151-7a61-4651-b91d-9f22a973cce5 |  None     |  1568507735.0|  i-0b1515ec2d4b0b9df|
|  ListBuckets|  5a1384c4-b77d-46e0-8c6d-4486a15ddb37 |  None     |  1568507365.0|  i-0b1515ec2d4b0b9df|
|  ListBuckets|  8f8158ff-b837-4bba-a413-43ebcc65107b |  None     |  1568507363.0|  i-0b1515ec2d4b0b9df|
|  ListBuckets|  13485b09-a4e8-4e62-aec1-c4d5982e86b3 |  None     |  1568507362.0|  i-0b1515ec2d4b0b9df|
|  ListBuckets|  1ef3785c-c2cb-4ee0-bb60-807c8e00b9b8 |  None     |  1568507361.0|  i-0b1515ec2d4b0b9df|
|  ListBuckets|  373507ce-1331-4682-81b7-313a260bcd7e |  None     |  1568507309.0|  i-0b1515ec2d4b0b9df|
+-------------+--------------------------------------+-----------+--------------+---------------------+
```

Figure 9. Table Output of AWS CloudTrail **`lookup-events`** *Command*

Each event has a unique event ID. Figure 10 shows the details for a specific event ID from the table shown in Figure 9. Here, we use a Linux application, JQ, to carve up JSON on the command line.

```
cybergoof> aws cloudtrail lookup-events --lookup-attribute  AttributeKey=EventI
d,AttributeValue=396ef72d-8b25-4adc-a84a-7f0e4a09be3f --query "Events[0].CloudT
railEvent" --region us-east-1 --output text | jq -r '.'
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
```

Figure 10. JSON Output of AWS CloudTrail **`lookup-events`**

This command shows the details of this particular AWS CloudTrail Event. JQ is an excellent tool for filtering, carving and formatting the JSON data in logs.

Uncover New Patterns and Apply Learned Lessons

Gathering data, running analytics and identifying the anomalies give the threat hunter unique insights into evaluating attack techniques and analyzing infrastructure systems. The team should become part of the threat modeling processes, helping the architecture and operations teams identify the cloud infrastructure that needs to be secured and evaluated. Changes such as improved monitoring, reduced chaotic deployments and better segmentation of infrastructure can all make threat hunting easier without losing operational capabilities.

Once threat hunters understand the challenges, they can start gathering detailed knowledge of potential threats, and the architecture and infrastructure management teams can support the threat hunters. It is time to begin collecting and analyzing the data needed to discover the attackers.

Inform with Data and Analytics

It is critical to get the right data into the right place for analysis. The data itself might need to be evaluated, enriched and prepared for analysis using scripts, tools or built-in cloud services.

Gathering the Data

The threat hunting team has to strike the right balance of how much data to capture. Requiring all the data from all the things increases costs, adds to the overhead of managing the data and increases the time and effort to sift through and analyze the enormous amounts of data. On the other hand, not having enough data will keep the threat hunters in the dark. First, identify any logs that are already being collected or are easy to obtain organically. AWS makes it easy to collect VPC logs showing data connections in and out of the VPC, API calls with AWS CloudTrail and Amazon S3 access logs, among others.

Then, using the attacker techniques, the team will focus on identifying the gaps in information and how to retrieve it. Most missing data is likely from applications or the host environment itself. Let's revisit the web application use case.

Web Application Use Case

For the web application use case, the VM itself has a wealth of information that could be of interest. Mainstream web servers generate standard logs that are stored on the VM. They also can be customized to generate more or fewer logs, or with changes to the format or location, and potentially compressed for transfer. Connection logs, for example, contain every HTTP request to the web server. Regularly managed web applications have a lot of the same connections. However, in a path traversal attack,[10] the path could contain unique path calls that are attempts to get access to files on the web server.

```
[/var/log/nginx]
datetime_format = %b %d %H:%M:%S
file = /var/log/nginx/access.log
buffer_duration = 5000
log_stream_name = {instance_id}
initial_position = start_of_file
log_group_name = /var/log/nginx
```

Figure 11. Amazon CloudWatch Logs Configuration File

After installing the Amazon CloudWatch agent, configure the Amazon CloudWatch configuration file to pull the Nginx access log **`/var/log/nginx/access.log`**. See Figure 11.

The Nginx connection logs are now stored in the **`/var/log/nginx`** log group, accessible from Amazon CloudWatch Logs. See Figure 12.

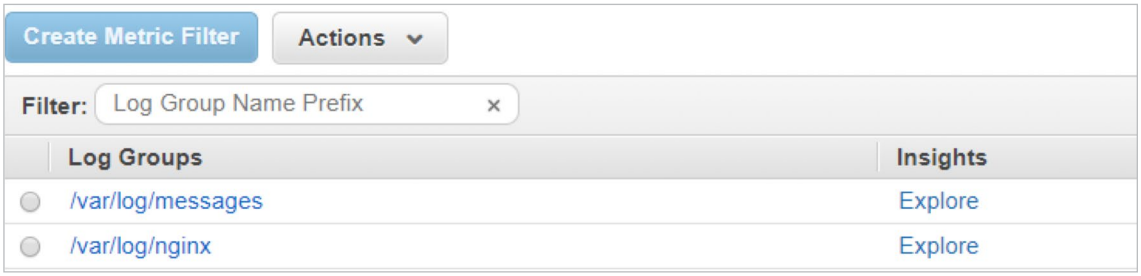


Figure 12. Nginx Connection Logs

Opening up the log group, it's possible to search for a string, as shown in Figure 13.

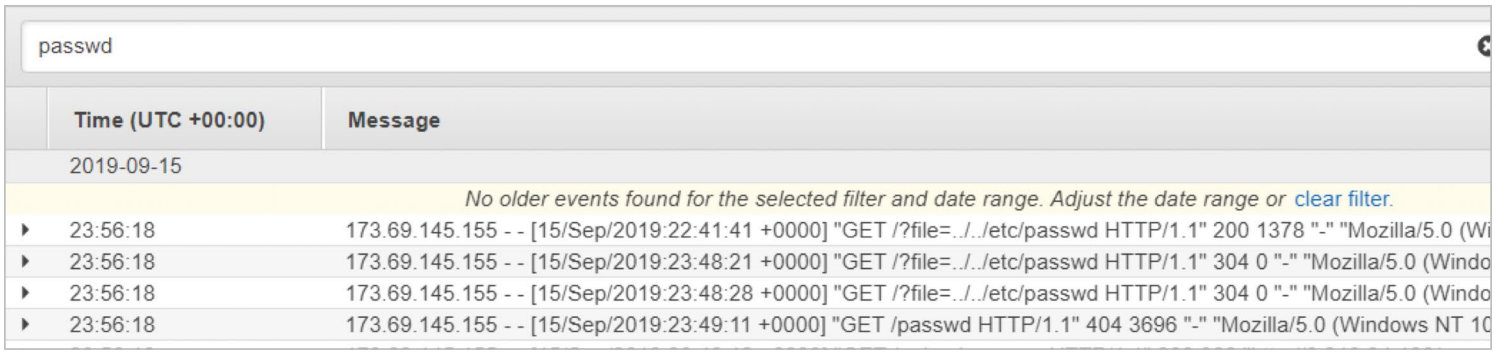


Figure 13. Quick Search for **`passwd`**

This is an easy search. AWS provides an advanced query service called Amazon CloudWatch Logs Insights. Using a custom query language, we can search across all hosts for a regex of **`passwd`**, **`etc`** or **`../`** as shown in Figure 14. Note that **`/`** is a special character in regular expression (regex), so it has to be escaped with **`\`**.

```
fields @timestamp, @message
| sort @timestamp desc
| limit 20
| filter @message like /passwd|etc|..\//
```

Figure 14. Query Amazon CloudWatch Logs Insights

[10] www.owasp.org/index.php/Path_Traversal

Figure 15 shows the results of the query.

Once the data is gathered, the data retention life cycle rule is applied and data is accessible, it's time to figure out how to make the data more useful to the threat hunters by enriching the data.

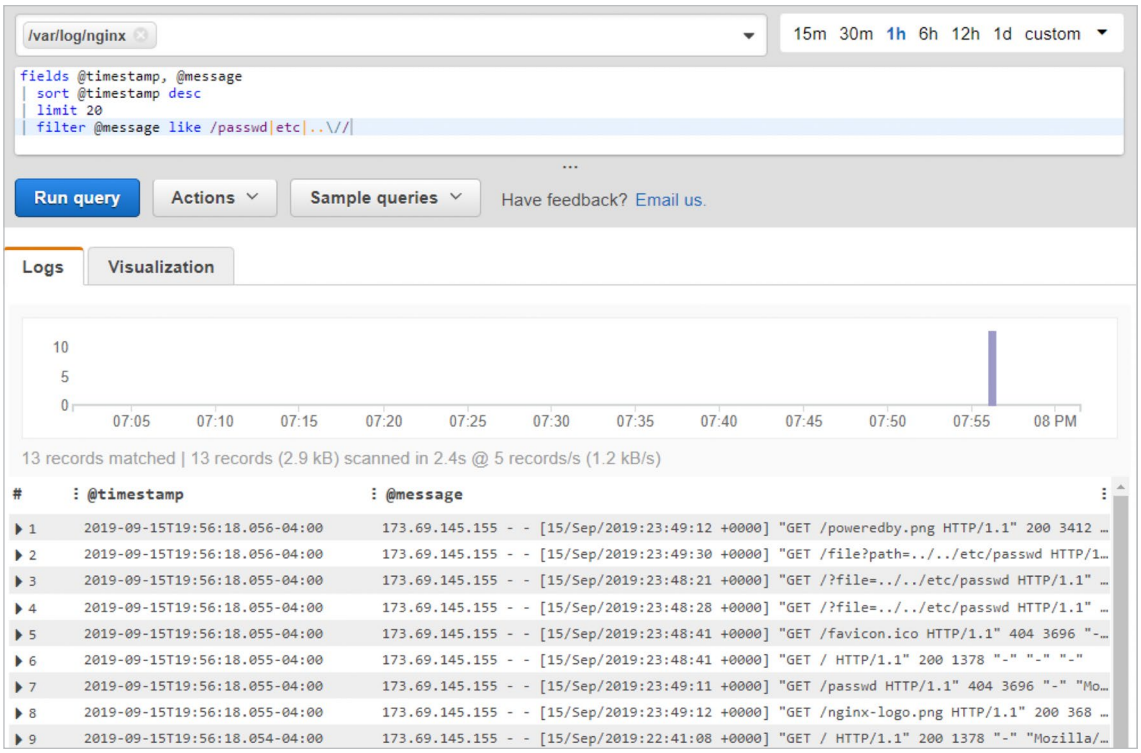


Figure 15. Query Results

Enriching the Data

When threat hunting, the data needs to tell a complex and complete story with multiple characters, settings and subplots. If a single log could tell the story, then a security product would quickly alert the SOC. Threat hunters are looking for more subtle anomalies in the data that look unique mainly because of the way an infrastructure is architected and operated. An attachment in the email is easily scanned and compared to a known list of malware. However, it's harder to identify a nefarious remote desktop connection compared to a legitimate one. One easy way to bring data to life is to automatically evaluate the data and tag it, add metadata or enhance the data itself.

Separate Security Account

It is good to gather and protect any logs from accidental or purposeful deletion. One recommendation is to use AWS Organizations to create a separate security organization (org) and to automatically move logs from the production org to the security org, where it can be protected and available to only the security or designated teams.

Web Application Use Case

There are several ways to automate the analysis and tagging or enriching the data. For logs collected by Amazon CloudWatch, such as Nginx connection logs, leveraging the alarms, metrics and dashboards works well. An Amazon CloudWatch Metric Filter will search for some specific patterns and create a metric count when that pattern shows up in the logs. An Amazon CloudWatch metric can generate an alarm, which can send an email or notify an AWS Lambda function. The AWS Lambda function can take action, such as copying the concerning data over to an Amazon S3 bucket for further analysis.

In the Amazon EC2 Role use case, the victim EC2 can perform S3 bucket reads. Let's say there are 50 EC2 instances in the account; that would be too much data to analyze. However, if the EC2 reads a different S3 bucket than it has ever read before, that is a new activity. You should tag those reads.

Analyzing the Data

Once the data has been gathered, enriched and tagged, the threat hunting team starts evaluating the data to identify anomalous behaviors against the hypothetical attack techniques. The threat hunting team must be able to evaluate anomalies and quickly determine if they warrant an investigation or not, so the data must be easy to search,

correlate and report. Various scripting tools and analytic platforms can provide threat hunters with raw log data to sift through. Comprehensive analytic platforms can also be utilized to help speed up analysis, and provide reporting services for sharing and collaboration among teams.

The next sections dive into options for analytic tools to bring into the environment to take threat hunting to the next level.

Tools for Analysis

Threat hunters can bring a wide range of tools to bear to analyze complex datasets from multiple sources, from scripts parsing raw data, to a full SIEM system that provides ad hoc and complex searching, reporting and investigations. The decision is usually about setup complexity, cost and the need to scale as the team grows. AWS provides several services that can be used and chained together to scripts and analytics.

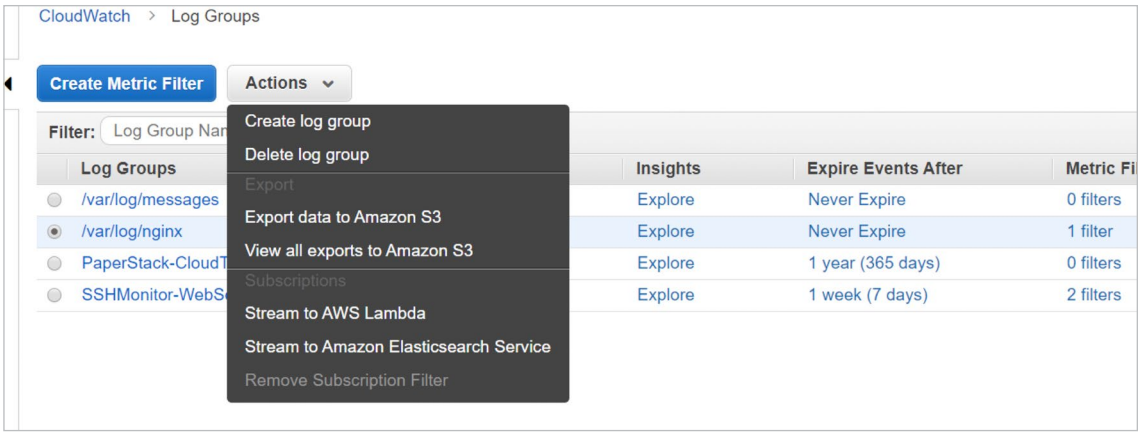


Figure 16. Exporting Amazon CloudWatch Logs

Analyzing Logs Directly

Amazon CloudWatch is the core service for monitoring an AWS environment, because it is easy to get up and running and providing basic metrics, alarming and dashboards. As was previously discussed, Amazon CloudWatch and AWS CloudTrail can be used together to interact directly with collected data. AWS offers methods of exporting Amazon CloudWatch logs, collected from custom applications to Amazon S3, AWS Lambda or Amazon Elasticsearch Service (see Figure 16).

AWS provides another service called Amazon Athena, which runs SQL queries against data in an Amazon S3 bucket (see Figure 17). Customers

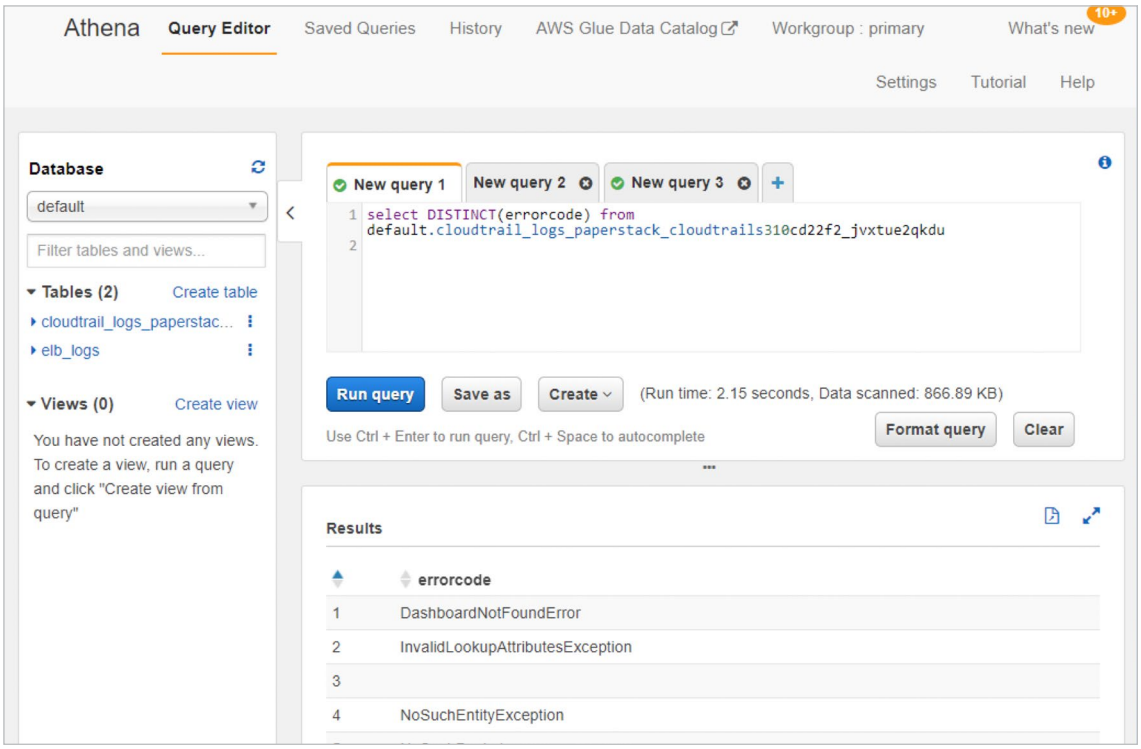


Figure 17. Amazon Athena Dashboard

build virtual tables that organize and format the underlining log data inside the bucket objects. It takes time to ensure that data is formatted and managed.

Amazon GuardDuty is a managed service that is evaluating a growing number of findings that detect adversary behaviors and alerting the customer. Amazon GuardDuty evaluates potential behaviors by analyzing Amazon VPC Flow Logs. A similar real-time VPC flow logs analysis engine can be created using AWS Lambda, Amazon Kinesis, Amazon S3, Amazon Athena and Amazon QuickSight.

SIEMs in the Cloud

As a threat hunting team starts to build a corpus of analytics that it wants to run repeatedly, or as its investigating, monitoring and reporting needs become more comprehensive, a full SIEM is likely of interest. Several cloud-specific services, as well as traditional on-premises SIEMs, work with cloud infrastructure.

The threat hunting team should focus on developing and managing a tactical SIEM, which could be different from the SIEM a SOC might use. The tactical SIEM will likely have unstructured data, a shorter retention policy than the SOC's SIEM, and the ability to easily determine what the infrastructure looked like in the recent past. In the cloud, good data management strategy should be implemented to be cost-effective, with pay-per-usage pricing. Generally, free or open source solutions tend to take more time and expertise to set up and maintain, but they are more customizable and cost little or nothing. Commercial solutions may cost more, but may come with better support, easy access to purpose-built connectors and more reporting options.

As the threat hunting team's analytics become more sophisticated, it may begin developing a set of repeatable analytics, enrichments or data gathering steps. If it's repeatable and articulate, it can be automated.

Elasticsearch, a favorite of the open source community, boasts a significant user base and supports plug-ins for data importing, translating and easy displaying with the Kibana application. AWS provides a managed Amazon Elasticsearch Service to make it easy to set up and run the search engine without having to do all the management heavy lifting. The company behind Elasticsearch, Elastic, has released a new app called the Elastic SIEM that is more focused on the security operations. Other products, such as ones from Sumo Logic and Splunk, also integrate directly with AWS and provide even richer and more full-featured analytic platforms.

After the tactical SIEM is stood up; the data is gathered, translated and enriched; and mechanisms for analytics and reporting are in place, the threat hunting team will start to discover repeated steps, analytics or actions. An emerging service that integrates with the SIEM, called Security Orchestration, Automation and Response (SOAR), can be helpful there.

Soaring with SOAR

Threat hunting is all about proactive analysis of data to detect the anomalous behavior that is undetectable by the security products. As the threat hunting team's analytics become more sophisticated, it may begin developing a set of repeatable analytics, enrichments or data gathering steps. If it's repeatable and articulate, it can be automated. A SOAR leverages the data storage and enrichment of the SIEM, understands basic rules of infrastructure integration and allows the easy buildout of playbooks to automate a course of action.

In the web application use case, if there are several failed SQL injection attempts, the final attempt could signify the last failure before success. The process of information from that host at that time would be of interest. A SOAR could be used to identify that ultimate SQL injection failure, tag it and then also tag the process log information from that time. The next step in the playbook could be to move those logs into a separate Amazon S3 bucket for more accessible analysis. The process logs by themselves could then be enriched by validating with a malware signature API to identify whether the process is known good or not. Gathering potential logs to analyze and automating the enriching processes when necessary could save threat hunters tedious and repetitive work. It could also help provide quicker triage. The SIEM with a SOAR could significantly improve speed to analysis.

Taking the playbook a step further, it's possible to use data pushed to the SIEM and SOAR, such as the SQL injection detection logs from the WAF, and initiate an action. Rather than always pull the process list on an hourly basis, the SIEM could execute host-based tools, such as OSQuery, to reach out to the suspect web server and pull the process list in near real time. This automated response action allows the team to limit what passive data has to be managed, and makes it easier to correlate the process logs returned with the suspicious SQL injection attacks.

In the Amazon EC2 use case, the SIEM/SOAR could review the READs from an EC2 to an Amazon S3 bucket and detect a first-time READ to an S3 bucket. The SOAR playbook executes a host agent such as OSQuery or uses AWS services such as Amazon Inspector and AWS Systems Manager to interact directly with that EC2 to pull fresh process information and kick off a scan with Amazon Inspector. It then gathers all these reports and provides them in a single artifact bucket for the security analysts, creating a high-priority message in the corporate chat system or sending out SMS alerts to on-call personnel.

Some of the more sophisticated SOARs, such as Palo Alto's Demisto and Splunk's Phantom, also allow for the detection of cascading anomaly triggers that can perform automated remediations—taking our use cases together to build a sophisticated SOAR playbook.

SOAR Playbook Use Case

The attacker performs several SQL injection attacks against a particular EC2. The SOAR kicks off a process listing and tags all logs from that EC2 with a unique identifier. One of those logs with the unique identifier specifies a failed Amazon S3 bucket listing attempt. In an automated system, the bucket is known, and a listing is unlikely to be normal. The SOAR identifies that this failed bucket listing happened on an EC2 that is being triaged. Because the organization is using auto-scaling, the SOAR notifies the auto-scaling system to deregister the EC2 (i.e., pull that EC2 out of service but keep it running). The SOAR playbook waits for the deregistering to finish, then removes all security groups except triage, and the triage group effectively isolates the EC2 from all other systems. The SOAR then performs a memory dump of the EC2, takes a snapshot and stops the EC2. All the data is gathered up and prepared in an Amazon S3 bucket for the security team when it is ready to investigate.

Conclusion

We are in the early days of threat hunting, specifically in cloud environments. Organizations are moving away from traditional server-based infrastructure into serverless, event-driven architectures that rely on native cloud services. Threat hunters will adapt their processes, tools and techniques to identify and neutralize the threats in this new infrastructure landscape.

Threat hunting is critical to finding the advanced attacker techniques that have escaped the detection of deployed security products. The threat hunting process requires constant learning about attacker techniques and your organization's attack surface. Proper strategy ensures the right data is collected, enriched and available to the tools the threat hunting team uses to tease out suspicious anomalies from the vast and ever-changing infrastructure. Your threat hunting process is always growing and adapting to new learnings, increasing experience and the changing threat landscape.

About the Author

Shaun McCullough is a SANS instructor for the SEC545: Cloud Security Architecture and Operations class and gives back to his profession by mentoring and supporting the next generation of cyber professionals. With 25 years of experience as a software engineer, he has been focusing on information security for the past 15 years. Shaun is a consultant with H&A Security Solutions, focusing on secure cloud operations, building DevSecOps pipelines and automating security controls in the cloud. He also served as technical director of red and blue team operations, researched advanced host analytics, and ran threat intelligence on open source platforms in his work with the U.S. Department of Defense.

Sponsor

SANS would like to thank this paper's sponsor:

aws marketplace

SANS

Enabling a Threat Hunting Capability in AWS

Learn how to conduct effective threat hunting in your Amazon Web Services (AWS) environment.

aws marketplace

Enhance your threat hunting capabilities in AWS with third-party solutions.

To begin threat hunting, an organization needs to create a solid foundation and capabilities to support that journey. Security Information and Event Management (SIEM) and Security Orchestration, Automation, and Response (SOAR) solutions help you to focus on threat hunting by increasing security efficiency. Security experts no longer need to spend 90% of their time on tedious and repetitive tasks. Time and resources can be gained back through automation and other activities and tools.

One way to do this is to generate consistent workflows with guardrails to create efficiencies and reduce human error. You can also use unified collaboration and case management to reduce the number of systems an analyst must utilize by providing a single source of truth. Real-time analysis of threat intelligence can lead to automation of both future analyses and remediation tasks. And by setting in place brokered access, you can reduce time and risk imposed by escalation bottlenecks. This alone can create a dramatic reduction in time to resolution.

Sumo Logic's Cloud-Native Machine Data Analytics Service is a SIEM solution that you can use in conjunction with SOAR products, such as Palo Alto Networks' Demisto. This combination allows you to automate the process of evaluating and enriching complex data sets.

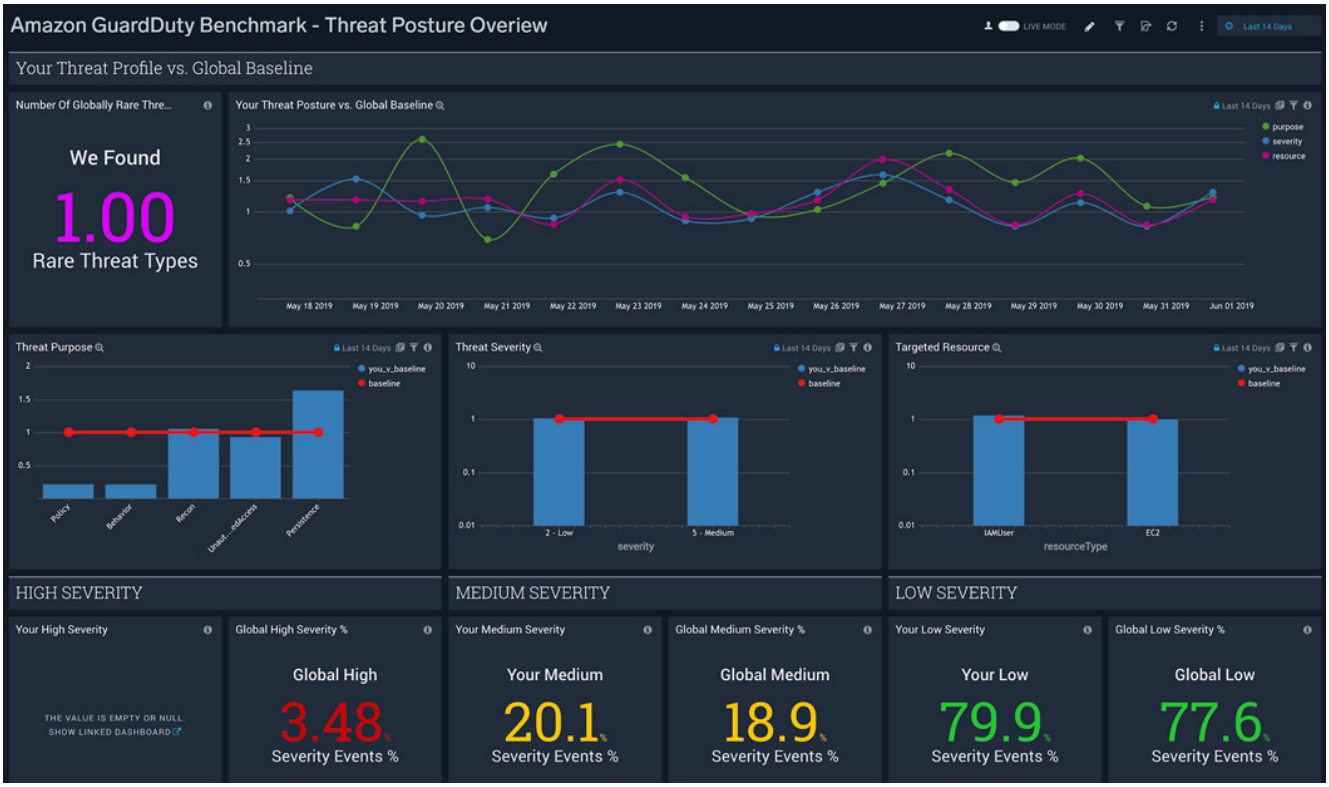
How AWS customers are using Sumo Logic to detect anomalous threats in their cloud environment

Sumo Logic is a cloud-native analytics platform that reduces the time necessary to investigate security and operational issues. Some of the ways customers are leveraging Sumo Logic to enhance their threat hunting strategies include:

- **Broad visibility across your AWS environment:** Sumo Logic has an architecture that processes more than 100 petabytes of data and handles 20+ million queries daily. It is an elastic solution that scales irrespective of data volume or number of users. This means it can handle a huge variety of formats, whether structured, unstructured, or semi-structured. This also means expansive visibility across your entire AWS environment.

- **Increased analyst productivity:** Sumo Logic pulls log files from a variety of AWS services, including AWS CloudTrail and Amazon VPC Flow Logs, and centralized metrics from Amazon CloudWatch to provide continuous machine learning-based intelligence. It analyzes logs for potential threats and indicators of compromise (IoCs) through a Threat Intelligence database that can be correlated with log data through queries. These activities can help increase analyst productivity by creating more signal and less noise.

- **Strong integrations with AWS:** Sumo Logic offers more than 150 applications and integrations that make it easy to aggregate data across your stack and down your pipeline. These tools include out-of-the-box pre-built analytics and dashboards for AWS services. One integration in particular that supports threat hunting for AWS customers is Sumo Logic's Global Intelligent Service (GIS) for Amazon GuardDuty. It provides customers with a baseline of what's normal, what's expected, and ways to dig deeper into the long tail of rare security events.



aws marketplace

Why use AWS Marketplace?

AWS Marketplace simplifies software licensing and procurement by offering thousands of software listings from popular categories like Security, Networking, Storage, Business Intelligence, Machine Learning, Database, and DevOps. Organizations can leverage offerings from independent security software vendors in AWS Marketplace to secure applications, data, storage, networking, and more on AWS, and enable operational intelligence across their entire environment.

Customers can use 1-Click deployment to quickly launch pre-configured software and choose software solutions in both Amazon Machine Image (AMI) formats and SaaS subscriptions, with software entitlement options such as hourly, monthly, annual, and multi-year.

AWS Marketplace is supported by a global team of security practitioners, solutions architects, product specialists, and other experts to help security teams connect with the software and resources needed to prioritize security operations in AWS.

How to get started with threat hunting solutions in AWS Marketplace

Security teams are using AWS native services and seller solutions in AWS Marketplace to help build automated, innovative, and secure solutions to address relevant use cases and further harden their cloud security posture. The following solution can help you get started:

sumo logic

Sumo Logic Cloud-Native Machine Data Analytics Service (Annually)
Continuous intelligence across your entire application lifecycle and stack