aws marketplace

SANS

Adopting Endpoint Detection and Response (EDR) in AWS Investigations

Learn how EDRs can enhance security in your Amazon Web Services (AWS) environment.

AWS Marketplace Introduction

Endpoint detection and response (EDR) is a key component of modern security. By integrating EDR capabilities, you can achieve more informed investigations and gain actionable details for remediation. In this whitepaper, SANS analyst and instructor, Justin Henderson explores how to leverage EDR within AWS to achieve a higher standard of security while simplifying what are normally complex tasks.

Building on Henderson's perspective, AWS Marketplace will share how you can specifically apply this strategy to your AWS environment. They will provide an introduction to relevant AWS services that can enhance your endpoint security. Finally, AWS Marketplace seller solutions will be featured as available options for supporting your EDR goals in AWS.

The featured solutions for this use case can be accessed in AWS Marketplace:

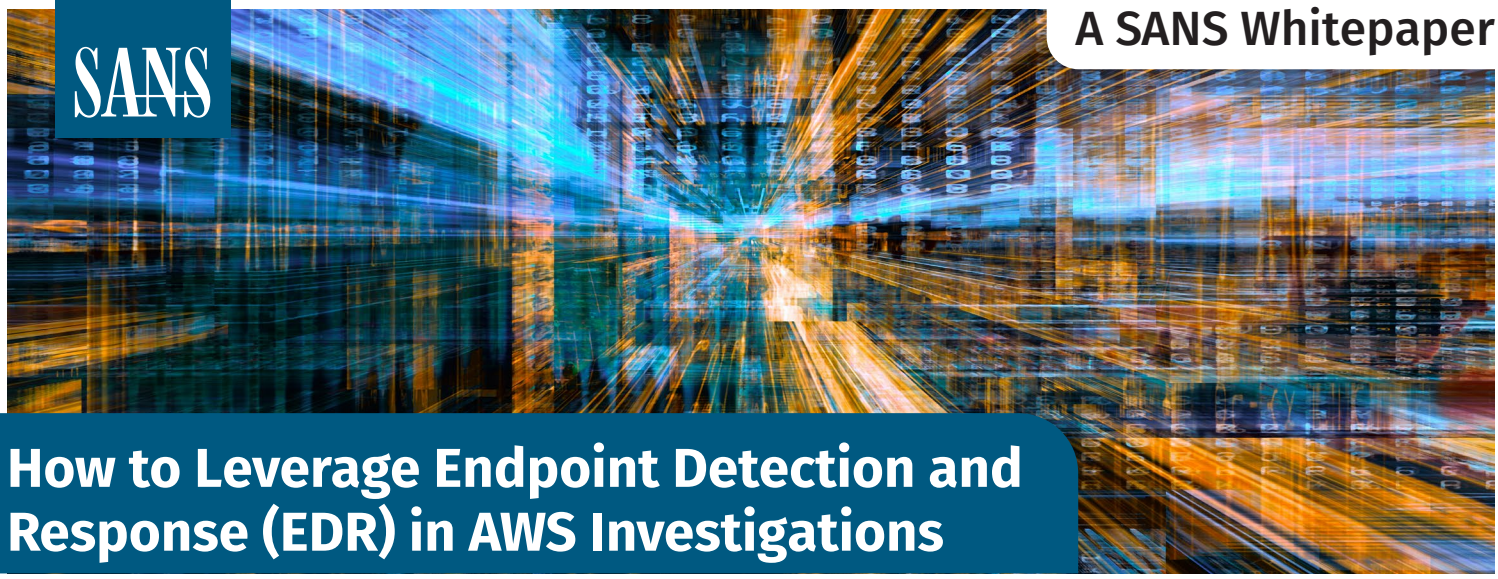
CrowdStrike Falcon Endpoint Protection Enterprise

Netskope Public Cloud Security

Sumo Logic Cloud-Native Machine Data Analytics Service (Annually)

SANS

A SANS Whitepaper



How to Leverage Endpoint Detection and Response (EDR) in AWS Investigations

Written by **Justin Henderson**

February 2020

Sponsored by:

AWS Marketplace

Introduction

The security challenges organizations face are often a direct result of evolving technologies such as virtual machines, containers, storage and even serverless code. Technology is not static. It changes dynamically via new developments such as infrastructure as code (IaC) and auto-scaling capabilities found at multiple layers of service. The result of this technological evolution is complexity in cloud environments. To secure such environments, you have to know and understand them.

Effective security teams implement appropriate technologies to mitigate potential weaknesses—for example, EC2 instances configured in a way that allows fileless malware such as the PowerShell **`Invoke-Mimikatz`** to steal credentials, or unsecured containers that an attacker can inject a PHP or .NET web shell into in order to access files and databases in Amazon S3 buckets, MySQL or an Amazon Relational Database Service (RDS). To enable more effective approaches to ensuring security, this paper illustrates how to leverage endpoint detection and response (EDR) in Amazon Web Services (AWS) to achieve a higher standard of security while simplifying management overhead. The goal is to ease the burden of cloud security via EDR technologies.

Acquiring Cloud Visibility

The first step in securing an AWS environment is not unique: Security teams need to understand what assets they have. After all, you cannot protect what you do not know exists. Traditionally this is a three-step process, as defined in Table 1.

Table 1. Asset Identification Process

Step	Definition	Example
Network scanning	A process to identify your assets and where they exist	Performing a port scan of Amazon EC2 instances
Service enumeration	A process to identify assets by querying a management service	Asking Kubernetes or Docker what containers exist
Agent installation	A process to push a security agent to an asset	Installing or using a log agent like `Syslog-NG`

But when it comes to cloud visibility, that traditional approach could leave gaps in coverage because of the way customers configure their environment. Good security practices involve customers locking down their assets, but a network scan would not identify all EC2 instances, because of customer configuration of Amazon security policies, network firewalls, and potentially endpoint controls or configurations. The lockdown of EC2 assets improves security, but it also makes 100% asset discovery difficult or impossible. Yes, an agent can easily be deployed to EC2 instances. However, because of an inability to see all instances and understand the underlying operating system, it is not possible to be aware of all assets in order to push agents to them. A more comprehensive approach is needed.

In addition, containers, Amazon S3 storage and serverless code execution are not traditional computer technologies. For them, deploying an agent is not necessarily an option, and even if it is for your organization, we recommend against this practice. Consider an Amazon EKS container running Nginx. This container is designed to run Nginx and nothing else, as indicated by the following code:

```
      PID  %CPU  %MEM  VSZ    RSS   TTY    STAT  START  TIME  COMMAND
root  1    0.0   0.1   10632  5488  pts/0  Ss+   18:49  0:00  nginx: master process nginx -g daemon off
nginx 6    0.0   0.0   11104  2664  pts/0  S+    18:49  0:00  nginx: worker process
```

Can you deploy an agent within a container? Yes. Should you? No, because deploying agents to a container introduces software dependencies, increases computational resources and adds management overhead.

However, without the ability to discover and protect containers, you are exposing yourself to a lot of risks. The same holds true for other services such as Amazon S3 storage. You cannot directly deploy an agent to an S3 bucket, but it still needs to be monitored for unauthorized access.

To achieve a holistic view of your AWS environment, consider adopting a modern methodology that integrates with AWS. AWS supports multiple EDR vendors that utilize Amazon APIs to move past the “everything requires an agent” approach. The steps outlined in Figure 1 on the next page show a more modern process.

Adopting a unified and holistic view of assets brings a simplified understanding of your environment. You can easily deploy these solutions, requiring you only to choose and subscribe to the vendor in AWS Marketplace. For example, subscribing to CrowdStrike's EDR[1] provides the capability to probe Amazon EC2, Amazon Elastic Container Service (ECS), and Amazon Elastic Kubernetes Service (EKS) to provide EDR, next-generation antivirus, threat intelligence and more.

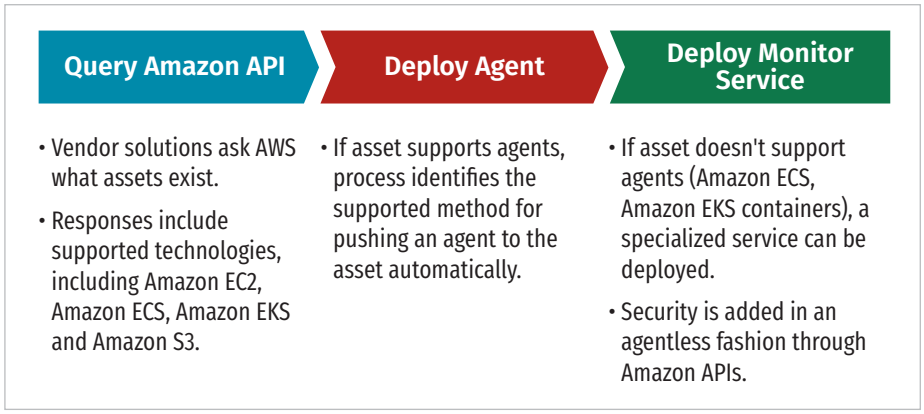


Figure 1. Modern Asset Identification Process

Deploying Controls to EC2 Instances

When implementing security controls to EC2 instances, it is imperative to plan for scale. What happens when you add or remove EC2 instances?

A good place to begin is with the Center for Internet Security's (CIS) Critical Security Controls[2] 1 and 2: Keep an inventory of authorized and unauthorized hardware and software. An effective AWS EDR strategy incorporates this principle by supporting automatic deployments.

Let's use CrowdStrike's EDR solution[3] to demonstrate how to integrate EDR in AWS. The process for deploying EDR in AWS using CrowdStrike follows these steps:

1. Subscribe to CrowdStrike EDR (found in AWS Marketplace).
2. Deploy CrowdStrike Falcon Discover.
   - **a.** Falcon Discover acquires access keys to query AWS. With these keys, it identifies all EC2 instances, even across regions.
   - **b.** The user authorizes Falcon Discover to deploy agents to specific EC2 instances or all instances automatically.
   - **c.** Agents continuously auto-deploy to authorized instances.
   - **d.** Optional: Falcon Discover is configured to monitor other assets such as CloudTrail. If enabled, this capability provides additional security controls such as alerting on tenant-level security controls.
3. The organization reports on asset coverage and monitors alerts.

An EDR solution should auto-scale and grow with you, not slow you down.

Through AWS integrations, EDR deployments are quick and efficient. Now both semi-permanent and temporary assets have EDR security controls continuously enabled. For example, deploying a multitiered web application using IIS, Microsoft SQL Server and middleware across 50 EC2 instances would result in 50 agents deployed without user intervention. Deploying 20 EC2 instances that fire up, run machine learning jobs and then terminate results in 20 agents deployed and decommissioned without user intervention. Coverage is no longer questionable but instead fully known.

---

[1] CrowdStrike, CrowdStrike Falcon and Falcon Discover are trademarks or registered trademarks of CrowdStrike Inc.

[2] www.cisecurity.org/controls

[3] This paper mentions solutions to provide a real-life example of how to integrate EDR in AWS. The use of these examples is not an endorsement of any solution.

Achieving Proper Security Controls

The phrase "Here be dragons" designates unexplored and potentially dangerous areas. For security professionals, there certainly are metaphorical dragons in EDR and caution is necessary. There are many products that claim to be EDR solutions. Although each of them provides endpoint controls, their depth of coverage and capabilities vary, resulting in different levels of protection.

Let's explore capabilities a successful EDR solution should provide by considering a plausible attack against an EC2 instance.

Attack Scenario

Consider this scenario:

> An organization is running a Windows EC2 instance with MSSQL services. An attacker is trying to identify critical assets but so far has only a standard account on a different EC2 instance. To escalate privileges, the adversary runs **`setspn`** to identify accounts vulnerable to what is commonly referred to as a *Kerberoasting*. Because MSSQL servers use service principal names (SPNs), the adversary finds the EC2 MSSQL service, pulls down a Kerberos ticket and then uses a password cracker to identify the MSSQL service account password. This account is then utilized to gain access to the EC2 instance using **`psexec`**. From there, the attacker establishes persistence by creating a digitally signed Microsoft executable due to a flaw from missing the patch for CVE-2020-0601, which allows abuse of the cryptographic process for Elliptic Curve Cryptography (ECC) handled by the Windows operating system. That process results in a persistent command and control that looks normal because the binary has been digitally signed by Microsoft. The further activity includes enumerating the MSSQL database.

Organizations need to cautiously evaluate EDR solutions against modern threats and risks.

The scenario provided is a bit convoluted. However, each step utilizes known attack techniques classified by the MITRE ATT&CK framework.[4] But just because something is a known technique does not mean it automatically should be blocked or flagged as an automatic alert. Consider the breakdown of this scenario:

- **MITRE T1208 Kerberoasting—`setspn`**, **`klist`** and PowerShell can be utilized to export a Kerberos token. This can then be password-cracked if the password is weak.

  Identification—Commands like **`setspn`** are not utilized by standard users and would often be an anomaly.

  Problem—System administrators do use **`setspn`**. Alerting on each use would generate multiple false positives.

- **MITRE T1035 and T1050—**The use of **`psexec`** to gain remote access would trigger a new service and its corresponding execution.

  Identification—`psexec` is not necessary if organizations use other remote access tools, such as PowerShell remoting.

  Problem—Organizations may utilize **`psexec`** as a standard remote access tool.

[4] https://attack.mitre.org; MITRE ATT&CK Matrix is a trademark of The MITRE Corp.

- **MITRE T1116—**Abusing CVE-2020-0601 to create a binary that appears to be digitally signed by Microsoft and then using that binary for persistent callbacks provides an adversary stealth communication.

  Identification—A digitally signed certificate should conform to proper Elliptic Curve Cryptography (ECC) standards.

  Problem—Software may use different algorithms, key lengths and other attributes when generating certificates.

- **MITRE T1219—**The adversary left a binary on the MSSQL server to maintain remote access.

  Identification—Persistence mechanisms generate network traffic to other assets that should not be happening.

  Problem—Because of asset management, patching and other system processes, it may be difficult to distinguish a good network callback from a malicious one.

Given this scenario, a proper EDR solution should provide multiple angles to identify the adversary. Each step could be a regular event. However, by analyzing the series of events, an EDR solution should clearly identify and even stop this attack. The following sections describe the features to look for in modern EDR solutions that would aid in this attack.

Process Tree

One method of finding unwanted activity is monitoring each process. This includes process, command line, parent process, parent process command line, user, integrity level and other related variables. This information then is correlated with the chain of processes occurring. EDR should identify abnormal processes or an abnormal chain of events and provide a visual process tree to explain why something is considered harmful (see Figure 2).

MITRE Tagging

Instead of reinventing the wheel, EDR solutions should integrate with known, proven frameworks. The MITRE ATT&CK framework is one of the most practical approaches to identifying attacker techniques, tools and behaviors. Each piece on its own is not enough to block an attack or generate an alert. However, specific techniques are more likely to be malicious than others, and EDR solutions can search for a

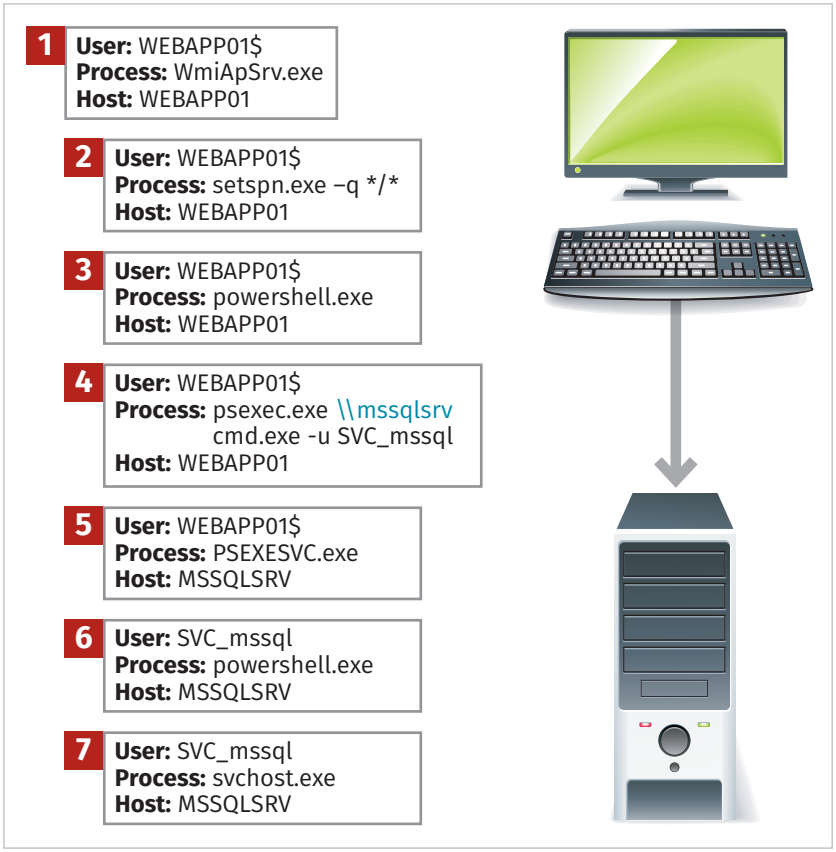


Figure 2. A Process Tree Diagram

combination or sequence of techniques and score them. Commercial EDR scores then combine to block or identify an attack, plus help analysts by telling a story of what happened. Figure 3 provides a sample visualization of the attack.

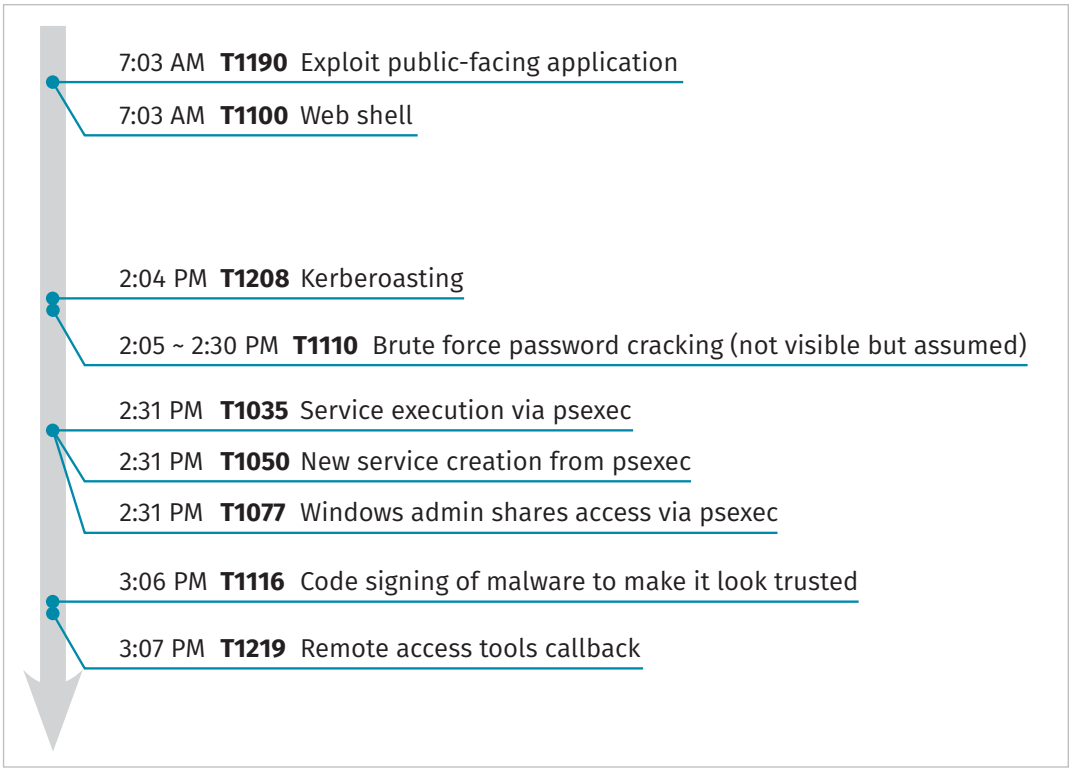


Figure 3. Graphical Description of the Attack

Signatures, Heuristics and Machine Learning

New attacks come out all the time. Therefore, EDR should include domain expert-based heuristics as well as potential algorithms that adapt over time, such as supervised or unsupervised learning. In the sample scenario, a basic heuristic check would identify that Kerberos reconnaissance commands were issued, followed by an authentication request from the original source EC2 instance. Machine learning may identify that the source user is highly unlikely to run commands like `klist` or `setspn`. Even traditional signatures may work by looking for an improperly formed Elliptic Curve Cryptography (ECC) generator set that abused CVE-2020-0601.

IoC Support

A robust EDR solution should offer the ability to identify a given activity and search for it across the entire environment. Put plainly, an organization should be able to identify the characteristics of an attack and document them in the form of an indicator of compromise (IoC). IoCs can be specific and straightforward, such as the `SHA1` of the binary used for persistence in the scenario. Or they can be specific, with broad characteristics such as looking for certificate files with specific algorithms, key lengths and file sizes.

EDR should be a sum of its parts: signatures for known bad, heuristics based on domain-expertise and machine learning for finding anomalies.

Ideally, IoC support should include vendor-defined IoCs that regularly update, plus the ability to develop internal IoCs and perform threat hunting with them. With such capabilities, organizations can perform investigations looking for IoCs from previously identified IoCs or proactively by looking for IoCs shared from external parties. Support for standard IoC formats such as `YARA` should be given consideration so that IoCs work outside the EDR platform.

Provide Attribution

Attribution is the ability to associate something with a person or entity. Within EDR, organizations should utilize MITRE ATT&CK and any proprietary sources to help them understand who or what is attacking them. At a minimum, such information is useful to understand what may occur next. For example, with profiling, various techniques, tools and IoCs may indicate that a known threat group is in play. In our scenario, profiling may inform the organization that the specific attack group has access to the EC2 instance, and the organization should look for specific backdoor programs. More importantly, the information can predict what the attack group's goal is, such as stealing healthcare information. Using this, an organization can make an informed decision to pause the EC2 instance or take alternative steps.

Response Capabilities

Given enough high-fidelity information, EDR should block or reverse the damage from the attack. If the attack was ransomware, EDR should restore encrypted files pre-ransomware. In our scenario, given that it is possible that the attack would not be blocked until a certificate was generated to exploit CVE-2020-0601, EDR would identify the attack and notify an analyst. Then, an analyst could choose to take remediation actions given in prior steps in the scenario. The response does not have to be the standard one of blocking a connection and removing a file. A response should mean taking steps against the full scenario—for example, removing the persistence file abusing CVE-2020-0601, killing the `psexec` process and killing the process providing remote access to the initial EC2 instance.

Real-Time Vulnerability Reporting

Because an EDR solution resides directly on an endpoint, it should identify all software that is installed or running. Because of this, it is possible to have real-time vulnerability reporting. Vulnerability scanning is hard to scale, but with an EDR partner that uses it for vulnerability reporting, it does not have to be.

EDR and Container Security

EDR solutions often employ agents for robust operating system visibility and protection mechanisms. But what about other deployments such as Amazon ECS and Amazon EKS containers? Some EDR solutions have no coverage for containers or anything that is not a traditional endpoint.

An EDR deployment in AWS should provide coverage to more than just EC2 instances. Fortunately, multiple vendors support a broader range of coverage in the AWS cloud. Regarding containers, the following foundational constraints need consideration.

- Containers, ideally, should run a single service.
    - **a.** Be sure to design a container around a single process.
    - **b.** Subprocesses such as an Nginx container running a master and worker are inline with best practices. Running multiple processes in parallel is not.

EDR should be a sum of its parts: signatures for known bad, heuristics based on domain-expertise and machine learning for finding anomalies.

- Containers should include only software that is necessary.
    - **a.** Adding an agent bloats container images.
    - **b.** Adding an agent also increases overhead computation, thus increasing costs.

Knowing the principles behind deploying and managing containers, deploying an agent is far from ideal. While technically an agent can be embedded into an image, it's a horrible idea due to the agent breaking the aforementioned foundational constraints. Still, most of the attack scenario described previously can work within containers, so if you use containers, be sure you identify an EDR solution that covers containers.

The implementation of EDR into containers requires software that can see into a container. Similar capabilities such as monitoring processes, files and network connections need to work inside a container. To accomplish this, either an agent needs to be deployed into each container, or an image, a sidecar container or a centralized agent needs to be implemented. Let's explore each option:

- **Agent—**Installing agents inside a container is against good practice, hard to manage and computationally expensive.
- **Sidecar—**A sidecar is a concept of deploying a container next to another container, similar to a motorcycle with a sidecar attached. In this case, the sidecar container receives access to the original container so it can monitor it. Technically this option works, but it adds additional computing resources and overhead to ensure each container gets a sidecar.
- **Centralized agent—**A better approach is to have one or more specialized agents that utilize Amazon APIs and access to dip into containers and corresponding images. For example, CrowdStrike EDR supports deploying a single instance of Falcon Insight. Falcon Insight then acts as a centralized agent that interfaces with Amazon ECS and Amazon EKS to secure containers and images.

Using an EDR solution that supports AWS integration dramatically simplifies deployment and ensures minimal gaps in security controls. A centralized agent such as Falcon Insight would identify the CVE-2020-0601 vulnerability in an MSSQL Server image or notify the analyst that the image is no longer vulnerable, but active containers still are. In addition, containers do not run full operating systems, and an EDR solution can more readily apply heuristics and anomaly detection. For example, an MSSQL container should only be running MSSQL. If a binary began a persistent callback mechanism, an EDR solution should be able to intervene to detect and block it.

While all assets eventually are decommissioned, containers are decommissioned much more so. Their ephemeral nature introduces new challenges that only a modern EDR can solve. Consider an MSSQL container that gets infected but later is stopped due to scheduled maintenance. After the maintenance, a new container is deployed without any known vulnerabilities. The problem is the old container included crucial forensics evidence regardless of the compromise. A reliable EDR solution would provide a way to access terminated containers in order to provide analysis in an ad hoc or as-needed basis. If data was stolen in the prior scenario, the solution could launch an investigation that analyzes the previously decommissioned container.

EDR Integrations: A Platinum Experience

EDR provides multiple angles of coverage from native AWS integration, asset knowledge, and detection and prevention capabilities, up to threat hunting and intelligence. Because of the extensive visibility capabilities and IoC support, organizations should consider EDR for a third-party integration. What if a breach occurred and data and/or malware was transferred into an S3 bucket or later shifted to an external SaaS provider, such as Dropbox? With data moved outside the endpoint, EDR protection generally stops. Yet some EDR solutions go the extra mile and support integration with other solutions, such as cloud security providers like Netskope.[5]

Instead of running multiple security solutions in parallel, they can be integrated. Think of this as a platinum experience, going above and beyond. Via AWS Marketplace, organizations can quickly subscribe and deploy multiple solutions. Then via partner sharing and documentation, they can quickly integrate multiple products into Amazon's APIs as well as from partner to partner APIs. The result is a streamlined solution with extended coverage.

As an example, consider the use of CrowdStrike and Netskope integration. The two solutions support integration and sharing of IoCs. They also support dynamic access control lists as a result. An IoC showing malware or files stolen can be shared as an IoC in CrowdStrike to Netskope and help identify where the files were staged or moved within multiple cloud tenants. Or maybe the attack never would have succeeded. In the scenario described earlier, the adversary first had to get onto the initial EC2 instance before pivoting to the MSSQL Server. If the first EC2 server was missing CrowdStrike's EDR agent, then a dynamic access control could limit cloud access via the CrowdStrike and Netskope integration. This control may also limit or identify the attacker trying to access or stage payloads.

Conclusion

The definition of an endpoint is evolving. Endpoints are moving past EC2 virtual machines, and it is imperative for EDR solutions to evolve and support this evolution. AWS is quickly adopting new methodologies of implementing and deploying endpoints as well as technologies such as infrastructure as code. As a result, organizations must understand the gaps and risks of not knowing and understanding the various endpoints found in their AWS infrastructure. Organizations should consider an EDR solution that provides advanced controls and works with their AWS environment rather than around it.

Organizations should choose an EDR that encompasses the multiple types of endpoints, such as Amazon EC2, Amazon ECS and Amazon EKS. Because of other infrastructures, such as containers, EDR needs to move past the mantra of every asset getting an agent. New methods such as centralized agents with Amazon API integration are required to come close to 100% asset coverage remotely. Asset coverage and security controls are further extended with EDR solutions that integrate with other partners via API hooks.

[5] Netskope is a trademark of Netskope Inc.

About the Author

Justin Henderson is a certified SANS instructor who authored the SEC555 SIEM with Tactical Analytics course and co-authored SEC455: SIEM Design and Implementation and SEC530: Defensible Security Architecture and Engineering. He is a passionate security researcher and security consultant with over a decade of experience in consulting and is one of the co-founders of H & A Security Solutions. Justin is the 13th of 20 GSEs to become both a red and blue SANS Cyber Guardian and holds 61 industry certifications. He specializes in threat hunting via SIEM, network security monitoring and ad hoc scripting.

Sponsor

SANS would like to thank this paper's sponsor:

aws marketplace

aws marketplace

Enhance your EDR strategy with AWS services and third-party solutions.

Security operations teams looking to enhance their AWS investigations with EDR must develop a holistic approach that includes integrating with virtual machines, containers, storage, and serverless code. Organizations should consider an EDR solution that provides advanced controls and works with their AWS environment rather than around it.

CrowdStrike Falcon Endpoint Protection Enterprise is a solution that can interface with Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS) to secure containers and images. CrowdStrike's indicator of attack (IOA)-based threat prevention can also be integrated with Amazon GuardDuty to help stop known and unknown threats in real-time.

Amazon GuardDuty is a managed threat detection service that identifies threats, such as unusual API calls or potentially unauthorized users attempting to access your servers. It can send all of its findings to AWS Security Hub, which provides a consolidated view of security-related information. With AWS Security Hub, you have a single place that aggregates, organizes, and prioritizes your security alerts, or findings, from multiple AWS services and AWS Marketplace ISV solutions.

How AWS customers are leveraging CrowdStrike to enhance their security investigations

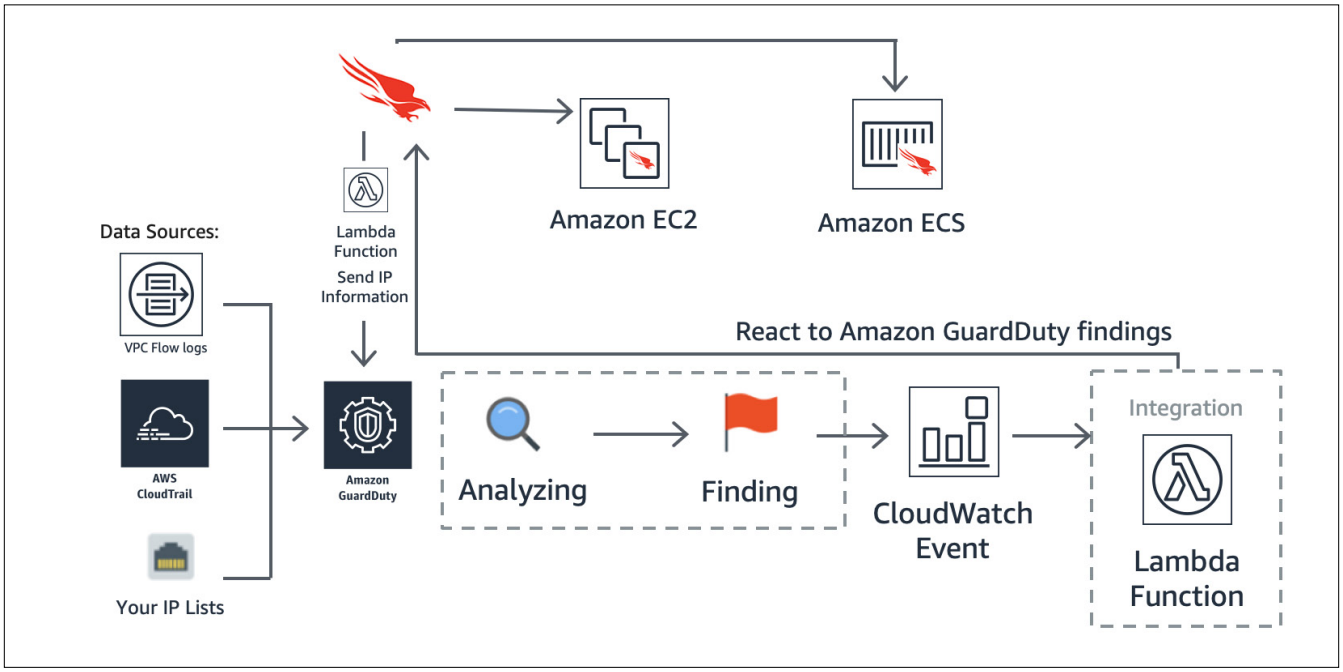
CrowdStrike is an AI-enabled endpoint protection platform that can help improve security investigations and accelerate detection and response across your AWS environment. Some of the ways that customers are leveraging CrowdStrike to enhance their EDR strategy include:

- **Provides complete visibility:** CrowdStrike offers extensive AWS visibility across your environment, accounts, and instances. The platform also offers full incident visibility that includes details, context, and history for every alert.

- **Delivers efficiency with MITRE ATT&CK framework:** CrowdStrike adopts the MITRE ATT&CK framework across multiple processes and products to make alerts easy to understand and evaluate. MITRE ATT&CK provides a list of tactics under each technique, so viewers can quickly spot the attacker's objective. In addition, security analysts can learn more about observed unusual behavior on their endpoints by clicking on different elements of the alert.

Mapping to the MITRE ATT&CK framework helps users instantly prioritize alerts by ranking their potential threat level. This saves customers time and provides additional clarity and advanced context for understanding their security alerts.

- **Stop attacks with IOA detection:** CrowdStrike helps AWS customers by stopping attacks with IOA detection. By recording and gathering the indicators of attack and consuming them via a Stateful Execution Inspection Engine, security teams can view malicious activity in real-time and infer intent to stop it in time.



CrowdStrike, coupled with Netskope Public Cloud Security, also provides a unified approach to threat intelligence exchange, endpoint protection, and adaptive access control. For example, stolen files can be shared as an indicator of compromise (IoC) in CrowdStrike to Netskope to identify where the files were staged or moved across your cloud tenants.

By adding a security information and event management (SIEM) solution to your EDR, organizations can increase the efficiency and visibility of their security investigations. For example, Sumo Logic's Cloud-Native Machine Data Analytics Service offers unique insights into your vulnerabilities, authentication services, anti-virus scans, domain name system (DNS) requests, and document access. Sumo Logic also uses real-time machine data to provide a comprehensive analysis of your CrowdStrike Falcon platform environment.

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 endpoint detection and response 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 solutions can help you get started:

CROWDSTRIKE

CrowdStrike Falcon Endpoint Protection Enterprise
Unified cloud native endpoint protection platform

netskope

Netskope Public Cloud Security
Real-time data and threat protection for cloud and web services

sumo logic

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