CYBER
THREAT
ANALYSIS

Recorded Future®

By Insikt Group®

August 23, 2022

# DETECTIONS IN THE SKY:
# Sigma Rules to Enhance Cloud Security for the Big Three

*This report provides an overview of tools used to attack the 3 major cloud providers, Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). It contains details on the capabilities of these attack tools and provides detections for suspicious and malicious behavior against these cloud providers. This report is intended for security operations audiences who focus on detection engineering. Sources include the Recorded Future® Platform and open-source research.*

## Executive Summary

Many organizations are migrating their data, resources, and/or services to the cloud. The cloud offers organizations the ability to scale services and provide capabilities that would not otherwise be feasible with the organization's on-premises resources. With the increased use of cloud services, the need for organizations to properly secure and monitor their cloud environments becomes more critical. Attacks against cloud infrastructure look different and require a unique approach to detection when compared to on-premises infrastructure. As a result, cloud infrastructure security best practices are distinct from other best practices that can be applied to conventional infrastructure.

While the Big Three platforms, Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) all have built-in security controls and provide logging capabilities, there is still ambiguity on threat detection and response for cloud environments. Our research provides details on the different types of attacks that can be performed against the Big Three platforms, such as artifact deletion, command execution, and cloud environment enumeration, among others, as well as Sigma rules to detect a subset of the attacks that we emulated as part of our research.

## Key Judgments

- The Big Three platforms provide logging capabilities to detect most malicious activity if combined with the infrastructure to collect, aggregate, and correlate the logs.
- The tools developed for attacking AWS and Azure are more sophisticated compared to those targeting GCP.
- Detection engineering for attacks against cloud infrastructure are complicated due to the distributed and ephemeral nature of the cloud, the many different workloads that can be run from a cloud infrastructure, and differences between the cloud providers.
- Opportunities for detecting attacks targeting cloud service providers exist on both the cloud provider as well as the hosted virtual machines for both Azure and GCP.

## Background

### Cloud Workloads

There are many different ways an organization can use cloud resources, referred to as cloud workloads. Dell defines a cloud workload as a "specific application, service, capability or a specific amount of work that can be run on a cloud resource". These workloads are potential targets for, or are at risk of, exposure from cloud-related intrusions.

Gartner has stated that the top 7 workloads that should be migrated to the cloud are:

1. Mobile device and application support
2. Collaboration and content management
3. Videoconference
4. Virtual desktops and remote workstation management
5. Scale-out application
6. Disaster recovery
7. Business continuity solutions

### The Big Three

We chose the top 3 providers for cloud infrastructure-as-a-service (IaaS) as the focal points of this research: Amazon Web Services (AWS), Azure, and Google Cloud Platform (GCP). Together, these platforms represent over 62% of the global cloud IaaS market for Q1 2022, with AWS at 33% of market share, Azure at 21%, and GCP at 8%.

All three platforms have similar service offerings; however, in a recent blog by BMC, AWS and Azure both offer approximately 200+ services, while GCP offers 100+. BMC also outlines the common services that each platform offers; the following is a summary of their research:

- Compute services (VMs, platform-as-a-service, containers and serverless functionality)
- Database and storage services (relational database management, NoSQL, object storage, file storage, archive storage, data warehouse/data lake)
- Networking (virtual network, load-balancing, firewall, DNS, content delivery)

BMC also states that for some of the more specialized services like robotics, end-user computing, game development, virtual reality, and IoT, both Azure and AWS offer more services when compared to GCP.

In the end, the choice of which Big Three platform to use is more dependent on the particular use case the organization is looking to solve.

## The Big Three Threat Analysis

Securing cloud infrastructure inherently follows a security model that was first introduced by

Sounil Yu at the 3rd Annual National Cybersecurity Summit (2020), and is known as the DIE triad. The DIE triad is an adapted version of the well-known CIA triad (confidentiality, integrity, and availability) but focuses more on the infrastructure on which an organization's resources or data reside. Copado also provides the following definitions for the DIE triad:

- Distributed: Are systems distributed to allow for scalability while preventing dependence on a single zone?
- Immutable: Can the infrastructure be disposed of and replaced in the event of an issue, aka infrastructure-as-code?
- Ephemeral: What's the period for system reprovisioning, and are assets disposable in the event of a breach?

Threat-modeling attacks against cloud infrastructure are increasingly complicated due to the distributed and ephemeral nature of the cloud and the variety of workloads that can be run from a cloud infrastructure. As a result, cloud infrastructure varies from organization to organization, and threats that may directly affect one organization's cloud infrastructure may not directly affect another organization's cloud infrastructure. These differences help illustrate the diverse nature of cloud environments and aid in explaining challenges associated with creating a uniform security model for cloud environments.

Using the Recorded Future Platform and OSINT research to identify cyber threats affecting each of these platforms, we found that misconfigurations present the greatest risk to Amazon AWS instances, followed by credential theft, for allowing threat actors to gain initial access. The most common misconfigurations that lead to compromise stem from cloud service configurations that allow the environment to be publicly accessible. Microsoft Azure, however, was affected by a diverse set of cybersecurity threats, of which misconfigurations were not a big portion. Finally, Google Cloud products were targeted in only 1 cyber incident, and were mentioned very few times in relation to misconfiguration issues.

There are many resources available to organizations to guide them in the implementation of cloud infrastructure security best practices. Such resources should be examined for applicability to an individual organization's situation, but common ones include those of the Cloud Security Alliance, the United States National Institute of Standards and Technology, and the US Cybersecurity & Infrastructure Security Agency.

For this research, our approach was to identify common tools and frameworks used to attack the Big Three platforms and then use those tools against our test environments. Figure 1 below illustrates our cloud detection methodology. At a high level, our methodology consisted of running these malicious tools against the cloud provider test environments we created; we analyzed the cloud provider logs to look for artifacts indicating the use of these tools and created Sigma rules based on them where possible.

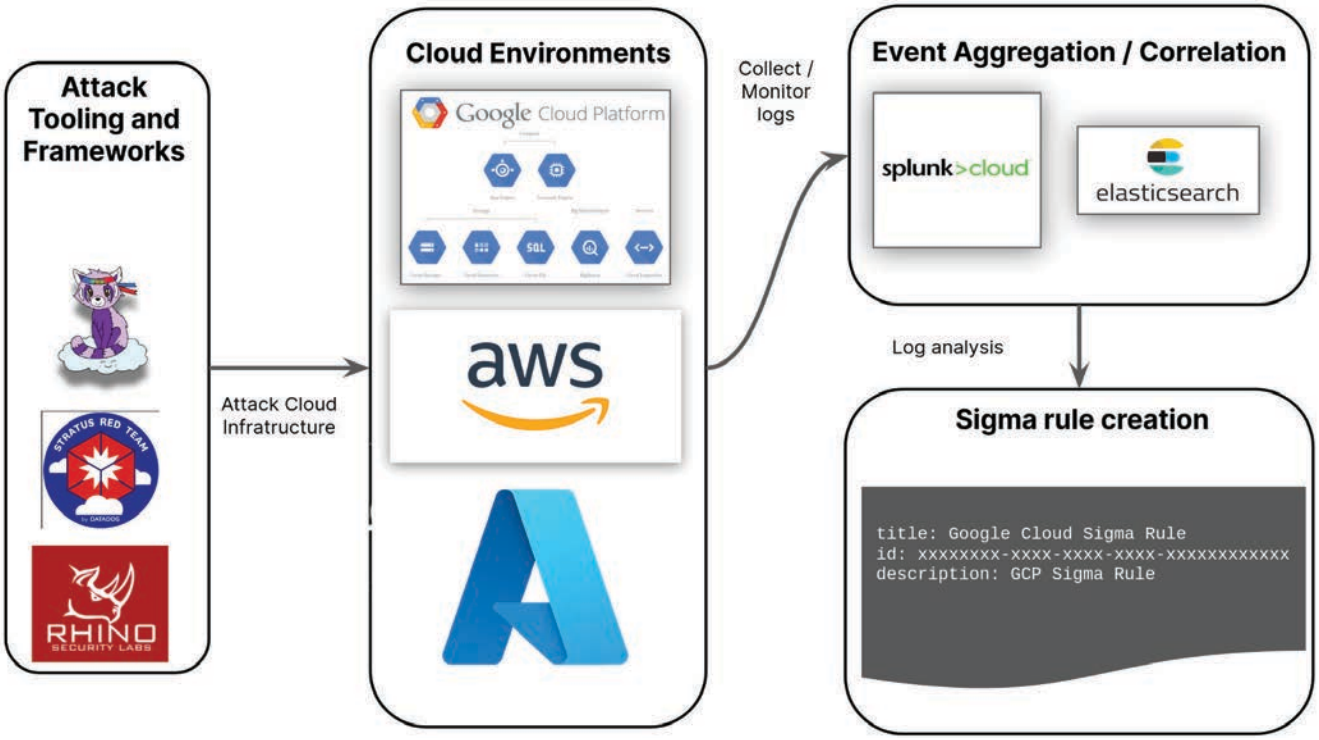


*Figure 1: Insikt Group cloud detection engineering methodology (Source: Recorded Future)*

## AWS Threat Analysis

### *Attack Surface*

AWS environments are commonly targeted by threat actors of all varieties with highly disparate motivations and attack techniques. There are myriad tools that could be used to emulate common attacks that affect AWS environments and to demonstrate attack strategies that may be implemented by a threat actor. We have identified a selection of these tools, focusing on recency and breadth of potential attack types, and used them to generate log data for the AWS logging service CloudTrail. Using this data, we created Sigma detections for the displayed behavior, allowing defenders and threat researchers to better identify potentially malicious activity in AWS environments and the use of these tools.

The tools that we used to generate CloudTrail log data are described below.

### *Stratus Red-Team*

Stratus Red-Team is a penetration testing tool created by DataDog, the creator of a cloud-based logging, monitoring, and security platform. Stratus is based on Red Canary's Atomic Red Team pentesting suite; however, Stratus focuses on cloud environments whereas Atomic Red Team is used to target Windows, Linux, and MacOS operating systems and networks.

Stratus contains more than 30 pre-written attack techniques, as shown in Figure 2. All but 7 of these techniques are used to perform various attacks against AWS environments. These attacks are varied and span the credential access, defense evasion, discovery, execution, privilege escalation, exfiltration, initial access, and persistence ATT&CK tactics.

| TECHNIQUE ID | TECHNIQUE NAME | PLATFORM | MITRE ATT&CK TACTIC |
|---|---|---|---|
| aws.credential-access.ec2-get-password-data | Retrieve EC2 Password Data | AWS | Credential Access |
| aws.credential-access.ec2-steal-instance-credentials | Steal EC2 Instance Credentials | AWS | Credential Access |
| aws.credential-access.secretsmanager-retrieve-secrets | Retrieve a High Number of Secrets Manager secrets | AWS | Credential Access |
| aws.credential-access.ssm-retrieve-securestring-parameters | Retrieve And Decrypt SSM Parameters | AWS | Credential Access |
| aws.defense-evasion.cloudtrail-delete | Delete CloudTrail Trail | AWS | Defense Evasion |
| aws.defense-evasion.cloudtrail-event-selectors | Disable CloudTrail Logging Through Event Selectors | AWS | Defense Evasion |
| aws.defense-evasion.cloudtrail-lifecycle-rule | CloudTrail Logs Impairment Through S3 Lifecycle Rule | AWS | Defense Evasion |
| aws.defense-evasion.cloudtrail-stop | Stop CloudTrail Trail | AWS | Defense Evasion |
| aws.defense-evasion.organizations-leave | Attempt to Leave the AWS Organization | AWS | Defense Evasion |
| aws.defense-evasion.vpc-remove-flow-logs | Remove VPC Flow Logs | AWS | Defense Evasion |
| aws.discovery.ec2-enumerate-from-instance | Execute Discovery Commands on an EC2 Instance | AWS | Discovery |
| aws.discovery.ec2-download-user-data | Download EC2 Instance User Data | AWS | Discovery |
| aws.execution.ec2-user-data | Execute Commands on EC2 Instance via User Data | AWS | Execution |
| | | | Privilege Escalation |
| aws.exfiltration.ec2-security-group-open-port-22-ingress | Open Ingress Port 22 on a Security Group | AWS | Exfiltration |
| aws.exfiltration.ec2-share-ami | Exfiltrate an AMI by Sharing It | AWS | Exfiltration |
| aws.exfiltration.ec2-share-ebs-snapshot | Exfiltrate EBS Snapshot by Sharing It | AWS | Exfiltration |
| aws.exfiltration.rds-share-snapshot | Exfiltrate RDS Snapshot by Sharing | AWS | Exfiltration |
| aws.exfiltration.s3-backdoor-bucket-policy | Backdoor an S3 Bucket via its Bucket Policy | AWS | Exfiltration |
| aws.initial-access.console-login-without-mfa | Console Login without MFA | AWS | Initial Access |
| aws.persistence.iam-backdoor-role | Backdoor an IAM Role | AWS | Persistence |
| aws.persistence.iam-backdoor-user | Create an Access Key on an IAM User | AWS | Persistence |
| | | | Privilege Escalation |
| aws.persistence.iam-create-admin-user | Create an administrative IAM User | AWS | Persistence |
| | | | Privilege Escalation |
| aws.persistence.iam-create-user-login-profile | Create a Login Profile on an IAM User | AWS | Persistence |
| | | | Privilege Escalation |
| aws.persistence.lambda-backdoor-function | Backdoor Lambda Function Through Resource-Based Policy | AWS | Persistence |
| azure.execution.vm-run-command | Execute Commands on Virtual Machine using Run Command | azure | Execution |
| k8s.credential-access.dump-secrets | Dump All Secrets | kubernetes | Credential Access |
| k8s.credential-access.steal-serviceaccount-token | Steal Pod Service Account Token | kubernetes | Credential Access |
| k8s.persistence.create-admin-clusterrole | Create Admin ClusterRole | kubernetes | Persistence |
| | | | Privilege Escalation |
| k8s.privilege-escalation.hostpath-volume | Container breakout via hostPath volume mount | kubernetes | Privilege Escalation |
| k8s.privilege-escalation.nodes-proxy | Privilege escalation through node/proxy permissions | kubernetes | Privilege Escalation |
| k8s.privilege-escalation.privileged-pod | Run a Privileged Pod | kubernetes | Privilege Escalation |

*Figure 2: List of attack techniques Stratus is capable of performing (Source: Recorded Future)*

### *Pacu*

Pacu is an open-source exploitation framework that focuses on performing attack techniques against AWS cloud platforms using pre-written modules. The tool is intended to be used for red-teaming to identify vulnerabilities and misconfigurations within AWS environments. It currently has 42 modules that perform reconnaissance, discovery (enumeration), privilege escalation, lateral movement, persistence, exfiltration, and defense evasion, among other tactics. Pacu also includes a list of exploitation-specific modules that allow the user to perform actions including:

- The automatic creation of identity and access management (IAM) keys in a target environment
- Lightsail SSH key discovery, SSH key generation, and temporary access-granting
- Remote code execution on EC2 instances as SYSTEM/ root

## *Sigma Rule Detections*

### *AWS Sigma Rule: Stratus Activity Detection*

This rule detects the string stratus-red-team, which is included in the user-agent any time Stratus executes a module. While Stratus is mainly used to demonstrate malicious activity for red-teaming purposes, it is possible that a threat actor could modify Stratus modules to weaponize the tool. Stratus's reliance on the Terraform infrastructure-as-code (IaC) tool to perform attacks and initialize the target environment means it is possible that the threat actor will leave the user-agent unmodified and the user-agent will continue to include stratus-red-team.

```
title: MAL_AWS_Cloudtrail_Stratus_Red_Team
id: f38da2e0-83d2-4113-88b3-4018dfeb3aa7
description: Detect operations made by the Stratus Red Team Pentesting Tool
references:
    - https://github.com/DataDog/stratus-red-team
status: stable
author: CKOVACS, Insikt Group, Recorded Future
date: 2022/06/21
level: medium
tags:
    - attack.t1530 # Data From Cloud Storage Object
    - attack.t1070 # Indicator Removal on Host
    - attack.t1528 # Steal Application Access Token
    - attack.t1562.008 # Impair Defenses: Disable Cloud Logs
    - attack.t1562.001 # Impair Defenses: Disable or Modify Tools
    - attack.t1526 # Cloud Service Discovery
    - attack.t1537 # Transfer Data To Cloud Account
    - attack.t1071 # Application Layer Protocol
    - attack.t1136.003 # Create Account: Cloud Account
    - attack.t1134.003 # Access Token Manipulation
    - attack.t1548 # Abuse Elevation Control Mechanism
    - attack.t1078.004 # Valid Accounts: Cloud Accounts
logsource:
    product: aws
    service: cloudtrail
detection:
    UA:
        userAgent|contains: stratus-red-team
    condition: UA
falsepositives:
    - N/A
```

*Table 1: MAL_AWS_Cloudtrail_Stratus_Red_Team Sigma rule (Source: Recorded Future)*

***AWS Sigma Rule: Port 22 Ingress***

The Stratus module aws.exfiltration.ec2-security-group-open-port-22-ingress allows the user to create a security group allowing inbound access to the environment over port 22. By creating this security group a threat actor would have a route to drop tools or provide C2 commands within a victim environment. The general activity of allowing port 22 ingress can be detected by identifying events with the event name AuthorizeSecurityGroupIngress and by identifying events where the request parameters toPort contain the string 22.

```
title: SUSP_AWS_Cloudtrail_Ingress_Over_Port_22
id: df1ad094-69be-4536-b54d-517f90f6844b
description: Rule to detect newly-created security group policies allowing in-
gress channels over port 22
references:
    - Internal research
status: stable
author: CKOVACS, Insikt Group, Recorded Future
date: 2022/06/23
level: medium
tags:
    - attack.t1105 # Ingress Tool Transfer
    - attack.t1041 # Exfiltration Over C2 Channel
logsource:
    product: aws
    service: cloudtrail
detection:
    port_22:
        requestParameters.toPort: 22
    ingress:
        eventName: AuthorizeSecurityGroupIngress
    condition: port_22 and ingress
falsepositives:
    - Legitimate ingress rules allowing traffic to port 22
```

*Table 2: SUSP_AWS_Cloudtrail_Ingress_Over_Port_22 Sigma rule (Source: Recorded Future)*

***AWS Sigma Rule: CloudTrail Trail Deletion***

The Stratus module aws.defense-evasion.cloudtrail-delete allows users to create a CloudTrail trail to remove logging data from the AWS environment. This technique can be detected simply by identifying events with the event name DeleteTrail but these events should be scrutinized to ensure that the deletion was not performed legitimately.

```
title: SUSP_AWS_Cloudtrail_Trail_Deletion
id: bc229f1b-705f-46fc-8104-12837a9600e7
description: Rule to detect when a Cloudtrail trail has been deleted
references:
    - Internal research
status: stable
author: CKOVACS, Insikt Group, Recorded Future
date: 2022/06/23
level: medium
tags:
    - attack.t1070 # Indicator Removal on Host
logsource:
    category: DeleteTrail
    product: aws
    service: cloudtrail
detection:
    deleteTrail:
        eventName: DeleteTrail
    condition: deleteTrail
falsepositives:
    - Legitimate trail deletions initiated by an administrator
```

*Table 3: SUSP_AWS_Cloudtrail_Trail_Deletion Sigma rule (Source: Recorded Future)*

### *Additional Mitigations*

In addition to the Sigma rules above, we recommend the following to mitigate malicious behavior in an AWS environment:

- Utilize compound detection analysis with additional data, such as hardware usage, user activity logs, and network traffic logs (among other data) to better determine whether an action is malicious or innocuous; keep detailed records of a newly identified or ongoing attack.
- Create baselines based on logging to determine common activity that occurs within your AWS environment. Use these baselines to identify aberrant activity and behavior that occurs within the environment.
- Whenever an asset is created, ensure that the security group associated with it does not allow any network connections by default. This can be achieved by creating an initialization security group that blocks all internal and external network requests. If an asset requires networking, determine the connections that must be established and then create a security group that permits only the connections that are necessary for the asset to function.
    - Note: Ensure that the user that created the asset or the root user is still capable of accessing the asset based on the initialization security group's configurations. It is possible to configure a security group that makes accessing an asset impossible, even by the AWS subscription's root user, by removing all network access.
- Define user roles using AWS's IAM and then determine what permissions these users have based on the role.
    - Do not create overly permissive roles that would allow a user to gain permissions similar to an administrator-level user.
    - Ensure that only the smallest possible number of trusted users (likely a system administrator within your organization) have access to the root account for your AWS subscription.
- Create security policies for AWS-hosted assets that restrict the access of certain objects and services to users with specific, predetermined roles.
- Do not expose details about the AWS environment (such as user credentials, API keys, or Amazon Resource Names [ARN]) in public spaces (such as GitHub, StackOverflow, or PasteBin). These can be used by threat actors to gain a foothold within an AWS environment or to perform reconnaissance against the environment.

## Azure Threat Analysis

Azure environments are targeted less frequently than AWS environments, based on previous research conducted by Recorded Future, and similarly have fewer tools and proof-of-concept attack techniques associated with the cloud platform than AWS. Despite this, attacks and malicious usages of Azure remain common, and successful attacks have demonstrated that the impact of compromising an Azure environment can be severe, based on our previous research.

In this section we investigate malicious behavior that a threat actor may exhibit during an attack on an Azure environment. The 2 key malicious functionalities we identified and tested for Azure were centered around execution and data exfiltration.

### *Attack Scenario*

As part of this research, we conducted several malicious actions that an attacker could perform on Azure, primarily using the penetration testing tool Stratus Red-Team. Stratus Red-Team offers 3 techniques for Azure: Execute Command on Virtual Machine using Custom Script Extension; Execute Commands on Virtual Machine using Run Command; and Export Disk Through SAS (Shared Access Signature) URL.

### *Sigma Rule Detections*

We wrote 3 Sigma rules based on each of the Stratus Red-Team techniques currently available for Azure.

#### *Execution Techniques*

Executing a command on a virtual machine using a custom script extension allows an attacker to pass PowerShell commands to the VM as SYSTEM. Custom Script Extension can run scripts downloaded from Azure Storage, GitHub, or provided to the Azure portal extension at runtime. PowerShell is a full-featured scripting language, and is often used by threat actors to download subsequent stages of malware, but can also be used to script the malware payload itself. When a script is deployed, it is stored on the Windows virtual machine in C:\Packages\Plugins\Microsoft.Compute.CustomScriptExtension\<version number>\Downloads\.

The Sigma rule below detects log events of type Microsoft.Compute/virtualMachines/extensions/write in the Azure logs.

```
title: MAL_Azure_Stratus_Command_Execution_VM_Script
id: bbdb4493-8904-4b3d-bd1e-11b46d4aef19
description: Detect commands being run on VM's using the CustomScriptExtension
references:
    - https://stratus-red-team.cloud/attack-techniques/azure/azure.execution.
vm-custom-script-extension/
status: stable
author: LKAYE, Insikt Group, Recorded Future
date: 2022/07/12
level: low
tags:
    - attack.t1530 # Data From Cloud Storage Object
    - attack.t1059.001 # Command and Scripting Interpreter: PowerShell
    - attack.t1078.004 # Valid Accounts: Cloud Accounts
logsource:
    product: azure
    service: azureactivity

detection:
    StartDiskSharingOperation:
      OperationNameValue: "MICROSOFT.COMPUTE/VIRTUALMACHINES/EXTENSIONS/WRITE"
      ActivityStatusValue: "Start"
      ResourceProviderValue: "MICROSOFT.*"
      CategoryValue: "Administrative"

    condition: StartDiskSharingOperation
falsepositives:
    - Legitimate usage of this feature by administrator
```

*Table 4: MAL_Azure_Stratus_Command_Execution_VM_Script Sigma rule (Source: Recorded Future)*

Executing a command on a virtual machine using the RunCommand feature can also be used to run PowerShell commands on a Windows VM as SYSTEM, and on Linux, it can be used to run shell commands as root. On Windows, the command can be run via the Azure portal, REST API, or PowerShell; on Linux, the command can be run via the Azure portal, REST API, or Azure CLI. In both cases, the threat actor is able to execute a command with elevated privileges. When a command is run on Windows, it is uploaded to the virtual machine via the agent and stored in a .ps1 file in C:\Packages\Plugins\Microsoft.CPlat.Core.RunCommandWindows\<version number>\Downloads. On Linux virtual machines, each run command creates a new directory per job in /var/lib/waagent/run-command/download/<job ID>.

The Sigma rule below detects events of type Microsoft.Compute/virtualMachines/runCommand/action in the Azure logs.

```
title: MAL_Azure_Stratus_VM_Command_Execution
id: e65f9547-545d-4282-ab44-a704933203d5
description: Detect commands being run on Azure VMs using 'RunCommand'
references:
    - https://stratus-red-team.cloud/attack-techniques/azure/azure.execution.vm-
run-command/
status: stable
author: LKAYE, Insikt Group, Recorded Future
date: 2022/07/11
level: low
tags:
    - attack.t1059 # Command and Scripting Interpreter
    - attack.t1530 # Data From Cloud Storage Object
    - attack.t1078.004 # Valid Accounts: Cloud Accounts
logsource:
    product: azure
    service: azureactivity

detection:
    RunCommandExecution:
      OperationNameValue: "MICROSOFT.COMPUTE/VIRTUALMACHINES/RUNCOMMAND/ACTION"
      ActivityStatusValue: "Start"
      CategoryValue: "Administrative"
      ResourceProviderValue: "MICROSOFT.*"

    condition: RunCommandExecution
falsepositives:
    - Legitimate use of the RunCommand functionality
```

*Table 5: MAL_Azure_Stratus_VM_Command_Execution Sigma rule (Source: Recorded Future)*

***Data Exfiltration Techniques***

Finally, exporting a virtual machine disk through SAS allows a threat actor to exfiltrate data from Azure. A SAS URI can be generated for unattached managed disks and snapshots to allow the data on the Azure Disk to be exported. If a threat actor knows the SAS URI, they can download the disk without any IP filtering before the expiration time, which is defined when the URI is created.

The Sigma rule below detects events of type Microsoft.Compute/disks/beginGetAccess/action in the Azure logs.

```
title: MAL_Azure_Stratus_Export_Disk_SAS
id: 1d827cdd-3183-4667-be83-0cf8dae6c795
description: Detect data exfiltration from Azure using a managed disk export feature
references:
    - https://stratus-red-team.cloud/attack-techniques/azure/azure.exfiltration.disk-ex-
port/
    - https://techcommunity.microsoft.com/t5/azure-architecture-blog/how-to-block-azure-
vhd-download/ba-p/1609898
    - https://docs.microsoft.com/en-us/azure/storage/common/storage-sas-overview
    - https://zigmax.net/azure-disk-data-exfiltration/
status: stable
author: LKAYE, Insikt Group, Recorded Future
date: 2022/07/12
level: low
tags:
    - attack.t1078.004 # Valid Accounts: Cloud Accounts
    - attack.t1530 # Data From Cloud Storage Object
    - attack.t1619 # Cloud Storage Object Discovery
logsource:
    product: azure
    service: azureactivity

detection:
    StartDiskSharingOperation:
      OperationNameValue: "MICROSOFT.COMPUTE/DISKS/BEGINGETACCESS/ACTION"
      ActivityStatusValue: "Start"
      CategoryValue: "Administrative"
      ResourceProviderValue: "MICROSOFT.*"

    condition: StartDiskSharingOperation

falsepositives:
    - Legitimate user exporting the Azure Disk
```

*Table 6: MAL_Azure_Stratus_Export_Disk_SAS Sigma rule (Source: Recorded Future)*

### *Additional Mitigations*

In addition to the Sigma rules above, we recommend the following to mitigate malicious behavior in an Azure environment:

- Configure the endpoints of Azure Disks to be "private" or select "Deny all" if export functionality is not needed, since Azure Disks are configured with a public endpoint for import and export of the disks by default.
- Consider defining new roles and limiting permissions for actions that can be used by threat actors as part of custom script execution or to run commands:
    - Microsoft.ClassicCompute/virtualMachines/extensions/write
    - Microsoft.Compute/virtualMachines/extensions/write
    - Microsoft.Compute/virtualMachines/runCommand/action
- Monitor a virtual machine's activity log for unusual behavior, and create alerts for "Run Command on Virtual Machine" and "Create or Update Virtual Machine Extension" if these are not regularly used by your organization.
- On the Linux and Windows virtual machines themselves, monitor for file creation events in the directories identified above for both RunCommand and Custom Script Execution.

## Google Cloud Threat Analysis

### *Attack Surface*

The Google Cybersecurity Action Team released a 2022 Threat Horizons report aimed at providing actionable intelligence to help organizations protect and mitigate cloud attacks. The report is based on intelligence from Google's Threat Analysis Team (TAG), Google Cloud Threat Intelligence for Chronicle, Trust, and Safety and other internal groups. The report advises that adversaries are using the following approach for their attacks:

1. Scanning for Apache Log4J vulnerable host and similar public facing exploits
2. Using known open-source tools like Sliver for backdoor and C2 capabilities
3. Using native cloud services such as Cloud Shell for reverse SSH tunneling
4. Using previously identified malicious domains

The report specifically mentions the backdoor Sliver, which is used once access is gained to a host. There are also tools to aid in achieving initial access or privilege escalation; some of the more popular ones are PurplePanda, Sneak, gcpHound, and CloudBrute.

#### *PurplePanda*

On April 22, 2022, GitHub user carlospolop updated a tool called PurplePanda, which is a reconnaissance tool that fetches resources from the cloud or software-as-a-service (SaaS) applications. It focuses on acquiring permission information for the purpose of identifying privilege escalation paths and dangerous permissions that may be exploited. It currently supports the following cloud applications:

1. GitHub
2. Google Cloud Platform (GCP)
3. Kubernetes (K8s)

PurplePanda searches within a platform and across platforms for privilege escalation paths. It also supports identifying the application and conducting a generic search. The tool supports 2 analysis modes: the "enumeration" mode is used to gather and analyze data, and the "analyze" mode is used to analyze provided credentials. At the time of writing, the tool has 50 forks and 402 "stars" on GitHub.

#### *Sneak*

On March 10, 2022, GitHub user ex0dus-0x released a proof-of-concept tool called "Sneak". Sneak is an offensive cloud security tool written in Go that is designed to leak and exfiltrate sensitive data from the instance metadata service (IMDS). The tool also allows users to enumerate server-side request forgery (SSRF) vulnerabilities in a cloud environment. The project is currently capable of enumerating environmental variables and collecting cloud metadata from AWS IMDSv1, Google Cloud, DigitalOcean, and Microsoft Azure. The author plans to extend functionality to other "network services", but has not described which services at this time. At the time of this writing, the tool has 3 forks and 7 "stars" on GitHub.

#### *gcpHound*

Offensive security engineers Madhav Bhatt and Brad Richardson, who are affiliated with financial company Credit Karma, introduced a tool called gcpHound via a blogpost by Madhav Bhatt. gcpHound is referred to as a "Swiss Army knife" and is an offensive toolkit for targeting environments in Google Cloud Platform (GCP). At the time of writing, gcpHound can be used for:

1. Enumerating user permissions and groups for privilege escalation
2. Persistence
3. Lateral movement
4. Discovering and collecting Google Cloud Storage (GCS) buckets
5. Exfiltrating data from GCS buckets

gcpHound has functions to enumerate IAM permissions of target organizations and projects. It can also enumerate all groups of an organization and fetch members of those groups. According to the author, gcpHound is still under development and future plans for the tool include the addition of features such as attacking secret management, abusing Google Cloud Key Management Service (KMS) to decrypt buckets, and leveraging network permission to find firewall rules of interest. The tool can be pulled from Docker and details for how to do so can be found in the above-mentioned blogpost.

### *CloudBrute*

A social media account @0xsha released an update on a tool they authored called CloudBrute. CloudBrute is a multi-platform tool that finds and enumerates a target company's cloud infrastructure, files, open buckets, applications, and databases hosted on top cloud providers and possibly applications behind proxy servers. The new update features a new cloud detection method. The new engine detects the cloud from HTML and JavaScript source codes when detection from IP address data fails. It has also added DigitalOcean applications on its supported Cloud providers. At the time of this writing, the tool has 82 forks and 484 "stars" on GitHub.

## *Sigma Rule Detections*

To emulate attacks against GCP we ran the tools previously mentioned — PurplePanda, Sneak, and gcpHound — against our GCP environment. We did not emulate CloudBrute as its main functionality against GCP was against storage buckets and can be detected with the open-source Sigma rules here. All the logs generated from the tool emulation were sent to Splunk Cloud using the "Splunk Add-on for Google Cloud Platform" plugin and then log analysis was performed from there.

We wrote Sigma rules for PurplePanda and gcpHound. Sneak stayed true to its name and didn't create any unique log artifacts that could be used for detection using Sigma.

### *GCP Sigma Rule: Google Cloud GCPHound*

gcpHound is run from a Dockerized environment; the following Sigma rule detects the static components of that environment along with an action to get the IAM policy. The static components we identified are:

- Google SDK Version(s):
    - google-cloud-sdk gcloud/366.0.0
    - google-cloud-sdk gcloud/354.0.0
- Python version string:
    - python/3.8.10

```
title: MAL_GCP_gcpHound_Enumeration
id: c6d4568d-4fbc-4606-962e-ab2cd6acd41b
description: Detects GCPHound enumeration running from dockerized environment
references:
  - https://desi-jarvis.medium.com/gcphound-a-swiss-army-knife-offensive-tool-
kit-for-google-cloud-platform-gcp-fb9e18b959b4
author: JGROSFELT, Insikt Group, Recorded Future
date: 2022/06/15
level: medium
tags:
  - attack.t1078.004 # Valid Accounts: Cloud Accounts
  - attack.t1087.004 # Account Discovery: Cloud Account
  - attack.t1580 # Cloud Infrastructure Discovery
  - attack.t1526 # Cloud Service Discovery
logsource:
  product: gcp
  service: gcp.audit
detection:
  sdk_version:
    protoPayload.requestMetadata.callerSuppliedUserAgent|contains:
      - google-cloud-sdk gcloud/366.0.0
      - google-cloud-sdk gcloud/354.0.0
  python_version:
    protoPayload.requestMetadata.callerSuppliedUserAgent|contains|contains: py-
thon/3.8.10
  gloud:
    protoPayload.requestMetadata.callerSuppliedUserAgent|contains: gzip(gfe)
  method:
    protoPayload.methodName: GetIamPolicy
  condition: sdk_version and python_version and gloud and method

falsepositives:
  - Legitimate tools that also use older SDK versions and the specific Python
version
```

*Table 7: MAL_GCP_gcpHound_Enumeration Sigma rule (Source: Recorded Future)*

***GCP Sigma Rule: Google Cloud PurplePanda Enumeration***

We could not identify a single event to be used to definitively detect use of PurplePanda. However, with correlation, we can detect with high confidence the use of repeated scanning of PurplePanda or related tooling. During our analysis we saw more than 25 events calling the following commands within a 180-second time span using the Google CLI user-agent:

- GetIAMPolicy
- ListServiceAccountKeys
- ListServiceAccounts
- ListRoles

```
title: MAL_GCP_Purple_Panda_Enumeration
id: f68834e8-9a54-449a-8d85-9c8a57b7aa82
description: Detects commands used by Purple Panda for cloud enumeration. IM-
PORTANT this rule relies on correlation to be effective (see correlation)
references:
  - https://github.com/carlospolop/PurplePanda/tree/master/intel/google
author: JGROSFELT, Insikt Group, Recorded Future
date: 2022/06/15
level: medium
tags:
  - attack.t1078.004 # Valid Accounts: Cloud Accounts
  - attack.t1087.004 # Account Discovery: Cloud Account
  - attack.t1580 # Cloud Infrastructure Discovery
  - attack.t1526 # Cloud Service Discovery
correlation:
    - Events should occur within a 180 second timespan
    - Should be greater than 25 matching events within the time period
    - For Splunk, append this to the end of the converted rule, '| transaction
maxspan=180s | search linecount > 25'
logsource:
  product: gcp
  service: gcp.audit
detection:
  commands:
    protoPayload.methodName|endswith:
      - GetIAMPolicy
      - ListServiceAccountKeys
      - ListServiceAccounts
      - ListRoles
  useragent:
    protoPayload.requestMetadata.callerSuppliedUserAgent: (gzip),gzip(gfe)
  condition: commands and useragent
falsepositives:
  - Legitimate identity management enumeration tools
```

*Table 8: MAL_GCP_Purple_Panda_Enumeration Sigma rule (Source: Recorded Future)*

### *Additional Mitigations*

In addition to the Sigma rules above, we recommend the following actions to mitigate malicious behavior in a GCP environment, based on best-practices-for-enterprise-organizations from Google.

#### *Google Identities*

- Use fully managed Google accounts that are tied to your corporate domain and manage them through Cloud Identity, which allows you to enable or disable access to Google services.
- Use the Identity and Access Management tool to manage the types of access users have to specific resources. Apply the security principle of least privilege, and grant only the necessary access to your resources.
- Have well-defined groups that most of your user base fits into so that you can manage access control on a group level and not a user level. Google recommends the following groups:

| Group | Function |
|---|---|
| gcp-organization-admins | Organization admins are responsible for organizing the structure of the resources used by the organization. |
| gcp-network-admins | Network admins are responsible for creating networks, subnets, firewall rules, and network devices such as Cloud Router, Cloud VPN, and cloud load-balancers. |
| gcp-security-admins | Security admins are responsible for establishing and managing security policies for the entire organization, including access management and organization constraint policies. |
| gcp-billing-admins | Billing admins are responsible for setting up billing accounts and monitoring their usage. |
| gcp-devops | DevOps practitioners create or manage end-to-end pipelines that support continuous integration and delivery, monitoring, and system provisioning. |
| gcp-developers | Developers are responsible for designing, coding, and testing applications. |

*Table 8: Google-recommended groups for Google Cloud Platform (Source: Google)*

***Network Security***

- Use Virtual Private Cloud (VPC) and subnets to map out your network. VPCs provide flexible network options for your Compute Engine VM as well as the services that use your VM instances, including but not limited to Google Kubernetes Engine (GKE), Dataproc, and Dataflow.
- Each VPC network has a virtual firewall. The firewall rules should be configured to allow or deny traffic to your VM instances. Best practices for firewall rules are below:
    - Block all traffic by default and only allow the specific traffic required for the resources.
    - Use a hierarchical firewall policy that first blocks traffic that should never be allowed.
    - Your allow rules should be restricted to specific VMs or Service Accounts.
    - Turn on Firewall Rules Logging and use Firewall Insights to verify that firewall rules are being used in the intended way.
- Limit access to the public internet to only those resources that need it. Use Private Google Access for resources that have a private IP but still need to access the Google APIs.

***Network Security of Applications and Data***

- Limit risk of data exposure by using VPC Service Controls to define a security perimeter around your resources.
- Use a Google Cloud global HTTP(S) load-balancer to provide both high availability and scaling for your internet-facing services and combine the load-balancer with Google Cloud Armor to provide DDoS protection
- Use an Identity-Aware Proxy (IAP) to control access to your applications and verify user identity.

## Outlook

Events triggered in cloud environments by offensive security tools demonstrate that cloud platforms have additional attack surfaces and logging capabilities that can be used for detecting threats. Given the disparate and ephemeral nature of cloud infrastructure, these threats may be unique or highly targeted, and will often require multiple layers of defense, detection, and analysis to properly identify and mitigate.

While data leaks due to exposed and misconfigured cloud instances are very common, publicly-accessible data suggests that the number of known threat actor groups that have conducted intrusions specifically targeting AWS, GCP, and Microsoft Azure environments remains relatively small when compared to more traditional environments, such as on-premises and locally-hosted environments. However, as cloud services continue to grow across corporate enterprises and more companies shift away from on-premises solutions, it is very likely that threat actors will increasingly seek to target and attack these services.

The Sigma rules provided aim to help detect cloud abuse attempts from the growing number of threat actors targeting cloud services; however, misconfigurations remain the primary concern for most cloud system users. Organizations operating cloud infrastructure should work to ensure that their environments are properly implemented and managed and are regularly audited. Those systems and software that are not the responsibility of the cloud platform should have firmly established update and patching processes. Standard security concepts such as least-privilege and security monitoring should be used in cloud environments as well as the on-premises components of an organization.

## About Insikt Group®

Insikt Group is Recorded Future's threat research division, comprising analysts and security researchers with deep government, law enforcement, military, and intelligence agency experience. Their mission is to produce intelligence on a range of cyber and geopolitical threats that reduces risk for clients, enables tangible outcomes, and prevents business disruption. Coverage areas include research on state-sponsored threat groups; financially-motivated threat actors on the darknet and criminal underground; newly emerging malware and attacker infrastructure; strategic geopolitics; and influence operations.

## About Recorded Future®

Recorded Future is the world's largest intelligence company. Recorded Future's cloud-based Intelligence Platform provides the most complete coverage across adversaries, infrastructure, and targets. By combining persistent and pervasive automated data collection and analytics with human analysis, Recorded Future provides real-time visibility into the vast digital landscape and empowers clients to take proactive action to disrupt adversaries and keep their people, systems, and infrastructure safe. Headquartered in Boston with offices and employees around the world, Recorded Future works with more than 1,400 businesses and government organizations across more than 60 countries.

Learn more at recordedfuture.com and follow us on Twitter at @RecordedFuture.