Catching "EC2 Grouper"- No Indicators Required! | FortiGuard Labs

By Chris Hall

Published: 2024-12-30 · Archived: 2026-04-05 13:33:05 UTC

Through the years of analyzing identity compromises in the cloud, we've seen the same attackers pop up regularly, some more frequently than others. Among the more prolific ones we've come to know is one we've dubbed "EC2 Grouper". Over the past couple of years, we've seen this actor in several dozen customer environments, making them one of the more active groups we've tracked. This usual suspect is attributed by their penchant for using similar user agents and the same security group naming convention in their attacks.

While indicators such as user agents and even security group names can assist in attribution and hunting, we have found them unreliable for comprehensive threat detection. In this blog, we'll detail tactics associated with EC2 Grouper and how Lacework FortiCNAPP can be leveraged to detect this threat, among others. More importantly, we will showcase how this is achieved without relying on actor-specific indicators, which can be transient in nature.

Tactics and Techniques

EC2 Grouper is characterized by their usage of AWS tools for PowerShell to carry out attacks. This is presumed by their user agent, which was consistent for a number of years:

AWSPowerShell.Common/4.1.90.0 .NET_Core/6.0.5 OS/Microsoft_Windows_10.0.17763 PowerShellCore/7.-1 ClientAsync

In recent attacks, they have updated their UA, which now contains new versioning and unusual # characters, which could indicate a possible detection countermeasure.

AWSPowerShell.Common/4.1.534.0 ua/2.0 .NET_Core#6.0.5 OS/windows#10.0.17763.0 md/ARCH#X64 PowerShellCore/7.-1 cfg/retry-mode#legacy md/ClientAsync

A more consistent indicator has emerged with a security group naming convention. Attacks in the cloud often leverage the CreateSecurityGroup API (T1098) to enable remote access and lateral movement in the cloud environment. EC2 Grouper will typically attempt to create multiple groups using the same naming convention of ec2group suffixed with a sequential combination of 1-5. Example request parameters:

{"groupDescription":"ec2group","groupName":"ec2group"}

{"groupDescription":"ec2group1","groupName":"ec2group1"}

{"groupDescription":"ec2group12","groupName":"ec2group12"}

{"groupDescription":"ec2group123","groupName":"ec2group123"}

{"groupDescription":"ec2group1234","groupName":"ec2group1234"}

{"groupDescription":"ec2group12345","groupName":"ec2group12345"}

In all instances of EC2 Grouper attacks, cloud activity appears to be largely automated. The attacker will initially make calls to DescribeInstanceTypes to inventory EC2 types within the environment and then DescribeRegions to retrieve information about regions available for resources. Upon acquiring available regions, the following API calls are iteratively executed for every available region:

- DescribeVpcs: Requesting information about VPCs (Virtual Private Clouds) in a given region. (T1580)
- CreateSecurityGroup: Creation of a new security group called "ec2group." (T1098 & T1021)
- DescribeSecurityGroups: Querying the security groups available in the account.
- DescribeAccountAttributes: Acquiring account attributes, perhaps for quotas related to resources. (T1580)
- GetServiceQuota: Checking service quotas, possibly to identify limits on resource usage. (T1580)
- DescribeInstances: Gathering information about existing EC2 instances that may be running, pending, or shutting down. (T1580)
- RunInstances: Attempting to launch new EC2 instances using the ec2group security group. (T1496)

Interestingly, we have never observed calls to AuthorizeSecurityGroupIngress, which is ultimately required to configure inbound access to any EC2 launched with the security group. However, on several occasions, we have observed CreateInternetGateway and CreateVpc, which are required for remote access. To date, we have not observed what could be classified as actions based on objectives or manual activity in a compromised cloud environment. It could be either that EC2 Grouper is selective in their escalation or compromised accounts were detected and quarantined before they had the opportunity to escalate. Despite this, resource hijacking (T1496) is likely the general objective. However, to what end is currently unconfirmed.

Detection

In every attack involving valid accounts, the credentials must originate from somewhere. One of the more common sources for compromised keys remains code repositories. Developers often mistakenly commit cloud access keys to public repositories. Once this occurs, the clock starts ticking until the credentials fall into the hands of attackers, are discovered by secret scanners, or both. This is believed to be the primary method of credential acquisition for EC2 Grouper, as their cloud attacks are frequently accompanied by attacks from other threat actors. EC2 Grouper, however, is by far the most prolific actor allegedly using this vector.

Given the popularity of obtaining credentials in code repositories, it can be prudent to look for legitimate secret scanning services as part of your detection strategy. These include GitGuardian and Github’s secret scanning

service. In our composite alerts, we have included secret scanning as a signal, as it is frequently seen in conjunction with illicit credential usage.

Of course, credential checking alone does not indicate a compromise, so other signals need to be correlated to reduce false positives. When alerting on EC2 Grouper, our composite alerts have evaluated other techniques, such as using specific APIs known to be leveraged in attacks. These are effectively mapped to the respective techniques with the assistance of the open-source TDiscover project.

Finally, we evaluate anomalies as part of the composite alert. An alleged attack may exhibit characteristics indicative of malicious reconnaissance or privilege escalation. However, it's crucial to confirm this through anomaly detection.

Conclusion

Identifying illicit usage of valid credentials in the cloud can be a nuanced and difficult task. This poses a considerable challenge when it comes to detection, as the vast majority of attacks in the cloud involve compromised credentials. While the attack detailed in this blog had various atomic indicators specific to the actors' tactics and techniques, most attacks do not exhibit these unique characteristics. To achieve higher accuracy, it becomes more critical to correlate weaker signals involving aspects that attackers cannot control. For example, while attackers can easily control their source IP and user agent, they *cannot* control whether it is anomalous to the environment. Similarly, they cannot control the APIs or sequence of APIs needed to carry out their objectives. By leveraging these as signals to a composite alerting mechanism, one can achieve a much higher level of detection efficacy.

How Can Lacework FortiCNAPP help?

Cloud detection and response (CDR) is a crucial component in addressing cloud identity compromises such as the one documented here. With over 80% of attacks in the cloud involving compromised credentials, the effectiveness of your CDR solution can directly dictate the severity of a cloud attack. Lacework FortiCNAPP offers comprehensive CDR protection with our innovative composite alerting technology. Cloud identity compromises can be difficult to isolate as they often blend in with legitimate activity. Lacework FortiCNAPP can evaluate numerous weak signals together through composite alerting, culminating in a much higher detection efficacy than point detection alone. Lacework FortiCNAPP also integrates other essential components, such as CIEM for informing the blast-radius of a compromised identity.

Read more about how Lacework FortiCNAPP can secure your cloud environment.
