Overview:

If the attacker changes the permissions of a role or the group, they could give themselves access to various privileges, such as, changing roles to have full access. These could prove to be detrimental to the system.

What Is A User?
A *user* is an entity that is created in AWS. A user represents a person or service as they interact with AWS. A user gives people the ability to sign in to the AWS Management Console and to interact with AWS services using API or CLI. At creation, users are given permissions by adding them to a group or individual policies.

What Is A Group?
A *group* is a collection of users. An IAM group is a collection of IAM users. Groups can be used to specify permissions for any collection of users. A group can be used to simplify the process of managing permissions for those users.

Example:

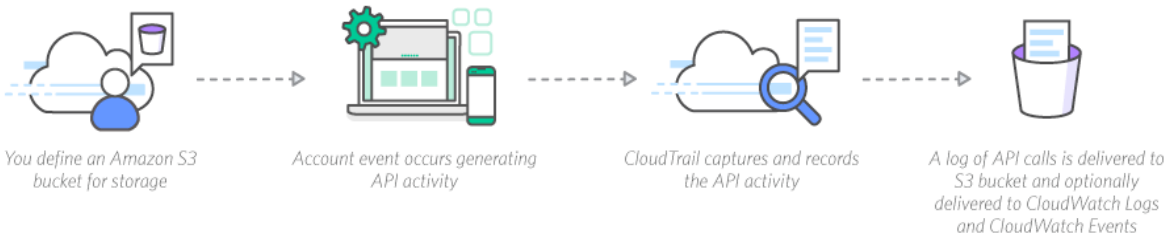
There is a group *Developers*, who have the permissions that developers typically need. Every user in this group automatically has permissions that are assigned to the group. All members in *Developers* have the ability to create, edit, and manage the code; however are unable to deploy the code. If a new user joins your organization and should have developer privileges, the proper permissions can be assigned by adding that user to the group. Similarly, if an employee's job changes and they should no longer have access to everything developers have access to, they can simply be removed from the group.

What Is A Role?
An IAM *role*, similarly to users, is an identity with permission policies. The policies determine what the identity can and cannot do in AWS. A role, unlike users, do not require credentials. A role is intended to be used by anyone who needs it. IAM users can temporarily assume a role to be given those specific permissions for a specific task. They can be used to delegate access to users, applications, or services that don't normally have access to your AWS resources. Roles can be assigned within the company, or to an external user.

Example:

You might want to grant users in your AWS account access to resources they don't usually have, or grant users in one AWS account access to resources in another account.

Discovery:



It is important to keep track of all ongoing activity logs of users to make sure that they are only using the applications that they have been given permission to.

AWS has logging features that can determine what actions users are taking as well as the resources they are using.

What Do Log Files Show?
The log files show:

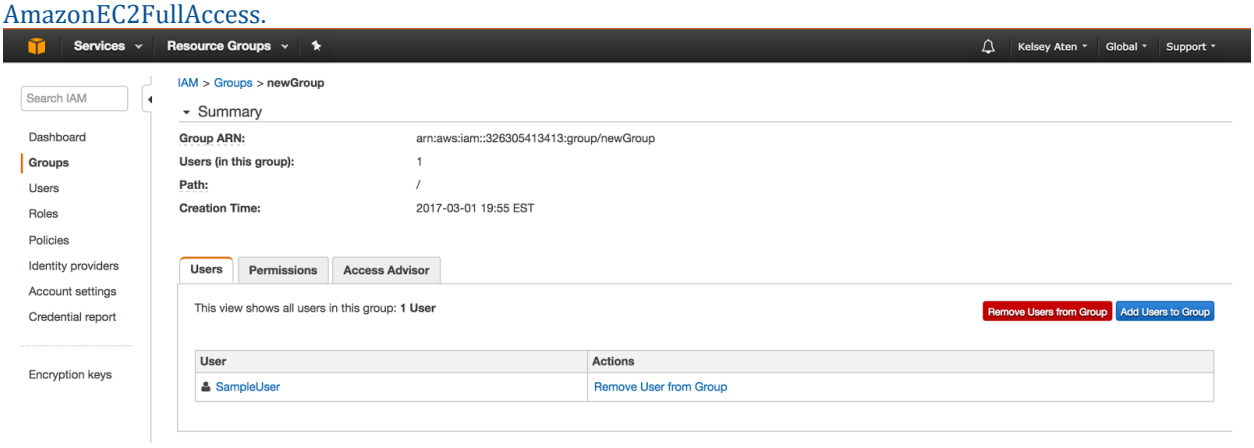
- Time and date of actions
- Source IP for an action
- Actions that failed due to inadequate permissions, and more.

AWS CloudTrail
AWS IAM, Identity and Access Management, is integrated with CloudTrail logs. Cloudtrail "logs AWS API calls and related events made by or on behalf of an AWS account."

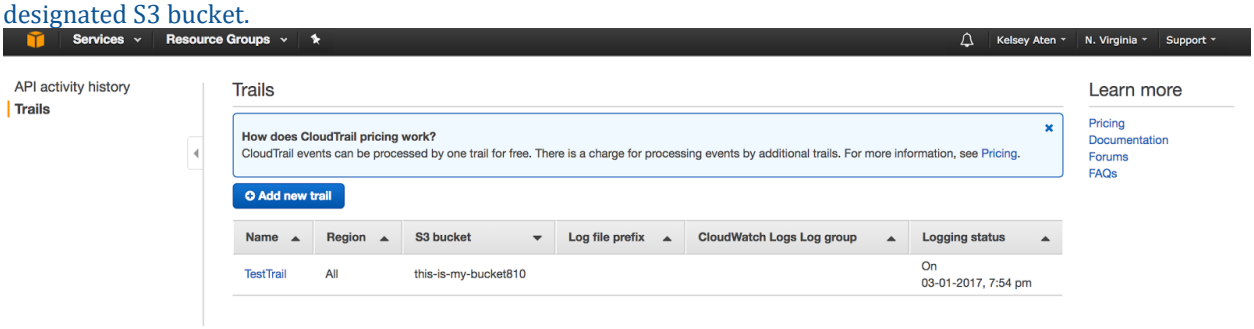
AW API Event in CloudTrail Log File
If a user changes as AWS role the user, time, and IP address will be stored in the log file. This is important to look at when a threat of a changed IAM role is suspected.

This can be shown in the figures below:

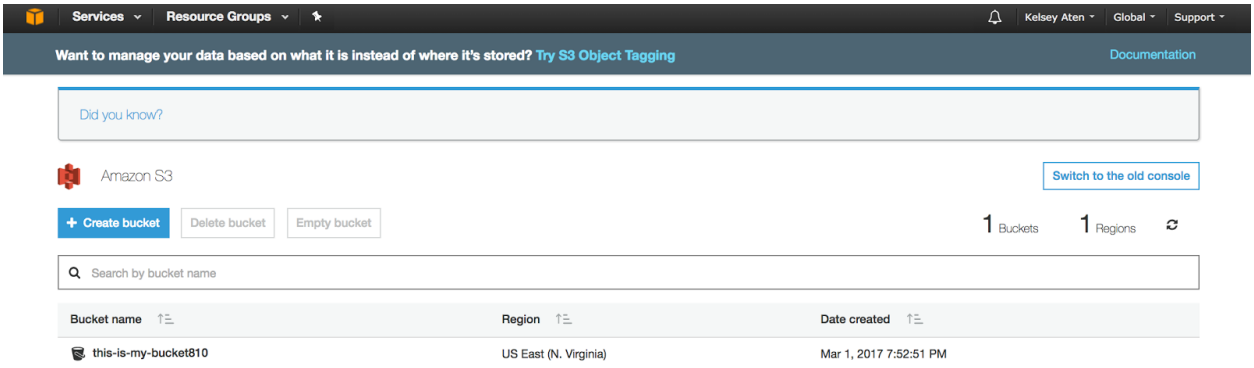
The new user below, SampleUser was added to the group newGroup, giving them access to AmazonEC2FullAccess.



After creating a trail names TestTrail in CloudTrail, with the S3 Bucket, the logs will be saved into the designated S3 bucket.



The logs can then be viewed within that S3 Bucket under the Amazon service S3.



More Information: https://aws.amazon.com/cloudtrail/ and http://docs.aws.amazon.com/IAM/latest/UserGuide/cloudtrail-integration.html#cloudtrail-integration-iam-information

Taken directly from http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

Remediation:

To remediate an AWS role or group attack on AWS, it is important to limit the access on who can change and/or create roles and groups on AWS. Another important thing to consider is separation of duties, the act of requiring multiple people to complete a task. Separation by sharing a task is a great internal control that will help prevent attacks on AWS IAM users.

Prevention:

By default, AWS IAM is secure. When an IAM user is created, they have no access to any services. Users are given access only when permissions are explicitly set.

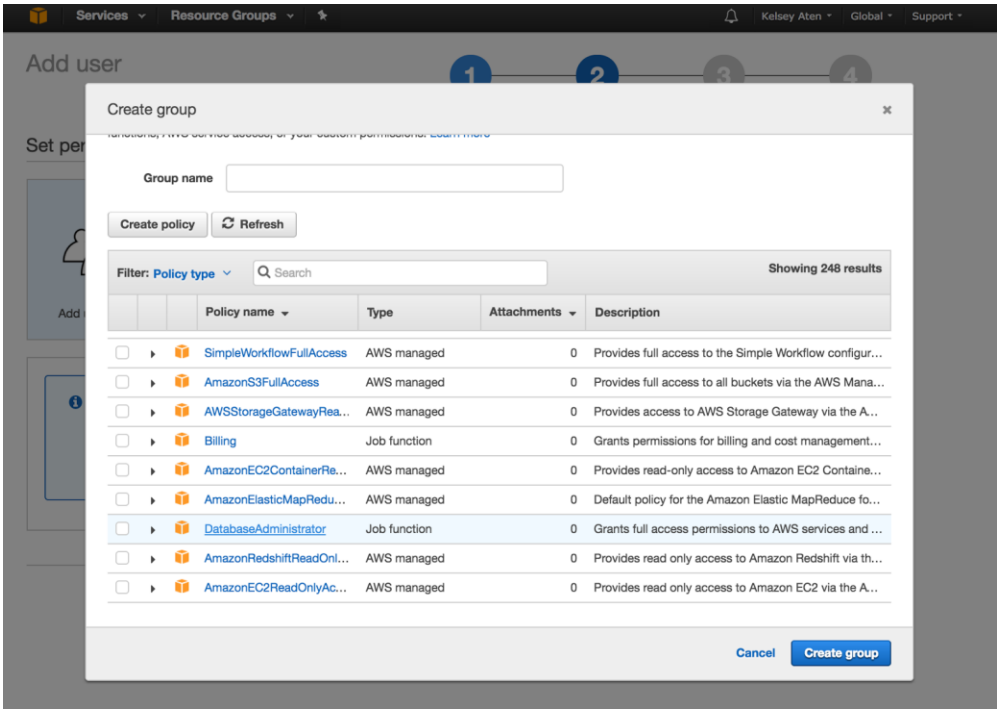
The best way to secure this is to Grant least privilege.

In the creation of IAM policies it is important to grant least privilege. Granting least privilege consists of granting only the permissions required to perform a task. It is important to determine what specifically the user will be doing and create policies that will allow them to perform **only** the necessary things in those tasks.

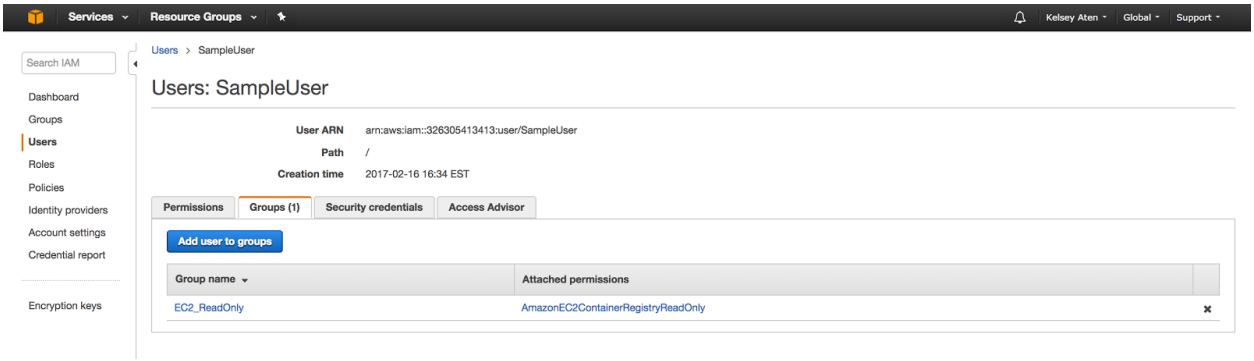
It is more secure to start by granting just the minimum permissions for each user and granting additional permission when necessary.

It is vital to be able to properly determine what it required for each task. Such as, what actions particular services support, and what permissions will be required by those actions.

As seen in the screenshot below, the user is being granted DatabaseAdministrator group access, this is *full* access to al AWS services and databases. This is something that should not be granted to the majority of employees as it would be giving them too much access.



A more common example of groups IAM users are added to include the EC2_ReadOnly group. This is the perfect group to add someone to that does not need to be able to edit the EC2 services, but should be able to view them and watch what is happening.



Useful Tool:

Access Advisor tab:

As seen above, on the left hand side of the figure, the Access Advisor tab is available in the IAM console summary page. The tab includes information about which services are actually used by a *user, group, role* or *policy*. This information can be used to identify what permissions will be necessary and what may be unnecessary. This allows you to keep IAM policies to the least privilege policy.

Websites Used:

http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html
https://d0.awsstatic.com/whitepapers/aws-security-whitepaper.pdf
http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html