A Focus on Identity and Access Management Deprovisioning

Santiago Salas
M.S. in Cybersecurity Candidate
NYU Tandon
santi.f.salas@nyu.edu

***Abstract*—This paper aims to discuss the process of deprovisioning accounts in infrastructures that use Identity and Access Management (IAM). A mock of an IAM hierarchy is created showing permission bindings. A use case that shows potential lingering access and over-provisioning access is analyzed. Then a possible solution of archiving permissions is suggested. The implementations are compared and limitations are discussed. The goal of the paper is to focus on implementations of removing permissions which is an area that has lacked focus in practical IAM implementations.**

***Index Terms*—identity, access, management, iam, deprovisioning**

I. Introduction

Identity and Access Management (IAM) is something that has been utilized in infrastructures to manage access to resources. It has become more common due to cloud architectures adopting IAM and lowering the barrier of entry for organizations. It is relatively easier now to setup a cloud infrastructure with Google Cloud Platform and Amazon Web Services being the two biggest providers.

IAM strategies follow a core principal of least privilege. However the strategies that are created and published focus mainly on the granting of permissions. This leads to the removal of permissions lacking well established processes for certain use cases. The research done here focuses on providing alternative strategies to address the removal of permissions. This is meant to act as a supplement to already implemented IAM strategies in an infrastructure.

This paper's contents are structured as follows. Section II - Related Research discusses some of the previous research done in IAM strategies and highlights areas where this research can prove as a valuable supplement to that research. In Section III - Motivating Example some use cases will be listed and analyzed. They'll be implemented in a mock environment. These use cases would aim to show how certain IAM implementations may leave security gaps when not removing permissions appropriately. Section IV - Empirical Evidence will show the implementation of an alternative that will be referred to as "IAM escrow". In Section V - Conclusion some key points are summarized as well as some limitations to consider.

II. Related Research

Ali M. Al-Khouri [1] identified manual employee account termination as an inefficiency, as well as the governing system to be complex when multiple groups perform account creation and deletion. Another issue identified in this research was that managing strategies that were being utilized were intended for the moment and not positioned for long term usage. This research was done with a focus on how government agencies can better optimize their IAM strategies to improve service delivery for their citizens. The solution was to incorporate smart cards into the identification strategy to help with scalability problems. There is not much focus on describing how deprovisioning access will function, but the system model shows that rights are assigned to identities where the card is a significant part of establishing an identity. So an identity is defined as a person and their card, therefore issuing a new token could break permission bindings as a way to decommission.

M. Gaedke et al. [2] developed a new model to address the problem of issuing permissions across organizational boundaries. The key thing here is the separation between the different organizations. Each organization has full control of permissions to their assets and delegation rules are used to provide an identity with a token that contain permissions to applicable resources. The focus here is on separating authorization and authentication, focusing on the interoperability of different services, specifically web applications. A key feature here is every organizational body is in charge of deprovsioning an identity. Although M. Gaedke and all's research focuses on the granting of permissions more than governance. This paper will focus on supplying a strategy that will aim to be a practical piece in a deprovsioning strategy.

Sebastian Kruk et al. [3] tried to address a similar scalability problem with a distributed IAM system focused on social networks. Establishing identity and granting access was based on distance calculations from direct relationships. They used a social network model and attached a numerical value to trust between direct friendships. Degrees of separation were taken into account when calculating the authenticity of an identity relative to a resource. This may make deprovisioning hard. This attempts to solve a scalability problem in managing identities for both users and orgs. Deprovisioning seems difficult as not much discussion is focused on how permissions are managed when users leave or leave and come back. It isn't clear what happens to the relationships when users leave the network, will they be remembered or not? There is a common pattern in a several papers that discuss

IAM strategies. The focus is always security and most of the time is aimed at the first part of the lifecycle authentication and granting authorization.

The use case this research will focus on will be when a user leaves and comes back. This seems like a good way to try and simulate how these other strategies will react and in some cases how these new strategies can work as an additional piece to an already implemented IAM infrastructure.

III. Motivating Example

TABLE I
IAM hierarchy for company ABC.co

Member	**Bindings**	
	Resource	***Permissions***
jack@abc.co	storage_bucket_alpha	‘WRITE’, ‘READ’
	storage_bucket_archive	’READ’
jill@abc.co	Network_configs	’READ’
	storage_bucket_alpha	’READ’, ’WRITE’, ’DELETE’
	storage_bucket_archive	’READ’, ’WRITE’
john@abc.co	storage_bucket_alpha	’READ’
	storage_bucket_archive	’READ’, ’WRITE’

TABLE II
IAM hierarchy after freeze

Member	**Bindings**	
	Resource	***Permissions***
jack@abc.co	storage_bucket_alpha	‘WRITE’, ‘READ’
	storage_bucket_archive	’READ’
jill_frozen@abc.co	Network_configs	’READ’
	storage_bucket_alpha	’READ’, ’WRITE’, ’DELETE’
	storage_bucket_archive	’READ’, ’WRITE’
john@abc.co	storage_bucket_alpha	’READ’
	storage_bucket_archive	’READ’, ’WRITE’

Number of active members: 2
Number of resources being bound to active members: 2
Number of frozen members: 1
Number of resources being bound to frozen members: 3

Considering a basic IAM structure like Table I as an example, we'll walk through some observations of a few options that might be taken. Assuming Jill must leave the company there are two options that a company may employ to handle deprovisioning, both having the same net effect. The first option is to freeze the account, This results in the bindings staying in the hierarchy. So the IAM structure would look like Table II.

The reason for freezing an account is usually because it is convenient and allows the user's resources to remain intact. The bindings that are left behind can also act as a map to see what permissions were necessary for a user to accomplish their tasks.

This does however cause problems if a user were to come back, and they were not in the same position. Even if they were there might be valid reasons for them not to start off with the same amount of access that they had when they left.

The second option would be to remove her account instead of just freezing. In this case we lose the ability to map out her permissions conveniently. The benefits of freezing or convenience of it may still cause orgs to opt for that method.

As can be seen in Table II there is a lingering access issue, and the potential of overprovisioning, if a user were to return.

IV. Empirical Evidence

TABLE III
IAM hierarchy after account removal

Member	**Bindings**	
	Resource	***Permissions***
jack@abc.co	storage_bucket_alpha	‘WRITE’, ‘READ’
	storage_bucket_archive	’READ’
john@abc.co	storage_bucket_alpha	’READ’
	storage_bucket_archive	’READ’, ’WRITE’

Number of active members: 2
Number of resources being bound to active members: 2

TABLE IV
IAM Escrow after archiving account

Member	**Bindings**	
	Resource	***Permissions***
jill_frozen@abc.co	Network_configs	’READ’
	storage_bucket_alpha	’READ’, ’WRITE’, ’DELETE’
	storage_bucket_archive	’READ’, ’WRITE’
	archived_ts	2019-08-01T21:09:33

Number of archived member permission sets: 1
Number of resources in archived sets: 3

TABLE V
IAM hierarchy after permission merge from escrow

Member	**Bindings**	
	Resource	***Permissions***
jack@abc.co	Network_configs	’READ’
	storage_bucket_alpha	‘WRITE’, ‘READ’ ’DELETE’
	storage_bucket_archive	’READ’, ’WRITE’
john@abc.co	storage_bucket_alpha	’READ’
	storage_bucket_archive	’READ’, ’WRITE’

Number of active members: 2
Number of resources being bound to active members: 3

A. IAM Escrow Proposal

One possible alternative would be to implement an option of archiving permissions. Freezing an account would archive permission sets and remove them from the IAM system, shown in Table III. If we were to archive Jill's access instead we would have still be able to map out her access to review later. A timestamp of when the set was archived is also stored. The timestamp would be appended to a member name if another snapshot is stored to allow the storage of different states for the same member. By separating the archived permissions we remove any lingering access that Jill's account may have had.

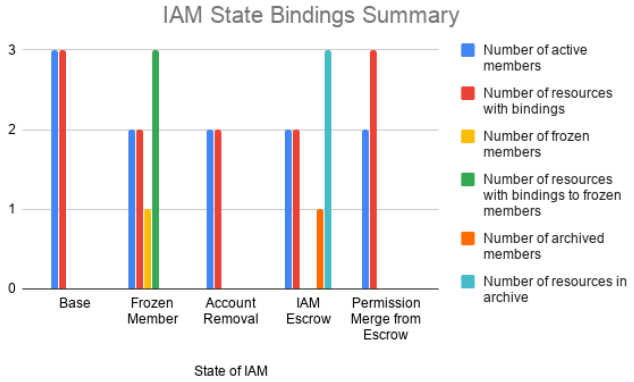

Fig. 1. Summary of all tables

If Jill rejoins company, she must get re-added, but will have no permissions tied to her. If needed escrow can be used to merge permissions back in. Let's say Jack needs to take over a service Jill previously managed. The IAM escrow structure can be used to review what access Jill had. If we assume Jack must gain all access Jill had. Escrow can be used to see bindings and merge access from archive with current active user. This would result in the IAM hierarchy shown in Table V.

Archiving permissions allows the org to keep the benefits of freezing by just adding another step to their implementation. This combines the benefits of a full account removal because the bindings are stored outside of the IAM hierarchy that is in use. It also provides a way to store snapshots of bindings, which can be merged in therefore having a functional use. Tools and logging exist that provide the same functionality of keeping an audit trail of permission bindings [4]. This is meant to act as a functional supplement. The goal was to combine the benefits of two commonly used options to try and eliminate unsafe conditions such as over-provisioning and lingering access. A summary of the states can be seen in Fig. 1.

B. *Criticism of IAM Escrow Implementation*

This implementation does have some drawbacks that should be considered. Granular control is needed over which permissions to add. Merging in from escrow may also cause overprovisioning, maybe the archive should be read-only.

Merging permissions in is one-way, so an account may gain permissions, but it may be difficult to remove permissons that were not needed. For this case archive bindings of the active account before merging in permissions, then use that snapshot to overwrite the permissions.

Terraform [5] is a tool for versioning and changing infrastructure. It can address similar issues. Such as when terraform is used to provision accounts.

Commenting them out will cause terraform's service accounts to destroy bindings. When combined with a repository to pull from you get the benefits of a change-history.

There is still the Issue of when accounts are frozen and bindings are left in terraform. When an account is frozen the underlying identifier is also changed. The bindings won't be able to be destroyed unless they are imported in from the frozen account and then removed.

V. Conclusion

This paper aims to provide a way to handle an issue of lingering access due to incorrect deprovisioning implementations. The strategies currently being employed in environments are done with the goal of achieving least privilege. The motivating example used here shows a use case that can help magnify the issue that an incorrect deprovisioning flow may have. There is still some future work that can be done to address the issue but lied outside the scope of this paper. IAM plans that are being researched can benefit from a focus on how to correctly deprovision access. Organizations may choose to follow a strategy that seems well defined but fail to implement a proper way to remove permissions that aligns with their goal. The solution proposed here was created to address a specific use-case and is meant to act as a proof of concept. IAM is a powerful component of any infrastructure, and is gaining use through increasing cloud adoption rates [6]. The process of deprovsioning access is something that is not focused on and may seem trivial. As more options are indtroduced for cloud frameworks, deprovsioning may also get more complex and issues like the ones mentioned in this paper may grow.

Acknowledgment

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.

References

[1] A. M. Al-Khouri,"Optimizing identity and access management (IAM) frameworks," International Journal of Engineering Research and Applications, pp. 461–477, 2011.

[2] M. Gaedke, J. Meinecke, M. Nussbaumer, "A modeling approach to federated identity and access management,", Special interest tracks and posters of the 14th international conference on World Wide Web ACM, pp. 1156–1157, 2005.

[3] S. R. Kruk, S. Grzonkowski, A. Gzella, T. Woroniecki, H. C. Choi, "D-FOAF: Distributed identity management with access rights delegation,", Proceedings of the Asian Semantic Web Conference, 2006.

[4] The Forseti Security Authors, "Forseti Security,", 11-Jun-2019 [Online] Available: https://forsetisecurity.org, [Accessed: 30-Jul-2019].

[5] Hashicorp, "Terraform," Terraform, 16-Mar-2018. [Online]. Available: https://www.terraform.io/. [Accessed: 13-Aug-2019].

[6] Gartner, "Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17.5 Percent in 2019," Gartner, 02-Apr-2019. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2019-04-02-gartner-forecasts-worldwide-public-cloud-revenue-to-g. [Accessed: 12-Aug-2019].