Final PhD topic” **Secure API Access Control Models For cloud computing environments”**

Proposed PhD Problem - Privileged Access Management for API and CLI interactions with Cloud Platforms

1. Privileged Access Management (PAM) or Privileged Access Governance (PAG) is one of the key focus solution areas in the world of Information Security, which helps enterprises gain protection against Insider Attacks. Therefore, not only PAM is important for securing enterprise IT operations, but also a mandate for adhering to regulatory compliance standards.
2. For a considerably long time, companies like CyberArk and [few others](https://solutionsreview.com/identity-management/privileged-access-management-solutions-directory/) have been key players in this market. Following videos (and many more available on youtube) describe PAM with a lot of practical details.

<https://www.youtube.com/watch?v=J9L8ttRw5UM>

<https://www.youtube.com/watch?v=AeMalNggPZU>

1. For past decade, PAM has been treated as a solution for only those scenarios when internal technical professional employees (System Administrators, DBAs) need to interact with datacentre servers (Unix boxes, Oracle databases) to execute patching, upgrade kind if activity on the native OS , databases or other installed applications. PAM scenarios during these interactions and also available solutions have matured a lot.

However, with the wide adoption of Cloud platforms like AWS and Azure and the need to have a [completely automated DevOps pipeline](https://www.youtube.com/watch?v=3eiYRPeNKMs), the interactions that technical professionals have with PaaS, SaaS and mainly IaaS platforms, have undergone drastic changes.

* Enterprises have started using machine instances available thru IaaS platforms (AWS, Azure) instead of hosting machines in their datacentre.
* The need for automation demands technical professionals to use a lot of scripts and Command Line Interactions (CLI) with cloud platforms.([Here](https://www.youtube.com/watch?v=iC8zVT5r7Jw) is a video on AWS CLI). CLIs internally use APIs exposed by Cloud platforms. And for any operations that need to be automated and scripted, for which CLI is not available, professional themselves invoke available APIs (Mostly [REST](https://www.youtube.com/watch?v=qVTAB8Z2VmA)). AWS EC2 APIs are described [here](https://docs.aws.amazon.com/AWSEC2/latest/APIReference/making-api-requests.html).

Even the administrative actions which technical professionals have with cloud platforms using browser UI based cloud consoles, internally invoke relevant underlying APIs.

* With most of the privileged interactions now being coordinated through APIs (including CLI and Console), it is important for PAM solutions to:
  + Control which user can be granted with right Privileged role/account. User should request (but within a time limit), right stakeholder should approve. Alternately, the user can get access without approval, in the case of access required for emergency operations, as indicated in the following references.

<https://www.beyondtrust.com/resources/webinar/break-glass-theory-designing-break-glass-process-provide-security-privileged-accounts/>

<https://hipaa.yale.edu/security/break-glass-procedure-granting-emergency-access-critical-ephi-systems>

<http://security-architect.com/how-to-balance-assurance-and-availability-in-pam-systems/>

* + Automatically revoke the granted privileged access when the time limit is over.
  + Closely Monitor the usage of what actions user carried out during the time when she used the underlying systems using the granted privileged access.

1. Proposed PhD work should focus on the following:
   1. Highlight gaps from PAM capabilities standpoint, in the current API Security models that only focus on authentication, authorization, encryption, signatures and other threat attacks like malwares, DOS etc.
   2. Propose and demonstrate a solution based on the red font text above. Use AWS Cloud platform as the base implementation of the solution. Following references will help in getting started.

AWS Security - <https://www.youtube.com/watch?v=UqKWHZ36yEM>

AWS CloudTrail - <https://www.youtube.com/watch?v=4eadWdLL9So>

AWS IAM concepts plus discussion about Privileged Access (using AWS IAM Roles)and its Monitoring (using AWS Cloudtrail) - <https://www.youtube.com/watch?v=DyHrppoArKQ>

DB Admin:

DB upgrade

IT Admin:

Unix patching

M2::DB

AWS RDS

Javed

Browser

HDFC bank**in AWS cloud**

Website

M1::Unix

AWS EC2 VM

Red Arrows:: Privileged Action, if used wrongly, can bring down the system/website.

Priv access management tools. Cyberark.

1. IT Admin, 5 PM, he has to upgrade Unix. Slightly before that, he will request Priv Access from the tool. Tool 🡪Req Grant, 5: 30, will take it back.
2. Monitor all actions and commands….record, between 5 to 5:30.

With Cloud…..

DevOps

Applications, oble apps, websites…….companies are building fast…..website update..2 days…..

Priv Actions…no more manual. **Heavy automation**.

Scripts, tools…..lot of automation software…..

These tools 🡪 invoke AWS API -🡪 carry out any Privileged Action (machine login…..copy package in directory, …server restart)

WORK TO BE DONE.

1. AWS account
2. AWS study
3. AWS EC2 VM. Website deploy.
4. Write REST API code to invoke AWS API 🡪 EC2 VM. API should login to AWS using AWS Privileged role.
5. Write code for Priv Access Management solution…..
   1. Control which user can be granted with right Privileged role/account. User should request (but within a time limit), right stakeholder should approve. Alternately, the user can get access without approval, in the case of access required for emergency operations, as indicated in the following references.

<https://www.beyondtrust.com/resources/webinar/break-glass-theory-designing-break-glass-process-provide-security-privileged-accounts/>

<https://hipaa.yale.edu/security/break-glass-procedure-granting-emergency-access-critical-ephi-systems>

<http://security-architect.com/how-to-balance-assurance-and-availability-in-pam-systems/>

* 1. Automatically revoke the granted privileged access when the time limit is over.
  2. Closely Monitor the usage of what actions user carried out during the time when she used the underlying systems using the granted privileged access. (CLOUD TRAIL LOG READ)