How to impress your security team in the cloud

Audun Solemdal

solom

The problem(s)

- In either a public or private organization, a new project is started or is planning to move to the cloud
- If there is a dedicated security team in the org, they are not necessarily cloud specialists
- You team might have a solution architect, but their focus is creating a conceptual architecture, not adhering to security best practices

The problem(s)

- The IT Ops team won’t always have competence or time to help you
- Both the budget and time schedule are tight.
- Who does the project manager expect to establish and ensure the solution is secure..?

The problem(s)

- More often than not, one or two people from the developer team take responsibility for establishing the cloud infrastructure
- In cloud migration projects you might be the only developer...
- **Sounds familiar?** This presentation is for you!

About

- Audun Solemdal
- Daglig leder @ solom
- Cloud Consulting
  - Azure, Github, containers
- Whatever needs improvement...

solom.no

solomno / audunsolemdal

audun-solemdal

Agenda

- Security concerns
  - General Azure configuration
  - Github configuration
  - Identity & RBAC
  - Networking

To be clear

- Cloud security is a wide subject
  - As a developer you should not be expected to resolve every issue

General configuration

- In the world of chess, players at the highest level rarely make blunders
  - Yet, after a few inaccuracies, they may get check mated

- In the world of cloud computing, blunders are more likely to happen
  - After a few inaccuracies, we may get «check mated»

General configuration

- When creating cloud resources, most settings are set implicitly, regardless of creation method
- The default settings may change over time, and can vary based on which tooling or version of the tooling you are using
- Once the implcit settings are set, they are usually never* changed

Quick glance at resource specs

- Example – Azure Storage account
  - Microsoft.Storage/storageAccounts - Bicep, ARM template & Terraform AzAPI reference | Microsoft Learn

Filtering the noise

- Microsoft Security baseline
  - Azure security baseline for Storage | Microsoft Learn
- Github Actions scans
  - Checkov
  - microsoft/security-devops-action

Checkov + Github Actions example

0 Open ✓ 5 Closed Closed as ▾ Language ▾ Tool ▾ Branch ▾ Rule ▾ Severity ▾ Sort ▾

Ensure storage account is configured with private endpoint Error Test main
#5 closed as fixed last year • Detected by checkov in platform/test/cloudshell.tf :33 last year

Ensure 'Trusted Microsoft Services' is enabled for Storage Account access Error Test main
#4 closed as fixed last year • Detected by checkov in platform/test/cloudshell.tf :33 last year

Ensure that Storage accounts disallow public access Error Test main
#3 closed as fixed last year • Detected by checkov in platform/test/cloudshell.tf :33 last year

Ensure Storage Account is using the latest version of TLS encryption Error Test main
#2 closed as fixed last year • Detected by checkov in platform/test/cloudshell.tf :33 last year

Ensure that Storage Accounts use replication Error Test main
#1 closed as fixed last year • Detected by checkov in platform/test/cloudshell.tf :33 last year

Filtering the noise

- Azure Policy
  - Not developer-centric, but works great
  - Can be used to audit or explicitly deny settings, among other things
  - Some policy initatives should ideally be set up at the org level

Azure Policy example

Name ⇅	Effect type ⇅	Version (prev
Local users should be restricted for Storage Accounts	Deny	1.*.*
Allowed Copy scope should be restricted for Storage Accounts	Deny	1.*.*
Storage Accounts should use a container delete retention policy	Deny	1.*.*
Storage accounts should prevent cross tenant object replication	Deny	1.*.*
Storage Accounts with SFTP enabled should be denied	Deny	1.*.*
Virtual network rules should be restricted for Storage Accounts	Deny	1.*.*
Storage accounts should prevent shared key access	Deny	2.*.*
Storage accounts should restrict network access using virtual network rule	Deny	1.*.*
Storage accounts should restrict network access	Deny	1.*.*
Network ACL bypass option should be restricted for Storage Accounts	Deny	1.*.*
Storage Accounts should restrict CORS rules	Deny	1.*.*
Resource Access Rules Tenants should be restricted for Storage Accounts	Deny	1.*.*
Resource Access Rules resource IDs should be restricted for Storage Accou	Deny	1.*.*
Storage accounts should be migrated to new Azure Resource Manager re	Deny	1.*.*

Filtering the noise

- Infrastructure as code
  - Choose the language the org wants, if no preference – set the standard
  - Use standardized modules (ideally standardize at the org level)
  - Azure Verified Modules

Github config

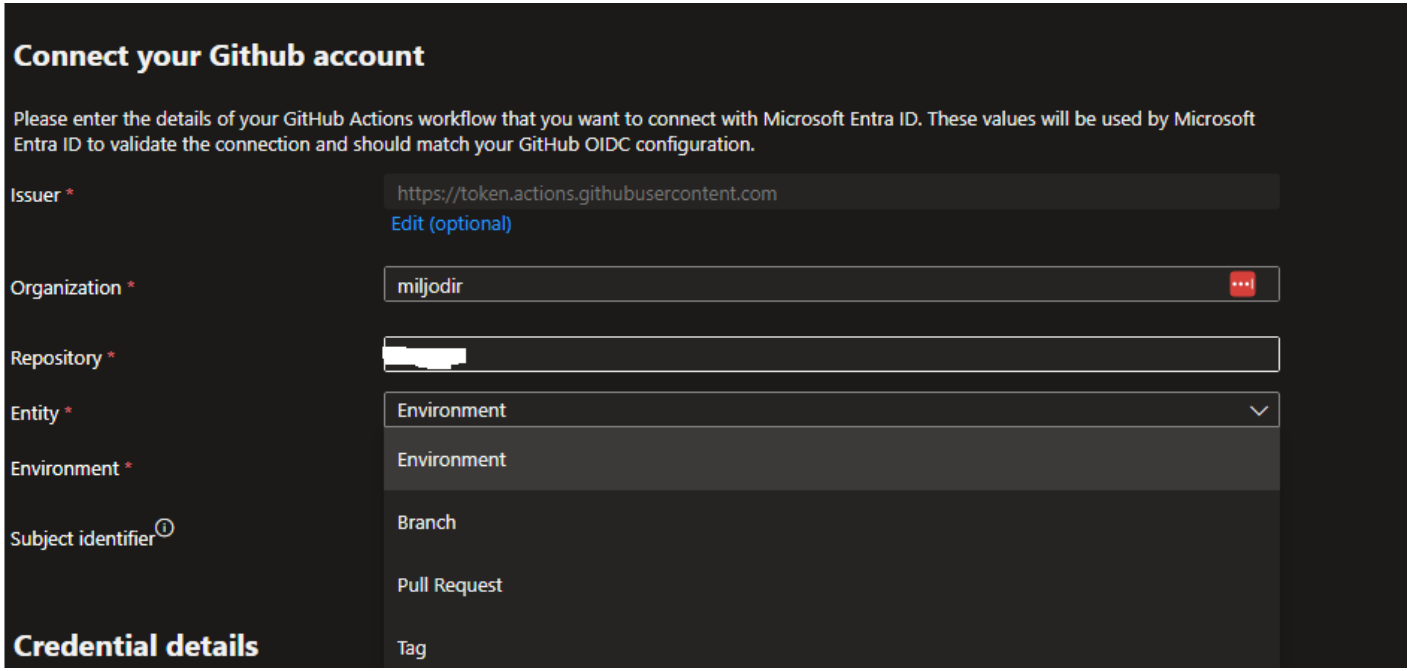
- Github
  - RBAC support is not the best, seems to be improving
  - Secrets are not shown in clear text – but can be manipulated
- GH Advanced Security is good – but costly
  - Push protection with custom patterns
  - Free on public repos
- Github Actions
  - Don’t trust every action blindly
  - Consider explicitly setting job permissions
  - Use tagged versions or commit SHAs

Github config

- Repository rulesets
  - Rapidly developing
  - Superseeds branch protection
  - Can protect branches or tags by pattern
- Environment protection
  - Support «gated» deployments

Github config

- OIDC
  - Default logic is repo + context, can be changed via API



Identity

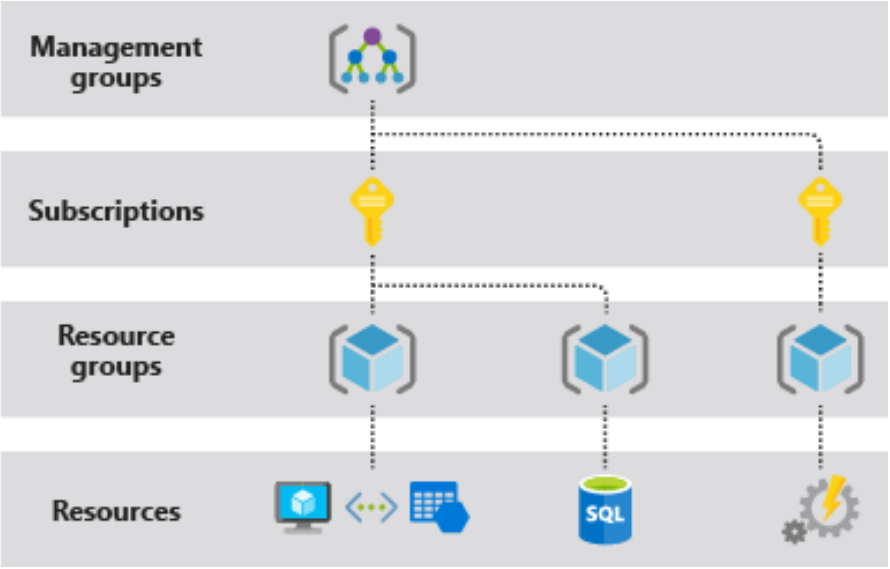
- Identity is a wide topic
  - We are focusing on the parts where developers can make a difference

Quick Azure RBAC recap

- Access is built on ARM which uses resource providers
  - E.g. Microsoft.Web, Microsoft.Storage, Microsoft.Compute
- Control plane & data plane
  - Actions, NotActions – control plane
  - DataActions, NotDataActions – data plane
- Role assignments can grant RBAC roles to principals
  - Examples - Entra ID groups, users, managed identitites

Quick RBAC recap

- Role assignments can be granted at different scopes and are inherited in a hierarchy



Common RBAC handling

```
// This can also apply at resource group level depending on your governance model
private void ReduceStressLevel(string devTeam, string? leadDeveloper)
{
    if (devTeam.IsCryingAboutAzurePermissions())
    {
        try
        {
            return GrantSubscriptionAccess(devTeam, "Contributor");
        }
        catch (StillCryingAboutAzurePermissionsException)
        {
            return GrantSubscriptionAccess(leadDeveloper, "Owner");
        }
    }
}
```

Common RBAC handling

- Contributor role
  - Used because it is a practical «catch-most» issue handling
    - Grants direct permissions to the control plane **only**
    - Grants «back door» access to the data plane
  - Some unfortunate accesses most principals **shouldn't** have
    - PaaS – often involves access to admin credentials towards data plane
    - IaaS – reset admin password / SSH key / run command
    - CRUD to any* Azure resource. Is this really required?

Authenticating with the data plane

- Nearly all Azure PaaS services support using keys for authentication
  - Contributors have access to these keys.
  - These keys are often used in app code (with or without Azure Key Vault)

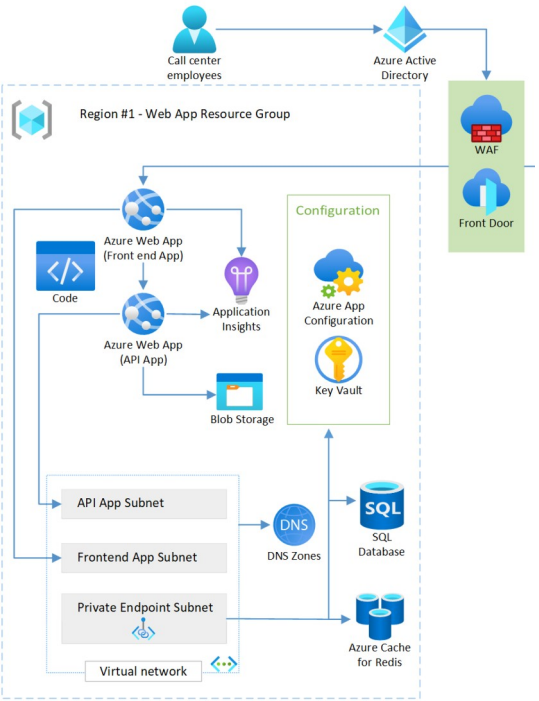
- Keys should ideally be removed or disabled
  - The best secret is the one which doesn't need to exist
  - Compute services can be assigned one or more managed identities
  - Use managed identites in your code whenever possible.
  - Assign RBAC roles with the required data plane permissions

Azure DevOps service connections…

- Check if your Sub / RG IAM looks something like this
  - This can now easily be improved via a few clicks
  - Also consider manual assignment to reduce the permissions

App	Contributor	Subscription (Inherited)
App	Contributor	Subscription (Inherited)
App	Contributor	Subscription (Inherited)
App	Contributor	Subscription (Inherited)
App	Contributor	Subscription (Inherited)
App	Contributor	Subscription (Inherited)

Access keys and Action roles



- How many keys and «Action-centric» role assignments are required for your app and developers?
- 0 Action-centric roles, 1 key (App Insights connection string)

Least privilege for the API

- Azure RBAC roles
  - Storage Blob Data Contributor / Reader
  - Monitoring Metrics Publisher + connection string
  - Key Vault Secrets User
  - App Configuration Data Reader
- Service-specific roles
  - SQL database
    - "CREATE USER [my-app-name] FROM EXTERNAL PROVIDER; ALTER ROLE(..)»
    - Also possible to grant Entra admin at server level if feeling frisky..
  - Redis cache - «Data Contributor» or custom access policy

DefaultAzureCredential

- Typically you only need to change this part of your app code

```
// Before
using Azure.Messaging.ServiceBus;

var client = new ServiceBusClient(connectionStringWithSecret);
client.CreateSender(queueName);

// After
using Azure.Messaging.ServiceBus;
using Azure.Identity;

var client = new ServiceBusClient(connectionStringWithoutSecret, new DefaultAzureCredential());
client.CreateSender(queueName);
```

OIDC

- DefaultAzureCredential
  - ManagedIdentityCredential
  - AzureCliCredential
  - ChainedTokenCredential
- WorkloadIdentityCredential
  - Kubernetes, Github, Azure DevOps ++
- Same principle can work against other Microsoft services
  - Microsoft Graph

Practical implementation in one slide

- Strongly consider IaC + Git to make this manageable!
- Grant needed roles access to groups and managed identities only
- Disable all keys / passwords whereever possible

- Suggested role assignments
  - Create one or more Entra ID groups with users
    - Assign roles based on needs

- Managed Identity in Github Actions needs Owner or permissions close to it
- Other Managed Identities - assign minimum role assignments required

Network

- Network is a wide topic
  - And can become very advanced very quickly
  - We are focusing on the parts where developers can make a difference

Networking

- In an perfect world, only your application and a minimum number of people can reach your services

Networking

- Integrate your compute services into Virtual Networks
- Azure PaaS based services support local firewall rules for the most part. Use them!
  - Allow traffic from selected subnets + specific public IPs
  - Or allow traffic via private endpoint + Virtual Desktop / Jumpbox
- Manage network rules as code in a Git repo for tracking

Networking

- App Gateway with Azure Web Application Firewall (WAF)

- Blocks malicious traffic against your web apps & APIs
  - Managed rulesets, custom rules

- Cons:
  - Costly to set up for a single project
  - Config can be complicated even when managed in code
  - Effort needed to avoid false postives

CI/CD networking

- Github and Azure DevOps support simple and cheap Azure VNET integration
  - Consider setting it up if needed

Thank you for attending!

Audun Solemdal

solom

audun-solemdal

solomno/sharing

solom.no