Azure for Students

Overview on Azure and their services (kinda ranked after use-cases)

1. Compute: VMs (IaaS), Container Instances (virtualised Applicationenvironments), App Service, Functions
2. Networking (VPN,...)
3. Storage
4. Mobile services (apps,...)
5. Databases
6. Web (Web apps, ...)
7. IoT (manage all IoT assets, ...)
8. BigData (cluster services, ...)
9. AI services and Machine Learning
10. DevOps

Notes

- CA from my static website: Microsoft Azure TLS Issuing CA 01
- Easy to differentiate between VM and Containers: VM virtualise the hardware, containers the OS
- `azurewebsites.net`, `azurestaticapps.net` as domain ending
- Static and web apps can be set up with the help of Git
  → learning something via Git Accounts?
- Possibilities to secure Azure: SecurityCenter, Sentinel, AzureDefender

To be checked out:

`https://sonraisecurity.com/blog/attackers-find-aws-s3-bucket-with-17m-users/`

Most common mistakes with Azure

- **Misconfiguration of Roles & Administration:** [1] [2] [3]

  making everyone as (co-)administrator in the Azure subscription. Team members often times remove or delete the Azure components, which creates lot of impact on overall application and ruins the stability of the environment. Members also unknowingly keep on provisioning Azure resources for research and evaluation purpose which causes a huge expenditure. Data Storage Access Misconfig: a user can set permissions that expose data to the entire internet in "Azure Storage" **(try to do that!)**

  → provide users only with the amount of permission they need to do their job!
  RBAC (Role Based Access Control) or ARM (Azure Resource Manager) or MFA (Multi-Factor auth.)

- **Choosing incorrect specifications (of Azure VM) & not securing access points** [1] [4]
  All aspects like overall user load, nature of the application and geography of the users must be considered by enterprises and individuals hosting their applications, before designing the infrastructure. Not securing access points to the clouds, allowing users to access the VM from any machine, anywhere.

- **Billing of over usage** [1]
  They keep running their Azure instances, even if they are not in use or when purpose is served.
  → keep resources in off-state or delete them once you no longer need them.

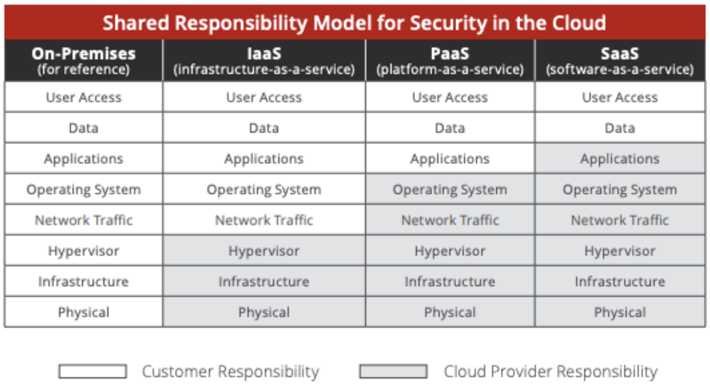
- **Weak, mismanaged Passwords:** [2] [4]
  Improper password management and bad password habits. Microsoft reports 10 million username/password pair attacks per day.

- **Misconfig/not enabling of Security/Managing Controls:** [2]
  Failed to turn on the logging feature. Necessary to permit access visibility but also to see who is accessing and managing the subscription. Failing to enable Azures security center and its native security tools (**check this out!**)
  TODO: subnets should not be assigned to a public IP that could open unwanted ports. (Network Security Groups (NSGs)) NSGs control access by permitting or denying network traffic via communication between different workloads on a vNET, network connectivity from on-site environment into Azure, or direct internet connection.

- **Lack of Oversight / Security Monitoring:** [2] [3]
  Missing ongoing management and security (often times just in the beginning and then not covered anymore) Security vulnerabilities result when Azure users don't

[1] https://blog.e-zest.com/here-are-10-most-common-mistakes-while-managing-azure-cloud
[2] https://www.lightstream.tech/top-5-azure-mistakes-your-security-team-is-making/
[3] https://www.viacode.com/most-common-azure-security-problems/
[4] https://techgyo.com/5-common-microsoft-azure-security-mistake

understand what they are responsible for and the tools and services Azure provides to help them.



Shared Responsibility Model for Security in the Cloud

On-Premises (for reference)	IaaS (infrastructure-as-a-service)	PaaS (platform-as-a-service)	SaaS (software-as-a-service)
User Access	User Access	User Access	User Access
Data	Data	Data	Data
Applications	Applications	Applications	▨ Applications
Operating System	Operating System	▨ Operating System	▨ Operating System
Network Traffic	Network Traffic	▨ Network Traffic	▨ Network Traffic
Hypervisor	▨ Hypervisor	▨ Hypervisor	▨ Hypervisor
Infrastructure	▨ Infrastructure	▨ Infrastructure	▨ Infrastructure
Physical	▨ Physical	▨ Physical	▨ Physical

□ Customer Responsibility ▨ Cloud Provider Responsibility [5]

- **Cloud misconfiguration:** [3]
  Especially the misconfiguration of databases and object storage services. (linked there - https://www.mcafee.com/enterprise/en-us/assets/skyhigh/white-papers/cloud-adoption-risk-report-2019.pdf)

- **Not encrypting data at rest:** [3]
  blobs are encrypted by default, but VM disks are not, creating vuln. User have to enable disk encryption on their own (for free!).

- **Not managing patches correctly:** [4]
  Not applying a patch at all or do it only half the way.

- **Not testing security or taking it for granted:** [4] [1]
  Users should use penetration testers (microsoft has a policy governing such tests). Even though Microsoft secures the platform, it can't protect against e.g. weak passwords

- **Not opting for Microsoft support option** [1]
  To save costs, companies try to avoid the support when purchasing the Azure subscription. But it can happen, that even a high skilled employer isn't able to e.g. shut down a VM. The support however can do this. If enterprises have not chosen the appropriate support option then there might be a huge impact.

- **McAfee Report** [5]
  More and more files contain sensitive information and too many are publicly accessible. Enterprise organizations have an average of 14 misconfigured IaaS/PaaS instances.

[5] https://www.mcafee.com/enterprise/en-us/assets/skyhigh/white-papers/cloud-adoption-risk-report-2019.pdf

Azure Key Vault

Search for `.vault.azure.net` websites (e.g. Dorking: `site:*.vault.azure.net/`)
→ no results, aren't listed...

Resolve the key vault uri (`https://super-secure-keyvault.vault.azure.net/`, `https://aiskeyrun.vault.azure.net`)
→ runs on `51.116.154.67` and `ASN 8075`

Azure SecurityCenter

All security recommendations:
`https://docs.microsoft.com/en-us/azure/security-center/recommendations-reference`

(Most) Common security recommendations:

- Enable MFA
- Secure (management) ports
  (JIT network access control should be applied on VMs, VMs associated with a NSG, Management ports should be closed on your VMs)
- Apply system updates
  (60 % of security breaches caused by vuln's which are already patched)
- Remediate Vulnerabilities
  (Vulnerabilities in ... should be remediated / Vulnerability assessment should enabled on ...)
- Enable encryption at rest
- Encrypt data in transit
  (... should only be accessible over HTTPS / Only secure connections to ... should be enabled)
- Access management and permissions
  (RBAC, permissions should be removed from your subscription)
- Manage network access
  (CORS should not allow every resource to access your ..., Access should be restricted, Remote debugging should be turned off, IP forwarding should be disabled)
- Apply DDoS protection
- Enable endpoint protection, monitoring agent & logging
  (Monitoring agent should be installed, Diagnostic logs in ... should be enabled)

Azure Storage

- ”Data Storage Access Misconfig: a user can set permissions that expose data to the entire internet in Azure Storage”
  → Try to leak something. Can you see that from the outside?

Azure Blobs consists of an account and container name with the scheme:
`https://{accountName}.blob.core.windows.net/{containerName}`

To find open accessible blobs: Google dorking with `site:"*.blob.core.windows.net"`
For secondary endpoints: `site:"*-secondary.blob.core.windows.net"`

Possible access rights for blobs: `private, blob, container`

- Container: `https://unprotectedblob.blob.core.windows.net/leakedfiles/Beach.png` is openly accessible
- Blob: `https://unprotectedblob.blob.core.windows.net/stillleakingfiles/harold.jpeg` is also openly accessible
- Private: `https://unprotectedblob.blob.core.windows.net/protectedfiles/street.jpg` is protected

Azure Functions

`https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview`

- Create functions app with wrong permission and see what you can see.

Finding publicly available Azure Functions

The ”vulnerability” for HttpTriggered Functions is to have the `"authLevel"` set to `"anonymous"` in the `function.json`.
If this is the case, everyone can execute the function.
Examples (for some reason, sometimes the site needs to be reloaded multiple times):

- `https://tryingtogetaccessfromeverywhere.azurewebsites.net/api/HttpTrigger1`
- `https://tryingtogetaccessfromeverywhere.azurewebsites.net/api/HttpTrigger1?name=philipp`

Scheme: `https://{accountName}.azurewebsites.net/api/{functionName}`

However, given an `azurewebsites.net` (e.g. find them via Google dorking: `site:"*.azurewebsites.net"`), we can look for `/api/ + <existing functionnames>` (default one for HttpTriggered: `HttpTrigger1`).

For this project we have to know which `azurewebsites.net` belong to which company.

Otherwise, we just check for unprotected AzureFunctions.
Each Azure Function has an own Blob storage with at least two containers called `azure-webjobs-hosts, azure-webjobs-secrets`.
The blob storage is then called `"storageaccount<first 5 letters from resourcegroup> <4 numbers/digits>"`
If someone totally screwed up the settings (i think you can only achieve this with intention) you can find the azure function container via
`https://{storageaccount ...}.blob.core.windows.net/{containerName}/ {resourceGroupName}/<file>`. However, we could use dirbuster or crawl to see containers.

Further ideas

- `https://github.com/cyberark/BlobHunter`
  Looks like they loop over all existing blob storages in the given account to check if the access level is public.
  They get access from the user of the program so they don't analyze from the outside.

- `https://ninocrudele.com/the-three-most-effective-and-dangerous-cyberattacks-to-a`

- `https://github.com/0xsha/CloudBrute`
  Insights so far:

  - Checks the domain for used Cloud providers in the HTML
  - If one is used they check for existing urls, starting with the given keyword concatenated with a dictionary (`storage_small.txt / storage_large.txt`).
    E.g. when Azure is used, they search for
    `<keyword><dictionarywords>.<azureservice>.core.windows.net`

  Seems like we could even extend this by looking for e.g. `.azurewebsites.net`, `.vault.azure.net`

Azure DNSZone

Nothing to be seen or learned here.
Just creating basic DNS entries or setting an IP *(querying the DNS entries is an old finding)*

azurewebsites.net

Organization Identifiable Information (OII) and Personally Identifiable Information(PII) is what we are searching for on Azure websites

- Can you link `aistestresources.azurewebsites.net` back to AIS? Via DNS/cert./the webpage?

Certificate: On the first look, you don't see anything. Analyzing the cert of `aistestresources.azurewebsites.net` via different tools and solution, you can't trace back the website to `ais-security.de`

DNS and the rest is also a dead end

- Can you link an `azurewebsites.net` url back to a subscription ID or related blob storage or anything else?
  - By checking DNS you get the location of the server which was selected in Azure
    E.g. `tryingtogetaccessfromeverywhere.azurewebsites.net` is in Germany West-Central

Conclusion

So far, we can identify public available *Azure Functions* and *Blob Storages.* However, we cannot trace them back to a company, a subscription ID, or any other OII or PII.
If you have given a company name, you could try to find something via Google Dorking (e.g. `allintext:ais security [more possible keywords here] site:"*.azurewebsites.net"`), however this can't really be automized.