

# Pilvitietoturvaa

Juho Myllylahti 5.11.2019



# Moro!

- Perusfaktat: Juho Myllylahti aka. Mutjake, töissä Solitalla
- ...ja ajattelin jutustella yleisesti (julki)pilven tietoturvasta
- Disclaimer: olen pääasiallisesti AWS-heebo, joten esimerkit jne. urautunevat siihen suuntaan; GCP:stä, Azuresta jne. tiedän toistaiseksi vielä vähemmän (WIP)
- Slidet eivät ole mitenkään kaikenkattavat, lähinnä kaivelin vähän satunnaisia asioita esille ja tarjolle herättämään ajatuksia ja antamaan kuvaa pilvitietoturvasta



# Yleistä

- Pilvialustat ovat palveluita, joissa luottokorttinumeroa vastaan voi käyttää erinäköisiä palveluita, vuokrata elastisesti erilaisia resursseja (verkkoa, virtuaalikoneita, levytilaa jne.) ja rakentaa niiden päälle haluamiaan asioita
- Julkipilvet ensin pääosin IaaS (infrastructure as a service) -tyyppisiä palveluntarjoajia, jotka vuokraavat virtuaalipalvelimia (VPS); Hetzner, Scaleway, DigitalOcean jne.
- Toisaalta taas esimerkiksi Amazonin ensimmäinen palvelu oli PaaS (platform as a service) -palvelu Simple Queue Service, AFAIK Googlen yksi ensimmäisistä oli App Engine
- SaaS-palvelusta esimerkkinä vaikkapa GitHub, Gmail, Office 365...



# Yleistä

- Nykyään suurimpiin pilvialustoihin liittyy tärkeänä osana Infrastructure as Code -ajattelu, jossa resurssien orkestrointi tapahtuu API:en välityksellä, pay as a go -ajattelu, sekä laaja kirjo IaaS/PaaS/SaaS-palveluita joita voi tarpeen mukaan hyödyntää
- Ns. serverless -ajattelu, jossa tietyt ongelmat voidaan ratkaista käyttämällä pelkästään platform-palveluita ja jättää suoritusresurssien allokatio pilvivendorin huoleksi (aja tämä kontti/tietokanta/funktio "jossain")
- Esimerkkityökaluja: AWS:n CloudFormation, Terraform, Serverless Framework
- Elastisuus, käytännössä rajaton resurssikapasiteetti (joskin tileillä on ihan hyvästä syystä usein soft limitejä, joiden säätäminen vaatii tiketin asiakaspalveluun)

▼ All services

📁 Compute

EC2  
Lightsail   
ECR  
ECS  
EKS  
Lambda  
Batch  
Elastic Beanstalk  
Serverless Application Repository

📁 Storage

S3  
EFS  
FSx  
S3 Glacier  
Storage Gateway  
AWS Backup

📁 Database

RDS  
DynamoDB  
ElastiCache  
Neptune  
Amazon Redshift  
Amazon QLDB  
Amazon DocumentDB

📁 Migration & Transfer

AWS Migration Hub  
Application Discovery Service  
Database Migration Service  
Server Migration Service  
AWS Transfer for SFTP  
Snowball  
DataSync

📁 Networking & Content Delivery

VPC  
CloudFront  
Route 53  
API Gateway  
Direct Connect  
AWS App Mesh  
AWS Cloud Map  
Global Accelerator 

🔧 Developer Tools

CodeStar  
CodeCommit  
CodeBuild  
CodeDeploy  
CodePipeline  
Cloud9  
X-Ray

🛡️ Customer Enablement

AWS IQ   
Support  
Managed Services

🤖 Robotics

AWS RoboMaker

🔗 Blockchain

Amazon Managed Blockchain

📡 Satellite

Ground Station

📁 Management & Governance

AWS Organizations  
CloudWatch  
AWS Auto Scaling  
CloudFormation  
CloudTrail  
Config  
OpsWorks  
Service Catalog  
Systems Manager  
Trusted Advisor  
Control Tower  
AWS License Manager  
AWS Well-Architected Tool  
Personal Health Dashboard   
AWS Chatbot

📁 Media Services

Elastic Transcoder  
Kinesis Video Streams  
MediaConnect  
MediaConvert  
MediaLive  
MediaPackage  
MediaStore  
MediaTailor  
Elemental Appliances & Software

🧠 Machine Learning

Amazon SageMaker  
Amazon Comprehend  
AWS DeepLens  
Amazon Lex  
Machine Learning  
Amazon Polly  
Rekognition  
Amazon Transcribe  
Amazon Translate  
Amazon Personalize  
Amazon Forecast  
Amazon Textract  
AWS DeepRacer

📊 Analytics

Athena  
EMR  
CloudSearch  
Elasticsearch Service  
Kinesis  
QuickSight   
Data Pipeline  
AWS Glue  
AWS Lake Formation  
MSK

🔒 Security, Identity, & Compliance

IAM  
Resource Access Manager  
Cognito  
Secrets Manager  
GuardDuty  
Inspector  
Amazon Macie   
AWS Single Sign-On  
Certificate Manager  
Key Management Service  
CloudHSM  
Directory Service  
WAF & Shield  
Artifact  
Security Hub

💰 AWS Cost Management

AWS Cost Explorer  
AWS Budgets  
AWS Marketplace Subscriptions

📱 Mobile

AWS Amplify  
Mobile Hub  
AWS AppSync  
Device Farm

🕶️ AR & VR

Amazon Sumerian


📁 Application Integration

Step Functions  
Amazon EventBridge  
Amazon MQ  
Simple Notification Service  
Simple Queue Service  
SWF

📁 Customer Engagement

Amazon Connect  
Pinpoint  
Simple Email Service

🏢 Business Applications

Alexa for Business  
Amazon Chime   
WorkMail

👤 End User Computing

WorkSpaces  
AppStream 2.0  
WorkDocs  
WorkLink

🌐 Internet of Things

IoT Core  
Amazon FreeRTOS  
IoT 1-Click  
IoT Analytics  
IoT Device Defender  
IoT Device Management  
IoT Events  
IoT Greengrass  
IoT SiteWise  
IoT Things Graph

🎮 Game Development

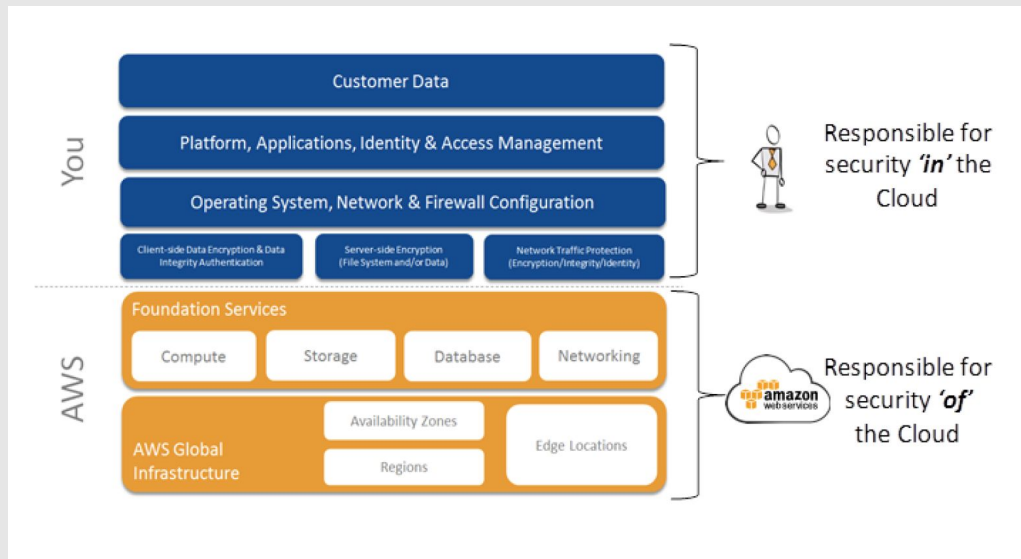
Amazon GameLift



# Shared responsibility model

Responsibility	On-Prem	IaaS	PaaS	SaaS
Data classification & accountability	Cloud Customer	Cloud Customer	Cloud Customer	Cloud Customer
Client & end-point protection	Cloud Customer	Cloud Customer	Cloud Customer	Cloud Customer
Identity & access management	Cloud Customer	Cloud Customer	Cloud Customer	Cloud Customer
Application level controls	Cloud Customer	Cloud Customer	Cloud Customer	Cloud Customer
Network controls	Cloud Customer	Cloud Customer	Cloud Customer	Cloud Customer
Host infrastructure	Cloud Customer	Cloud Customer	Cloud Customer	Cloud Customer
Physical security	Cloud Customer	Cloud Customer	Cloud Customer	Cloud Customer

Legend: Blue square = Cloud Customer, Grey square = Cloud Provider





# Suojauksen eri tasot

- IaaS-taso
  - Pääosin verkkosegmentointi, palomuurit, IP-whitelistaukset, VPN:t, ja muut perinteiset konstit ja työkalut
  - Ohjelmistokomponenttien omat työkalut, esim. tietokantapassut
  - SSH
  - Paljon samaa kuin on-premise -maailmassa
- PaaS/SaaS-taso
  - Pilvivendorin tarjoamat työkalut, usein Identity and Access Management työkalut ja -käsitteet: policyt/roolit/käyttäjät/tilit
  - Azuressa Azure AD
  - TLS&sertifikaatit, JWT, HMAC
  - API-avaimet
  - Käyttäjätilit



# Yleisiä haasteita tietoturvan näkökulmasta

- Ylläpitäjän roolin marginalisaatio: pahimmillaan devitiimille lyödään pilvitilin tunnukset käteen ja kutsutaan sitä devopsiksi
  - Layered security -meininki voi jäädä uupumaan, logitus ja muu tilannekuvatekeminen jää vähemmälle, samoin incident response -valmius ja muu opsailu
- Muutos on-premisehiekkalaatikosta jaettuun platformiin: esimerkiksi vanhan projektin DNS-osoitus johonkin poistettuun S3-ämpäriin vs. osoitus sisäiseen IP-osoitteeseen
- Pilvien API-avainten hallinta, etteivät ne päädy esim. githubiin, ylipäättään niiden arvon hiffaus
- Pilvialustojen skaalautumisominaisuudet: esimerkiksi foorumispämmin seuraukset on-prem -hostauksessa vs. autoskaalautuvassa julkipilvessä
- Paradigman muutos ylläpidossa: on-premisen työkalupakki ei välttämättä toimikaan samoin pilvessä, esim. verkkoliikenteen monitorointi; uudet opeteltavat jutut kuten IAM (Identity and Access Management); lisääntynyt automaatio sekä resurssien väliaikaisuus (esim. forensiikka ja autoskaalautuva serveri-instanssiryhmä)



# Yleisiä haasteita tietoturvan näkökulmasta

- Lisäksi on-premisessä käytettyjen työkalujen lisenssimallit voivat olla hankalia, jos pyritään moderniin tilirakenteeseen (useita pieniä tilejä, joiden laskutus on yhdistetty yhden päätilin alle): esim. laskutus voi olla per tili per kuukausi -> problem
- Resurssien avaaminen maailmalle huomattavasti helpompaa kuin on-premise -maailmassa
  - Viimeisimpänä omaan haaviin osuneena esimerkkinä tästä on julkisesti saatavilla olevien Elastic Block Storage (EBS) -levyjen sisällön tonkiminen
  - Dashboardit, tietokannat, S3-bucketit, RDS, FTP...
  - Sisäiseksi tarkoitettut API:t, jotka ovat saatavilla julkiverkosta
- Usual suspects: OWASP top 10, salasanojen rotaatio ja turvallisuus, käyttäjätilien ajantasaisuus, tilannekuva mitä tileillä pyöritetään, varjo-IT (ml. randomit firman kortilla avatut tilit) jne.



# Pilvipalveluiden recon

- Tämä lista ei ole kaikenkattava, mutta joitain esimerkkejä aiheesta
- DNS-recordeista selviää usein paljon, esim. TXT-recordit sisältävät usein domainautentikaatitokeneja palveluita varten
- <https://www.shodan.io/>, <https://buckets.grayhatwarfare.com/> jne.
- Käyttää julkisia palveluita ja katsoo, minne liikenne menee :-)
- Azuren AD saattaa olla väärin configuroitu, jolloin siitä saa enumeroitua käyttäjätilejä pihalle
  - Tsekkaa adsecurity.org & “Attacking & Defending the Microsoft Cloud” -slidet, BH 2019

# Esimerkkejä pilvihyökkäyksistä: subdomain takeover



- S3-ämpäreitä voidaan käyttää nettisivujen hostaukseen, esim. testi-niminen ämpäri
- Oma domain saadaan käyttöön tekemällä CNAME record, joka osoittaa esim. <http://beta.yritys.fi/> -osoitteen <http://testi.S3-website-eu-west-1.amazonaws.com> -osoitteeseen
  - HTTPS vaatii TLS-sertifikaatin private keyn -> ei hyökättävissä samalla tavalla
- Myöhemmin siirryttään tuotantoon ja projekti muuten päättyy ja sen resurssit tuhoetaan pilvitililtä -> testi-niminen ämpäri vapautuu jonkun muun käytettäväksi (ämpäreiden namespace on jaettu kaikkien AWS:n asiakkaiden kesken)
  - “The specified bucket does not exist” kertoo, että bucketin nimi on vapaata riistaa
- Hyökkääjä voi luoda ämpärin nimeltä testi ja laittaa sinne esimerkiksi phishing-sivuston, jonka käyttäjä luulee olevan validi, koska sen osoite on <http://beta.yritys.fi/>
- <https://github.com/EdOverflow/can-i-take-over-xyz> sisältää listan pilvipalveluista, joissa tämä hyökkäys on riski

# Esimerkkejä pilvihyökkäyksistä: liian laveat pääsyoikeudet



- Useita pilviresursseja voidaan konfiguroida avoimeksi maailmalle, jos halutaan esimerkiksi tarjota tiedosto julkisesti ladattavaksi
- Valitettavasti tämä tapahtuu helposti myös epähuomiossa tai laiskuuden vuoksi, kun “en pääse lataamaan vieläkään sitä tiedostoa”
- Samantyyppinen ongelma kuin julkiset FTP-serverit tai käyttäjäkansiot HTTP-palvelimilla + puuttuva tai liian lavea .htaccess
- <https://businessinsights.bitdefender.com/worst-amazon-breaches>
- Security by obscurity ei toimi, koska ämpäreiden nimiä voi arvailla sanakirjahyökkäyksillä ja static website hostingin päällelaitto luo DNS-entryn -> DNS:ää seuraamalla saa feedin uusista S3-bucketeista:  
<https://github.com/eth0izzle/bucket-stream>
- AWS on tuonut uusia ominaisuuksia ja työkaluja, jotka vaikeuttavat tai kokonaan estävät julkiämpäreiden konffauksen



# Esimerkkejä pilvihyökkäyksistä: SSRF

- Server-side request forgeryssä huijataan security boundaryn paremmalla puolella oleva palvelin tekemään pyyntö jotain sisäistä resurssia vasten
- Esim. blogimoottori mahdollistaa liitteen lisäyksen antamalla tiedoston URL:in, jonka backend hakee ja cachettaa liitteeksi

-> Miten olisi `file:///etc/passwd` tai `http://internal.firma.fi/`

- Useissa pilvipalveluissa virtuaalikoneiden käytössä taikaosoite, josta voi hakea hyödyllistä metadataa käyttöönsä, esim. AWS:ssä `http://169.254.169.254/latest/meta-data/` tarjoaa lokaalin IP-osoitteen ja instanssille konfiguroidun IAM-roolin AWS tokenit -> avaimet valtakuntaan, jos ko. rooli on konffattu laiskasti -> hyökkääjä voi käskyttää AWS:n apia Internetistä, esim. käynnistää EC2-koneita tai deletoida kaikki logigroupit tililtä (erillinen logging account on hyvä idea)
- Mahdollisia hyökkäystapoja mm. PDF-printterit, XML-parserit
- Jänniä URL schemeja mm. `file://` `dict://` `gopher://` `sftp://` `tftp://` `ldap://` `http://`



# Esimerkkejä pilvihyökkäyksistä: Kubernetes

- En ole ns. k-mies, mutta Internet kertoi, että kubernetesen defaultit eivät ole sieltä tietoturvasuosioista päästä
- Esim. shodanilla voi etsiä julkisia kubernetes-dashboardeja
- Suojattavia asioita myös mm. kubernetesen API ja etcd
- Käytössä usein pilvimaailmassa, joko itse asennettuna tai hostattuja Kubernetesistä käyttäen
- SSRF
- For more information tsekkaa esim:  
<https://www.slideshare.net/Lacework/practical-guide-to-securing-kubernetes>

# Miten ehkäistä riskejä?

- Kuten perinteisessä tietoturvassa: least privilege -ajattelu
  - Varaa aikaa IAM-askarteluun ja policyasetusten opetteluun, niiden tekeminen kunnolla helpottaa myös riippuvuussuhteiden ymmärtämistä myöhemmin
- Kytke päälle billing alertit (löytyvät ainakin AWS:stä)!
- Minimoi root accountin käyttö, käytännössä sillä pitäisi vain säätää laskutusasetukset ja luoda minimaaliset IAM-resurssit, joilla tekemistä jatketaan
- Älä luota security by obscurityyn, vaikka pilvi siihen houkutteleeekin
- Segmentoi tilit (AWS)/resource groupit (Azure)/projektit (GCP), harkitse esim. erillistä IAM/logitustiliä, joka pienentää blast radiusta
- Luo henkilökohtaiset IAM-käyttäjät, harkitse SSO-integraatiota
- Tutustu platformin tarjoamiin työkaluihin, kuten CloudTrail, WAF, AWS Config...
- Azure: tutustu Azure AD:n suojaukseen ja jos tili on vanhempi, tarkista tilin oletusasetusten tietoturva



# Miten ehkäistä riskejä?

- <https://github.com/toniblyx/prowler> - työkalu AWS-tilien auditointiin
- Security layers -ajattelu
  - Onions have layers, security has layers
- JWT:t/HMAC:it -- varmista että signaus oikeasti tarkistetaan
  - Monissa kirjastoissa ja esimerkeissä ollaan laiskoina tämän suhteen
  - Jos toteuttaa esim. webhook-API:a, voi osaksi testiapia tai tuotantopuolella mahdollista rekisteröintifunktiota vasten tehdä kahden kutsun strategian: laittaa ensin validin ja sitten virheellisen kutsun:
  - Jos molemmat menevät identtisesti läpi, voi vastapuolta varoittaa tai kokonaan hylätä rekisteröitymisen
- Solita Whitehat ;-)





# Kiitos!

**Juho Myllylahti**

[juho.myllylahti@{solita.fi, iki.fi}](mailto:juho.myllylahti@solita.fi)  
@Mutjake



