Assume Breach Mindset for Cloud Services

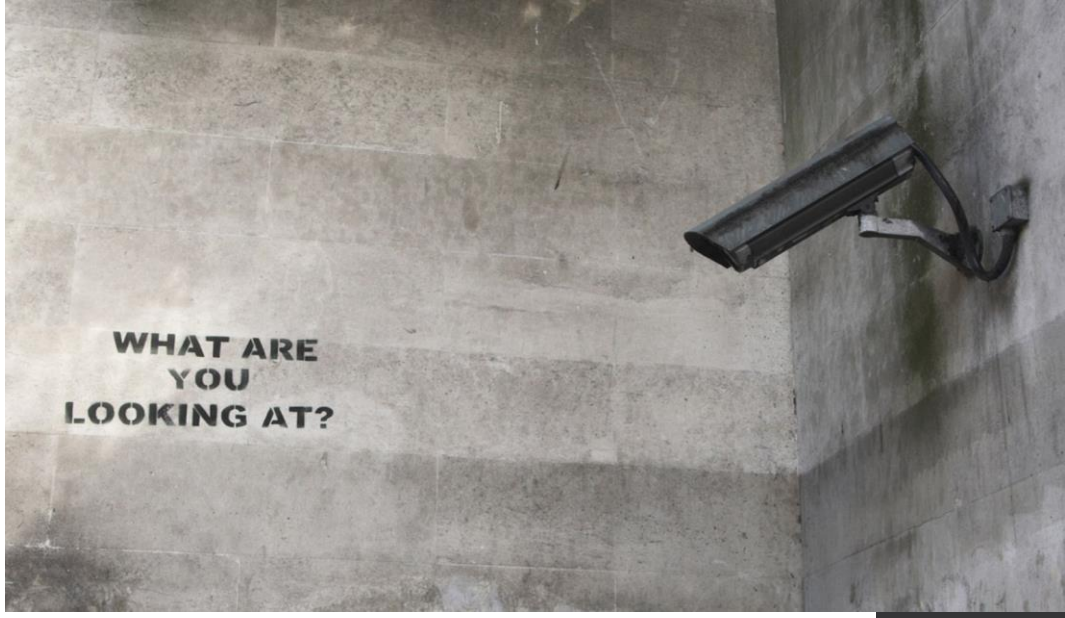
Seosaimh O'Shea (Artwork – Bansky)

Me ..

Security Fundamental ->
Application / Product Security ->
DevOps / Cloud Security

1998 ……..2006……2010……..2012..

Talk will explore ..



- Why we should assume breach ..
  - Reflection on how we process of risk ..
  - Cloud security benefits are not automatic
- **Assume breach**
  - Example scenarios: help determine is a service breach ready
  - Goals: Detect, Investigate, Remediate
  - Lateral movement
- Cloud design considerations to help **limit lateral movement**
- Conclusion

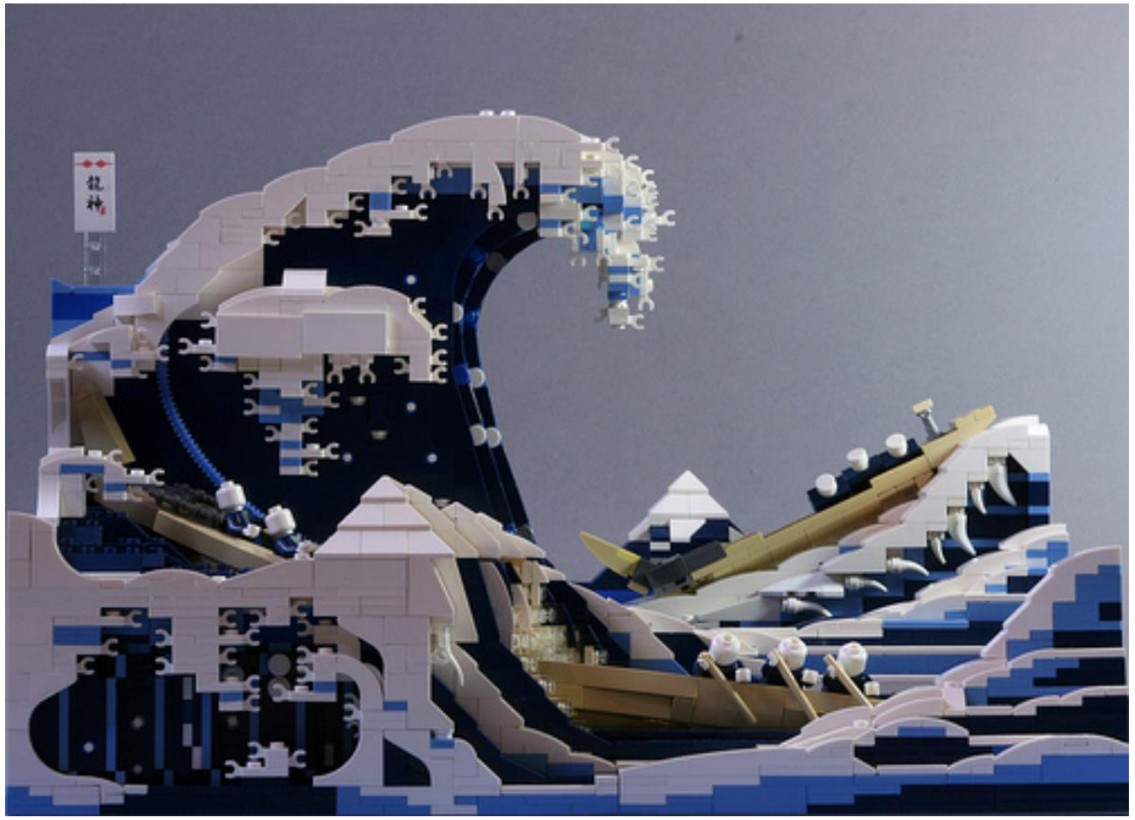
Why we should assume breach ..

- Security + engineering often separate AppSec / OpSec / Cloud Sec risks
- Need an attacker perspective, they are pragmatic, identify weakest link
  - Move the conversation away from just applying best practices
- Threat modeling often focus on individual interactions
  - is OpSec covered? (e.g. patching, VM refreshes), Deployment pipeline?
  - access control considerations? (e.g. subscription access, standing access etc)
- Assume Breach scenarios encourage defence in depth review
- Verizon breach report - 34% breaches involving insiders
- Do we think about the Risk of data Deletion or Disclosure

Awareness of how we process risk ..

Masha Sedova, (Elevate Security)

- industry-recognized people-security expert
- Presentation: ***Why humans suck at calculating risk and how it affects security***
- We process threats with the following characteristics
  - Intentional
  - Immoral
  - Imminent
  - Instantaneous
- Summary Learnings
  - Knowing about a threat often doesn't trigger a response
  - "Bad things wont happen to me"
  - The more we avoid failure, the riskier our actions
  - Security issues that don't register as one of the 4 categories above don't trigger appropriate responses



Credit: koffiemoc@flickr / Hokusai

Cloud security benefits are not automatic

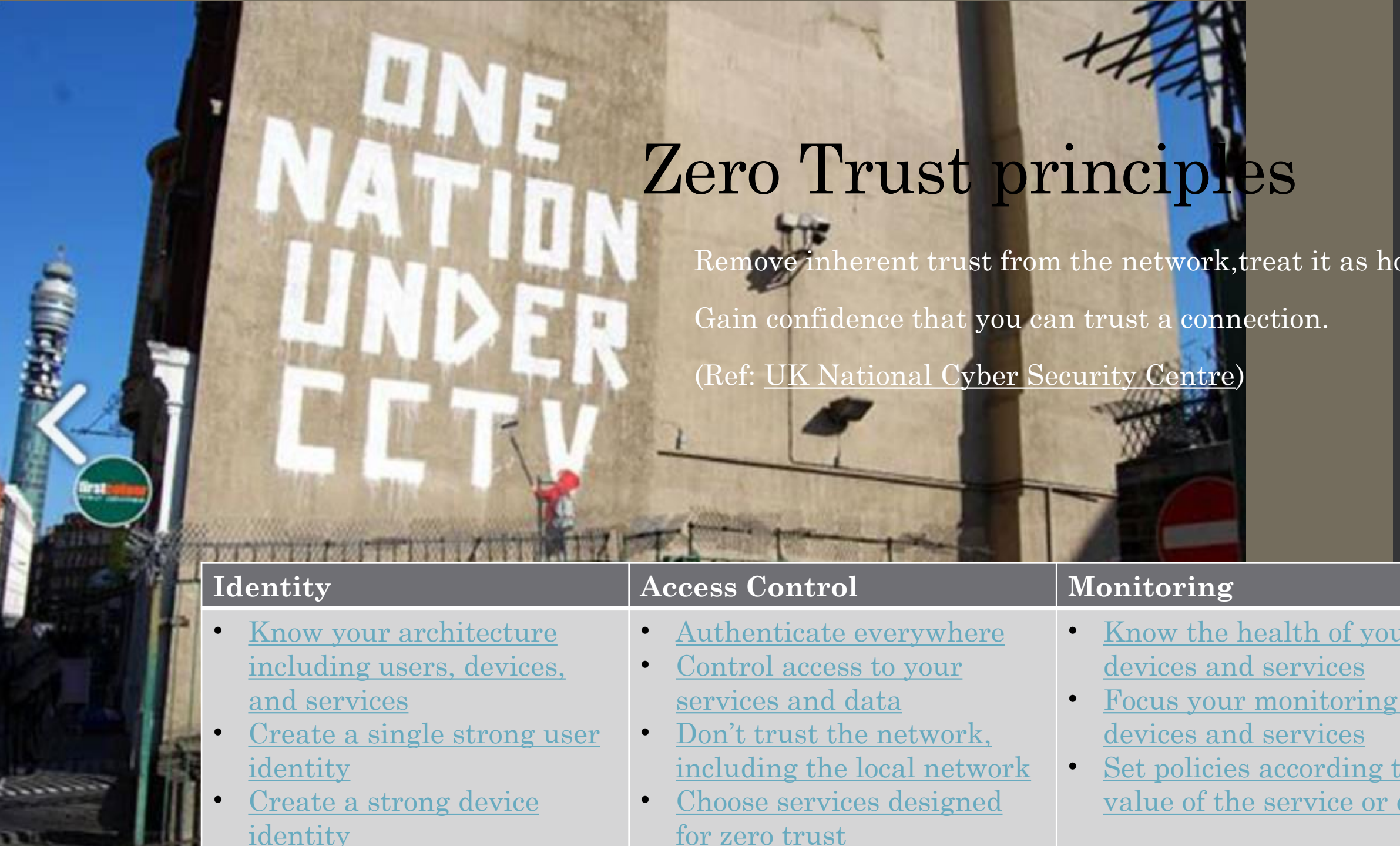
- Need to establish **trust boundaries** in our solution design
- Need to consider how to **enforce standards and policies**
  (consider enabling AWS Organizations service control policies, or Azure management group policies)
- Need to enable **MFA**, **JIT**, and use **isolated identities**
- VM's will still need system owner to be responsible for **patching** responsibilities
- Attack **surface** continuously changing, Needs oversight, (e.g. RDP port scanning, security groups)
- Cloud represents a new tech stack – are existing operational tools (e.g. IDS) still appropriate?
- Access Management: Storage & key access often not associated with users.
- **New Application** definitions. Access afforded to apps - can't be easily assessed or managed
- Who manages cloud portal / **subscription access?**
- **Hybrid** cloud - internal network has a cloud attack surface

Anti patterns?

- Lift & shift migrations from DC to single subscription / cloud account
  - Ops folks think about securing the network or external perimeter
  - There is no isolation gain if all resources are deployed to a single subscription
  - Have Ops staff been skilled-up for cloud? Is current Ops toolset sufficient?
  - **Consider** defining cloud account usage around production services
- Application definitions often are much too broad
  - many product capabilities all grouped under a single application definition.
  - Access granted to the app or by the app to other apps or users become difficult to manage.
  - Avoid too much access granted to an attacker in the event of a service compromise.
  - **Consider** defining applications scope that is specific to a DevOps team or functional area.
- Clustering of unrelated applications on a single container service instance for cost
  - Use of container clustering offering can violates isolation principles – The trust boundary becomes the clustering resource instance.
  - **Consider** deploying applications that are dependent to each other within a single cluster
  - but deploy applications that are not related in separate cloud accounts.
- Other factors for consideration
  - **Data storage can become fragmented** across storage resources across different cloud accounts
  - There are often **no offline backups** of data.
  - **Patching hygiene** can suffer - VM's that persist need monitoring for updates.
  - Hybrid Cloud adoption – network perimeter extended, needs careful architectural oversight.

Accepted design principles

- Isolation
  - Related resources grouped and isolated from other systems. Access and identity rules restrict access and form a trust boundary.
- Reduce Attack Surface
  - restrict what is discoverable to an attacker, and limit permissions
- Restrict Access
  - Access should be time bound, restricted by scope.
  - Least privilege
  - Zero standing access - extends to all cloud services
  - specialized JIT. Avoid logging into services
- Security monitoring: service aware /specific
- Incident response: planed with responsibilities



Zero Trust principles

Remove inherent trust from the network,treat it as hostile.

Gain confidence that you can trust a connection.

(Ref: UK National Cyber Security Centre)

Identity	Access Control	Monitoring
• Know your architecture including users, devices, and services • Create a single strong user identity • Create a strong device identity	• Authenticate everywhere • Control access to your services and data • Don't trust the network, including the local network • Choose services designed for zero trust	• Know the health of your devices and services • Focus your monitoring on devices and services • Set policies according to value of the service or data

Assume breach scenario examples

Assume breach – don't need to justify how… consider what would happen next ..

Choose whatever scenarios are relevant to your service and environment.

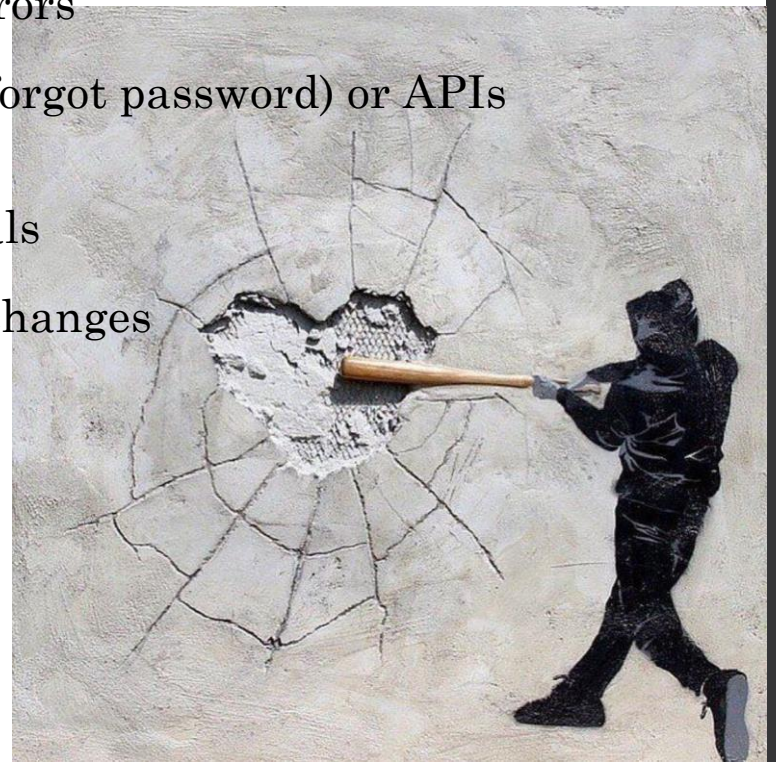
- Compromised application CDN account or other service dependency
- Use of a vulnerable 3rd party library
- Malicious code submitted to build system (e.g. unit tests).
- RDP port left exposed brute-forced, VM compromised
- Phishing attack can grants malicious app access to your Cloud Account portal
- Dependent Service compromised and provides untrusted data
- Disclosure of a storage key leads to unauthorised data access (API or storage account) or deletion.
- Hybrid cloud: access to cloud systems from corporate network achieved (e.g. credential disclosure in source code)
- **Spear phishing** attack targeting admin or customer

BREACH: Detect the attackers

Detect attacker in your Service: info gathering/ Lateral movement/ data exfiltration

Not a job for Blue team alone… Service owners need to monitor for service misuse

- Detect unusual access to data by accounts or large data exfiltration
- Monitor failed logins, authorization failures or API usage errors
- Unusual API usage, esp relating to User account APIs (e.g. forgot password) or APIs with financial consequences. (e.g. bet placement)
- Detect a compromised app requesting all KeyStore credentials
- Unexpected attempts to deploy new code, updates or config changes



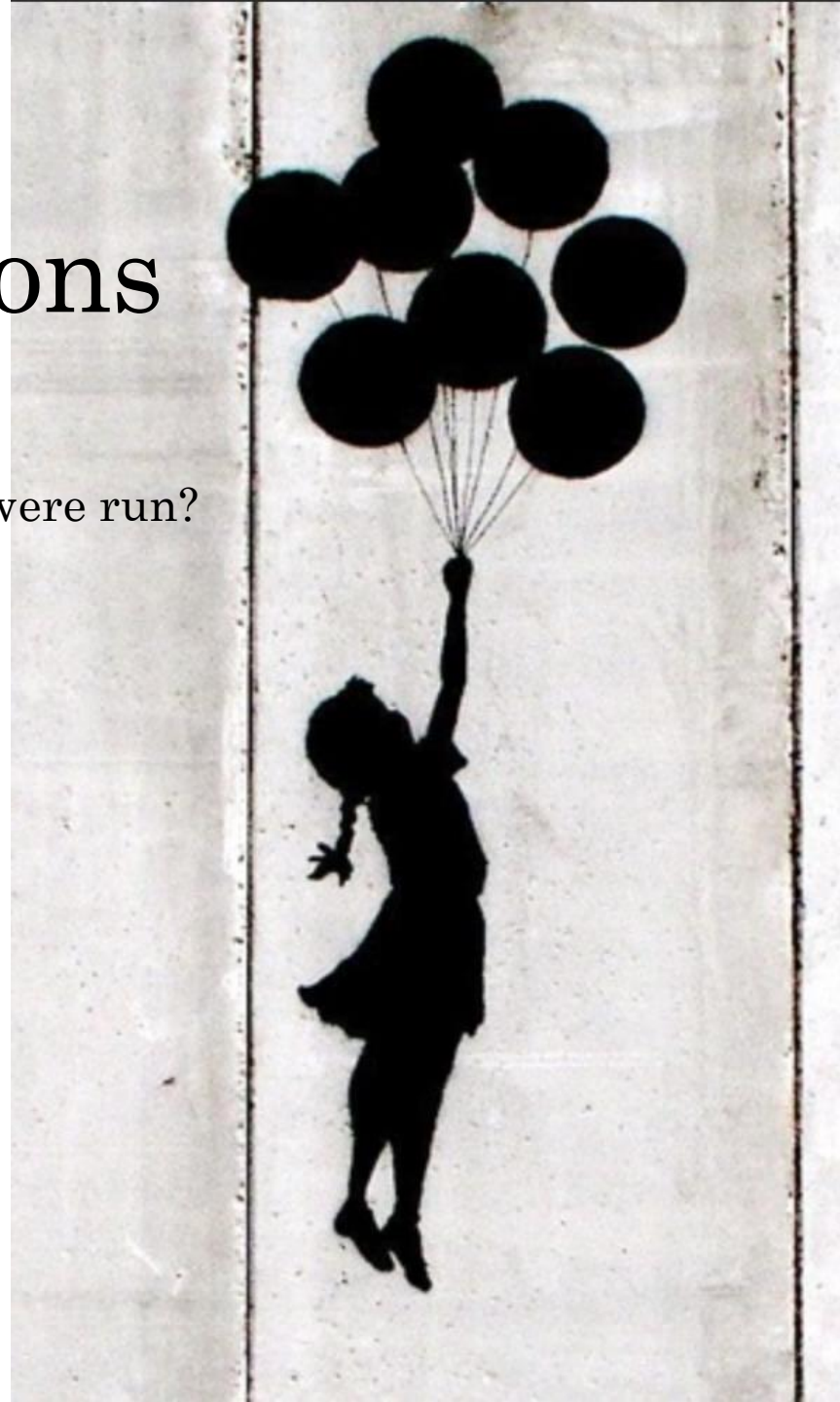
According to Verizon 2019:

- 32% of breaches involved phishing
- 71% of breaches were financially motivated
- 29% of breaches involved use of stolen credentials
- 28% involved Malware
- 34% involved Internal actors

BREACH: Investigate actions

We need to go beyond just detecting a possible breach.

- What is the extent of the attackers access? What commands were run?
- What access & accounts were used? TTP
- When did it first occur?
- Are customer impacted?
- Has data been corrupted?
- Were config settings modified?



BREACH: Service remediation plans

How confident are we that we can we evict the attacker?

- Can we restore a valid dataset?
- Asset clean-up?
- Are other dependent services impacted?
- At what point have we removed the attackers access / evicted him
- When can we perform key / credential rotations

Define a detailed plan for relevant scenarios.

Lateral Movement

“RUSSIAN GROUPS ACHIEVE LATERAL MOVEMENT WITHIN 18 MINUTES “ -

CROWDSTRIKE 2019 GLOBAL THREAT REPORT

Mitre Lateral movement techniques

Top 20 Techniques from ATT&CK Group/Software Data

A *starting point!* Not representative of all adversary behavior

1. Standard App Layer Protocol
2. Remote File Copy
3. System Information Discovery
4. Command-Line Interface
5. File and Directory Discovery
6. Registry Run Key/Startup Folder
7. Obfuscated Files or Information
8. File Deletion
9. Process Discovery
10. System Network Config Discovery
11. Credential Dumping
12. Screen Capture
13. Input Capture
14. System Owner/User Discovery
15. Scripting
16. Commonly Used Port
17. Standard Crypto Protocol
18. PowerShell
19. & 20 (tie!)

Masquerading and New Service

Limiting lateral movement

- Limit Access / Blast radius– don't afford the attacker privileged access
  - Limit what is deployed to a cloud account,
  - use isolated identities. Administration using non-corporate / non-public identities
- Restrict services sharing the same application definition / permissions
- JIT & Least privilege access.
  - Avoid standing access accounts. Access must be on-demand, Automate admin tasks.
- Restrict Access – within subscription:
  - credential Storage containers,
  - build & deployment permissions
- Restrict other environments sharing app creds

Limiting lateral movement

- Enable code signing checks to protect against unauthorized code deployments
- Limit different microservices deployed to microservice clusters (increase isolation vs service efficiency)
- Continuous Monitoring for RDP & remote access ports
- Access limitations within cloud account: Network Security Groups / VPC's, define roles / permissions for accounts
- Protect CICD pipeline, limit permissions.
  - Run unit tests in VMs, try to avoid excessive permissions being granted
- Limit destruction risks:
  - Consider Resource Locks to protect against possible data loss. Read only / delete
  - Apply restrictions to management groups/ Organizations, cloud accounts, resource group, or resource level.
- Data Backup kept outside the subscription

Conclusion

Expected Human Behaviours	Expected Responses
• Lazy Vulnerability tracking • Lazy Access Management • Credential disclosures • Operational mistakes • Attack Surface expansion • Medium/High risk appetite • Lack of consideration for Defence in Depth	• Expect a breach. Plan to detect & respond. • Base security assessments on real scenarios • Involve service owners & engineers in the assessment task • Design services with defence in depth and isolation