

# Netflix's Layered Approach to Reducing Risk of Credential Compromise

WILL BENGTON  
SENIOR SECURITY ENGINEER

TRAVIS MCPEAK  
SENIOR SECURITY ENGINEER

NETFLIX



# PIZZA

API  
PROTECT



CREDENTIAL  
COMPROMISE  
DETECTION



REPOKID



ROLE  
PROTECT



ZERO  
STATIC KEYS



ANOMALY  
DETECTION



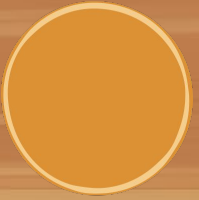
**Segment  
environment into  
accounts**



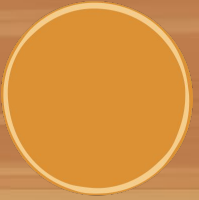
A large wildfire is burning on a hillside, with thick black smoke rising into the sky. Several firefighters in yellow gear are visible on the ridge, working to contain the fire. The foreground shows dry, scrubby vegetation and cacti. The text "If the account gets compromised, damage is contained." is overlaid in white on the lower left. The Netflix logo is in the bottom right corner.

**If the account gets  
compromised, damage is  
contained.**

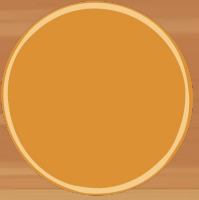
**NETFLIX**



**Useful when  
broad permissions  
are required.**



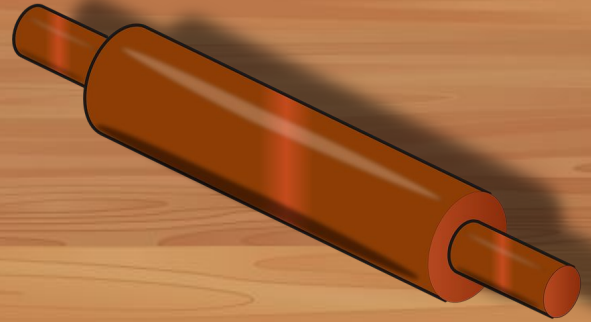
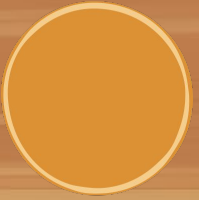
**Useful for  
separation of duties.**



**Useful for sensitive  
applications and data.**



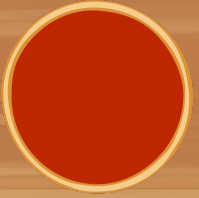
Reduce friction by  
investing in tooling to  
*C.R.U.D.* AWS accounts.



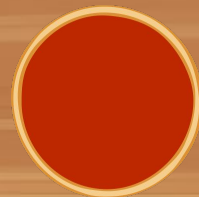
NETFLIX


**Remove static keys**





**Static keys never expire  
and have led to many  
compromises.**



Fraud Amazon Web Services Amazon.com (product) +4 

## My AWS account was hacked and I have a \$50,000 bill, how can I reduce the amount I need to pay?

### I was billed for 14k USD on Amazon Web Services 😱

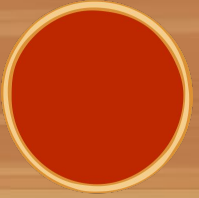
*What happened?*

A file containing my AWS credentials hadn't have been ignored in git, so when I pushed my local repository it all, even my credentials, went online in github (for couple of minutes, but they were there).

### Ryan Hellyer's AWS Nightmare: Leaked Access Keys Result in a \$6,000 Bill Overnight

"In total, there seemed to be around 600 servers running. The time between realizing all this and uploading my Git repository was approximately 12 hours."

"But those horrid little AWS access keys were sitting on the repository in view of everyone. I immediately deleted the entire repository from GitHub."



**Short-lived keys,  
delivered securely,  
rotated automatically.**

# Permission Right Sizing





**We continuously and automatically remove unused permissions.**

# Applications converge to least privilege.



NETFLIX





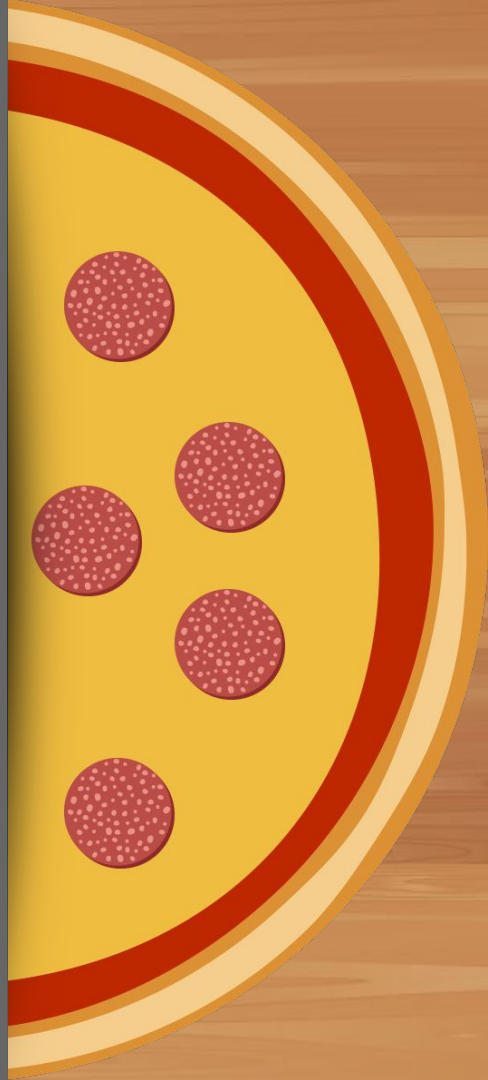
Unused applications  
converge to ***zero.***

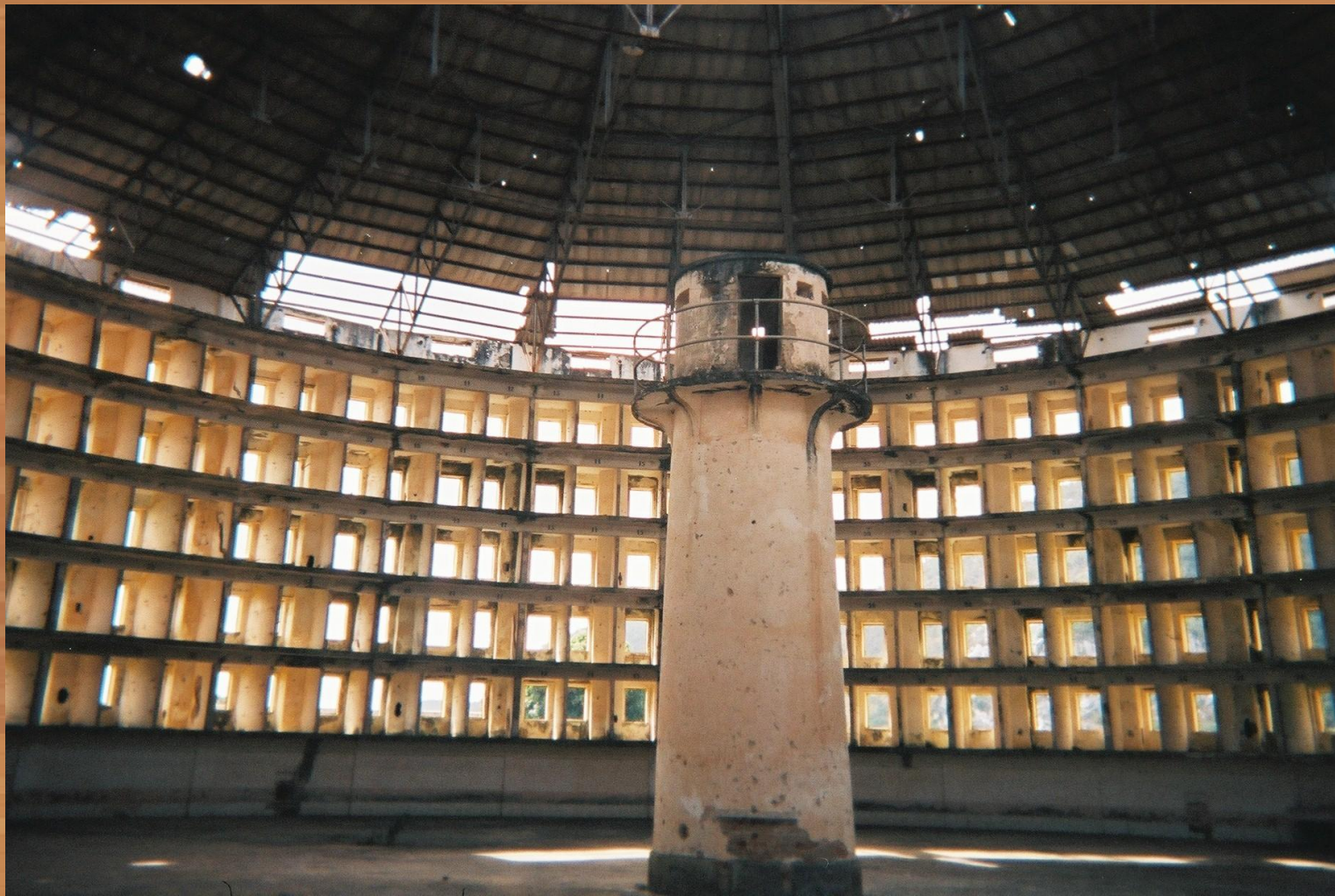


<https://github.com/Netflix/repokid>

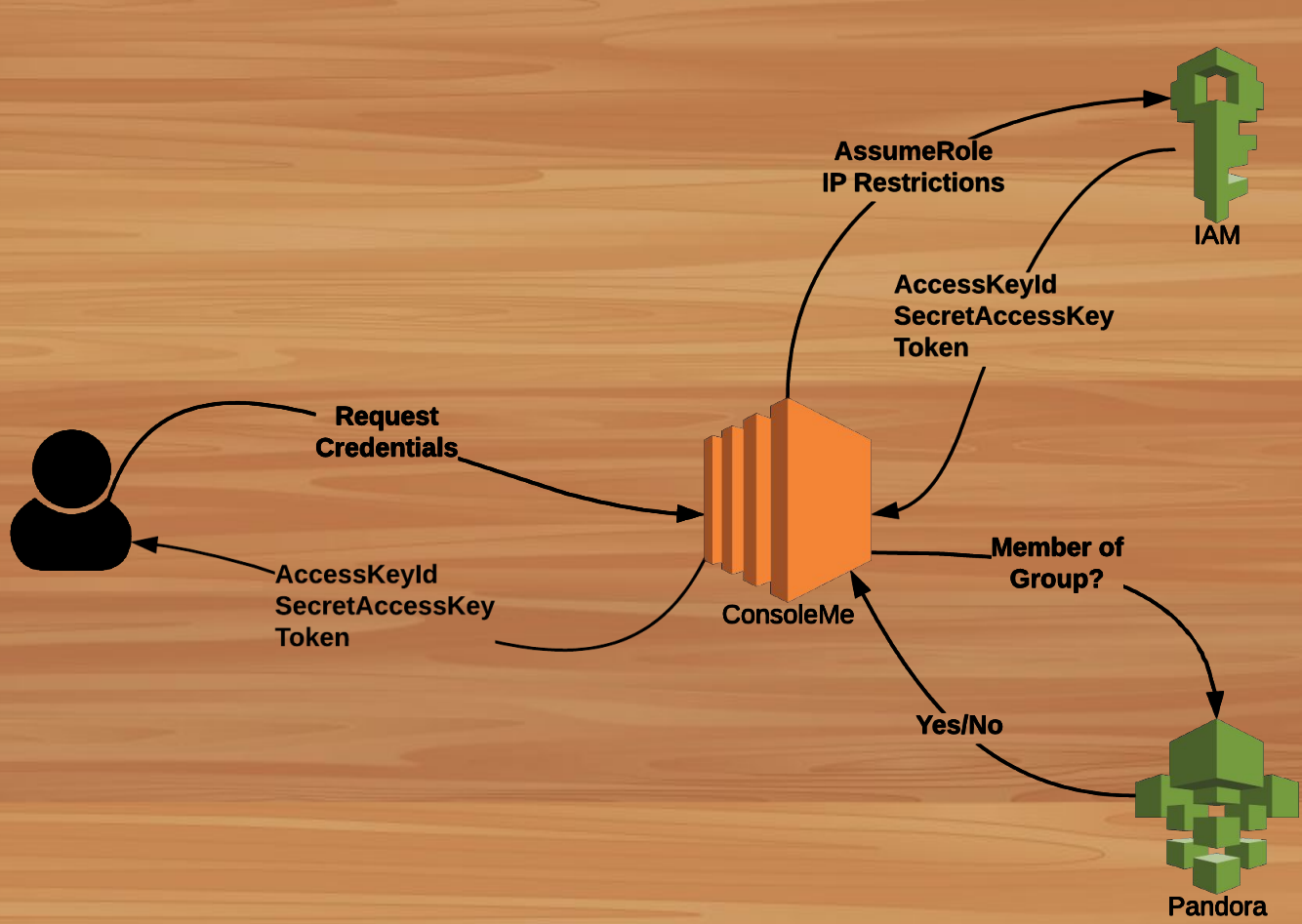
NETFLIX

# Paved Road for Credentials





**NETFLIX**



**NETFLIX**

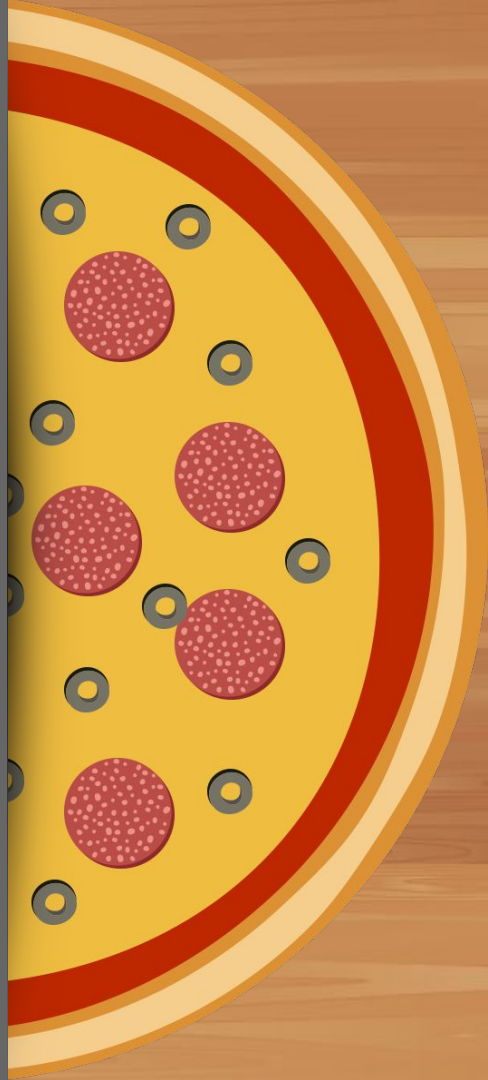


# Auditing / logging



# Anomaly detection

**Prevent instance  
credentials from  
being used  
off-instance**







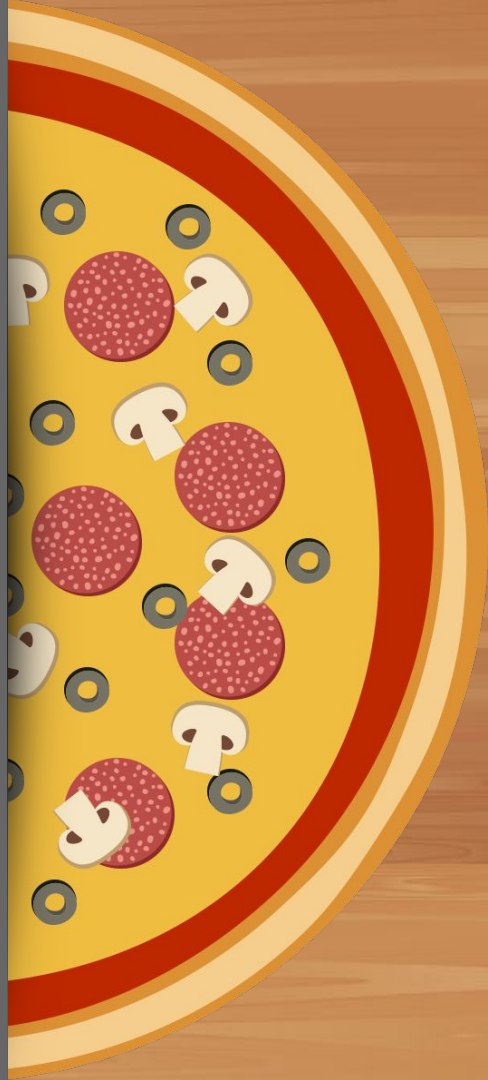
**If attacker tries to steal  
creds, they don't work.**



**IRL**

**NETFLIX**

# Delivery Lockdown





# We use Spinnaker



**We applied tags that  
restrict roles to specific  
applications.**



**We applied application  
specific AuthZ controls  
(Fiat).**



**Together: these two  
controls prevent privilege  
escalation.**

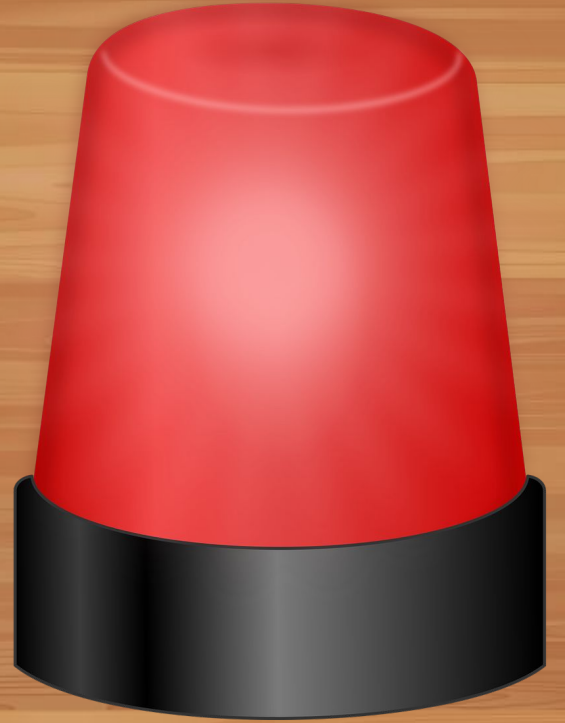


**We rolled this out  
environment wide.**

**NETFLIX**



**Automated  
detection and  
alerting *FTW*.**



**Detect instance  
credentials used  
off-instance**





**We already block it.**

**An attempted use is a  
signal.**



# Example

**Detect anomalous  
behavior in  
environment**





**We track baseline behavior  
for an account.**



Some  
regions, **resources**, services  
shouldn't be used.



**A perk of continuously  
watching CloudTrail.**

**NETFLIX**





# Example

# Detect anomalous behavior by roles





**Same idea, but at the  
application/role level.**



**Applications have  
relatively consistent  
behavior.**

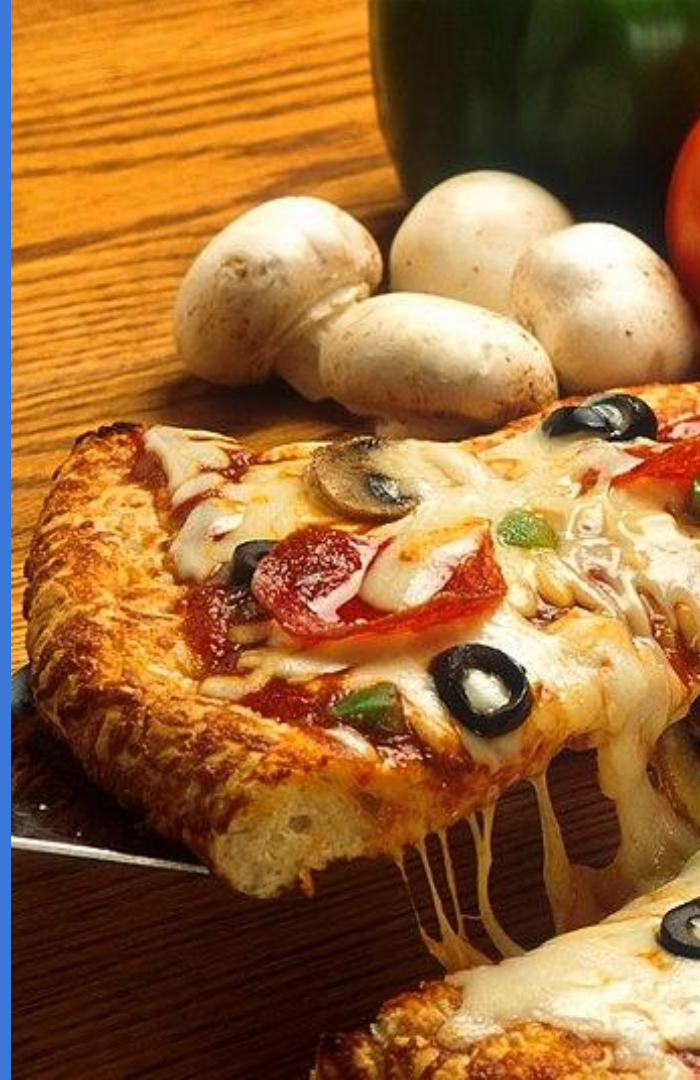


**Look for common first  
attacker steps.**



# Example

**Future:  
One role per user**



**User roles  
become unique  
as a fingerprint.**





**This can be used to detect  
unusual behavior.**

**Future:  
Remove users  
from accounts  
they don't use**

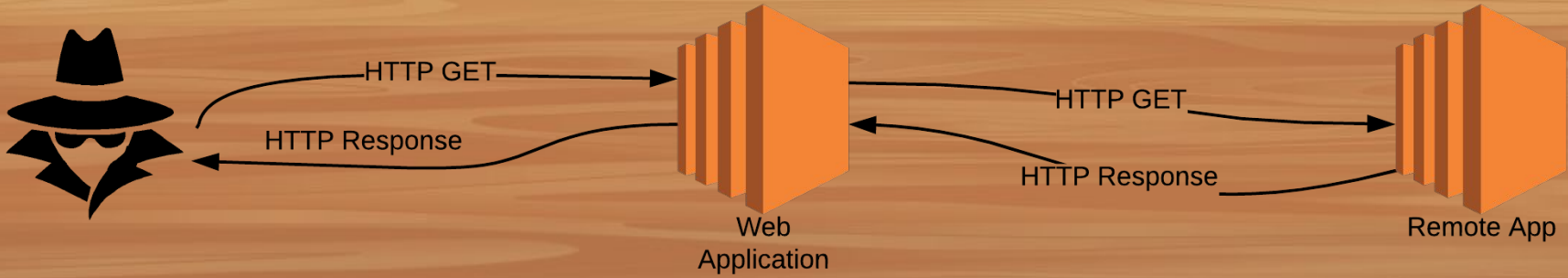


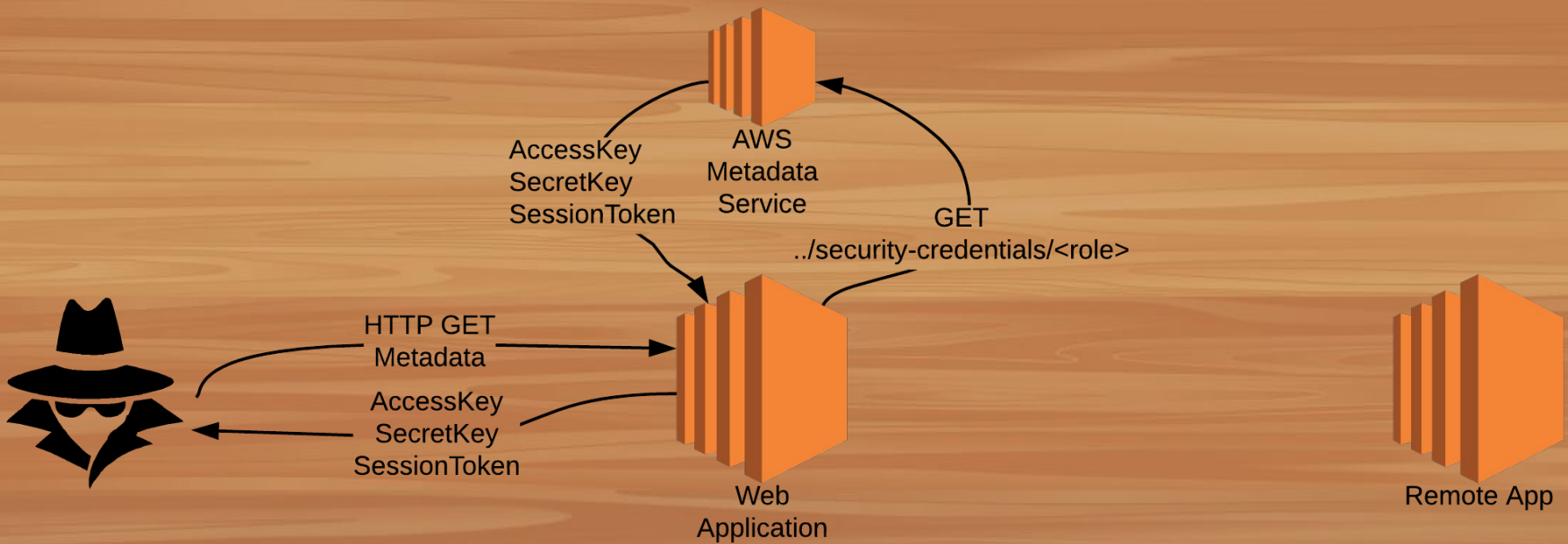
**Reduces the risk of user  
workstation compromise.**

# Future: Metadata work



# Prevent compromise at the credential source



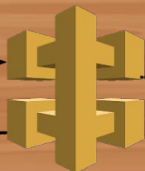




Python  
(boto)

GET ../security-credentials/MyRole  
User-Agent: Boto3/1.....

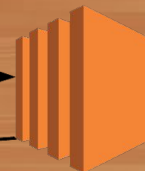
AccessKey  
SecretKey  
SessionToken



Metadata  
Proxy

GET ../security-credentials/MyRole

AccessKey  
SecretKey  
SessionToken



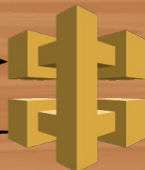
Metadata  
Service





Vulnerable App

GET ../security-credentials/MyRole  
User-Agent: python-requests/1....

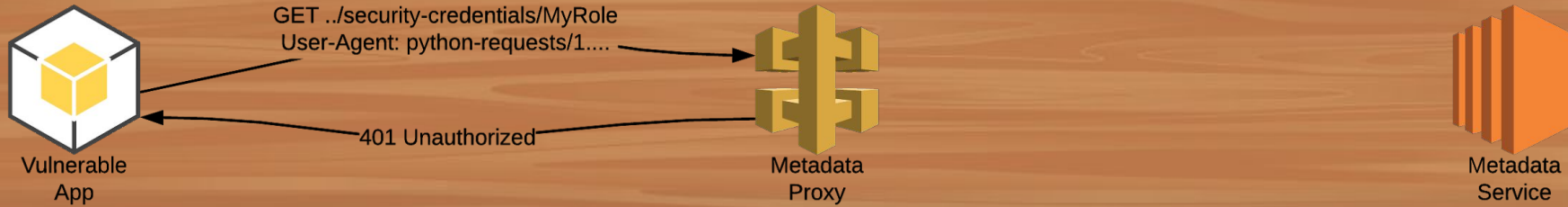


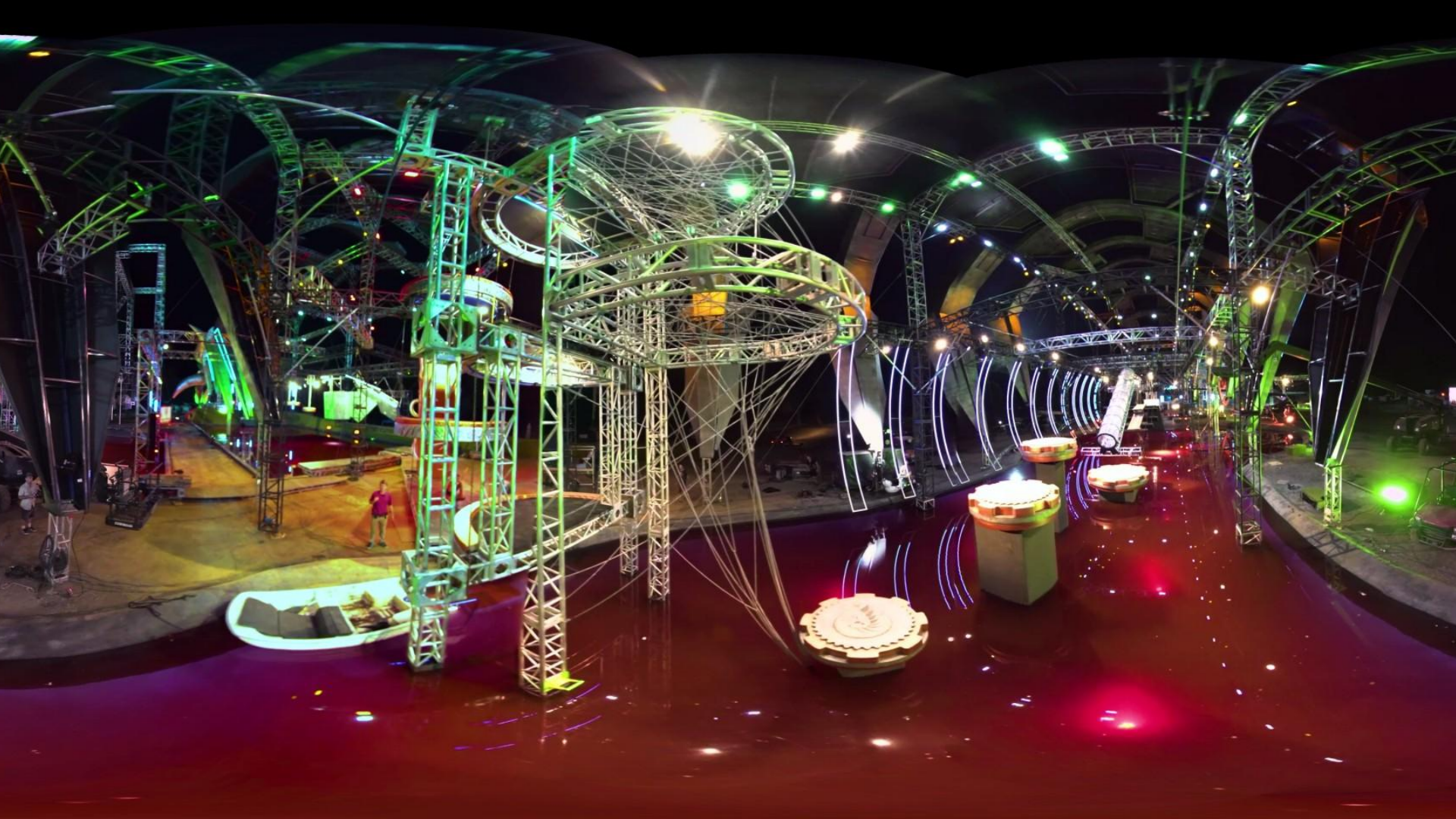
Metadata Proxy

401 Unauthorized



Metadata Service





# THANK YOU!

