AzureGoat : A Damn Vulnerable Azure Infrastructure

Jeswin Mathai, Nishant Sharma, Sherin Stephen, Rachana Umaraniya

About US

Jeswin Mathai

- Chief Architect, Lab Platform @ INE
- Published Research at Black Hat US/Asia Arsenal, DEF CON USA/China Demolabs
- Gave research talk at DEF CON China and Rootcon Philippines
- Co-Trainer in Training:
  - Black Hat Asia
  - HITB AMS, GSEC
  - NZ OWASP day
  - Rootcon 13

About US

Nishant Sharma

- Director, Lab Platform @ INE
- Firmware developer, Enterprise WiFi APs and WIPS Sensors, Mojo Networks (Acquired by Arista Networks)
- Masters degree in Infosec
- Published research at Blackhat US/Asia, DEF CON USA/China, HITB Amsterdam and other venues
- Conducted trainings in HITB, OWASP NZ day and for multiple private clients

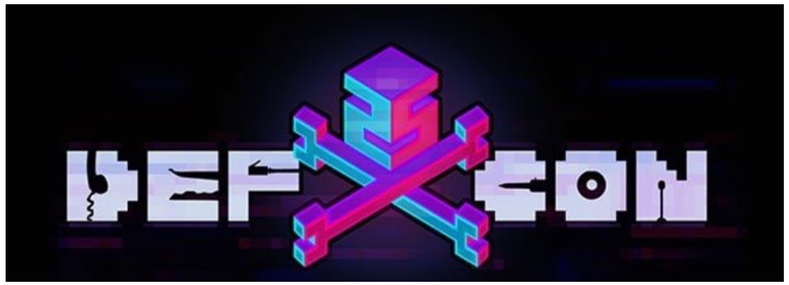
About US

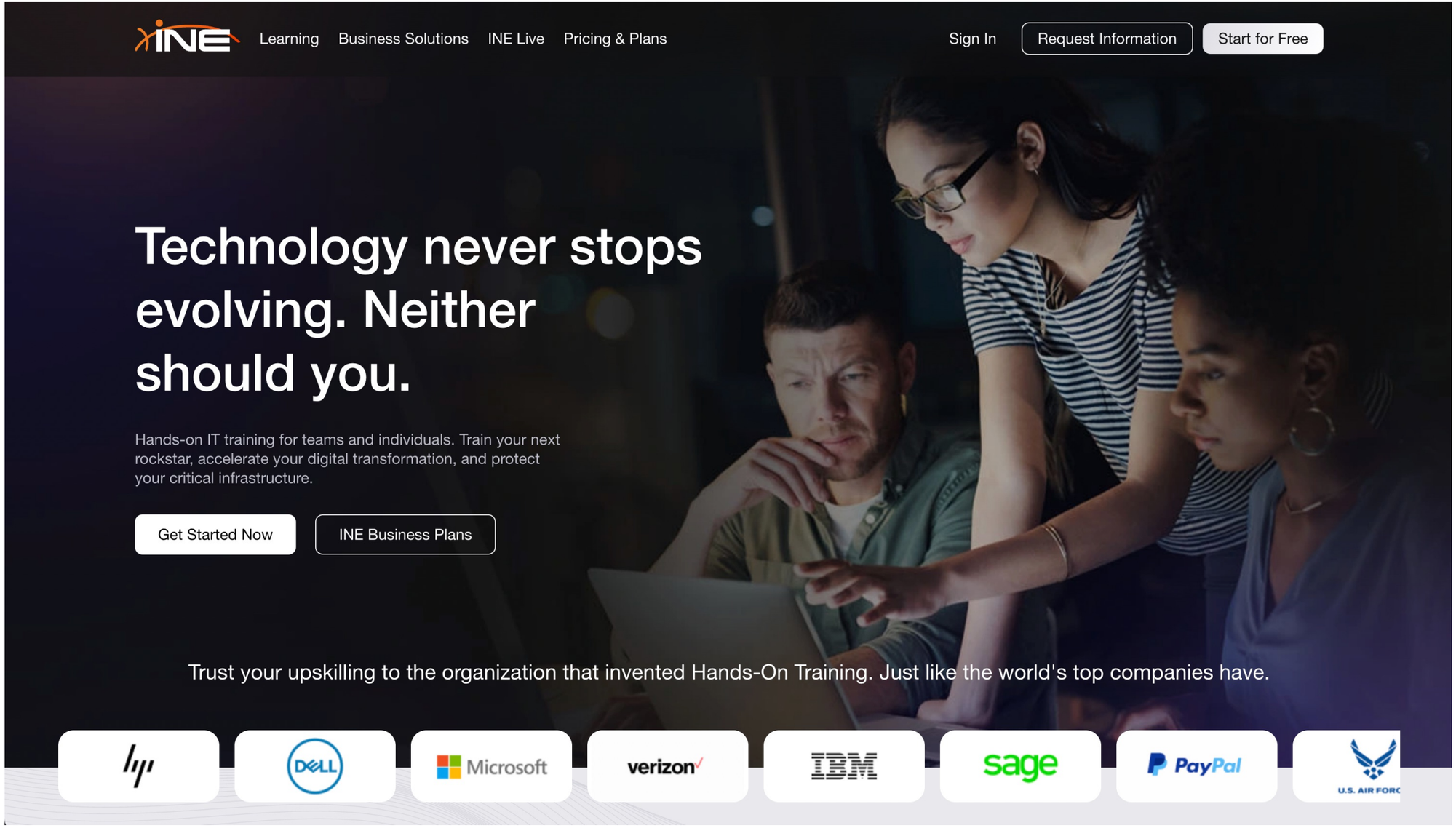
Sherin Stephen

- Cloud Developer @ INE
- Presented his work at BlackHat Asia Arsenal 2022
- Experienced in Building and maintaining reusable code and robust cloud services

Rachana Umaraniya

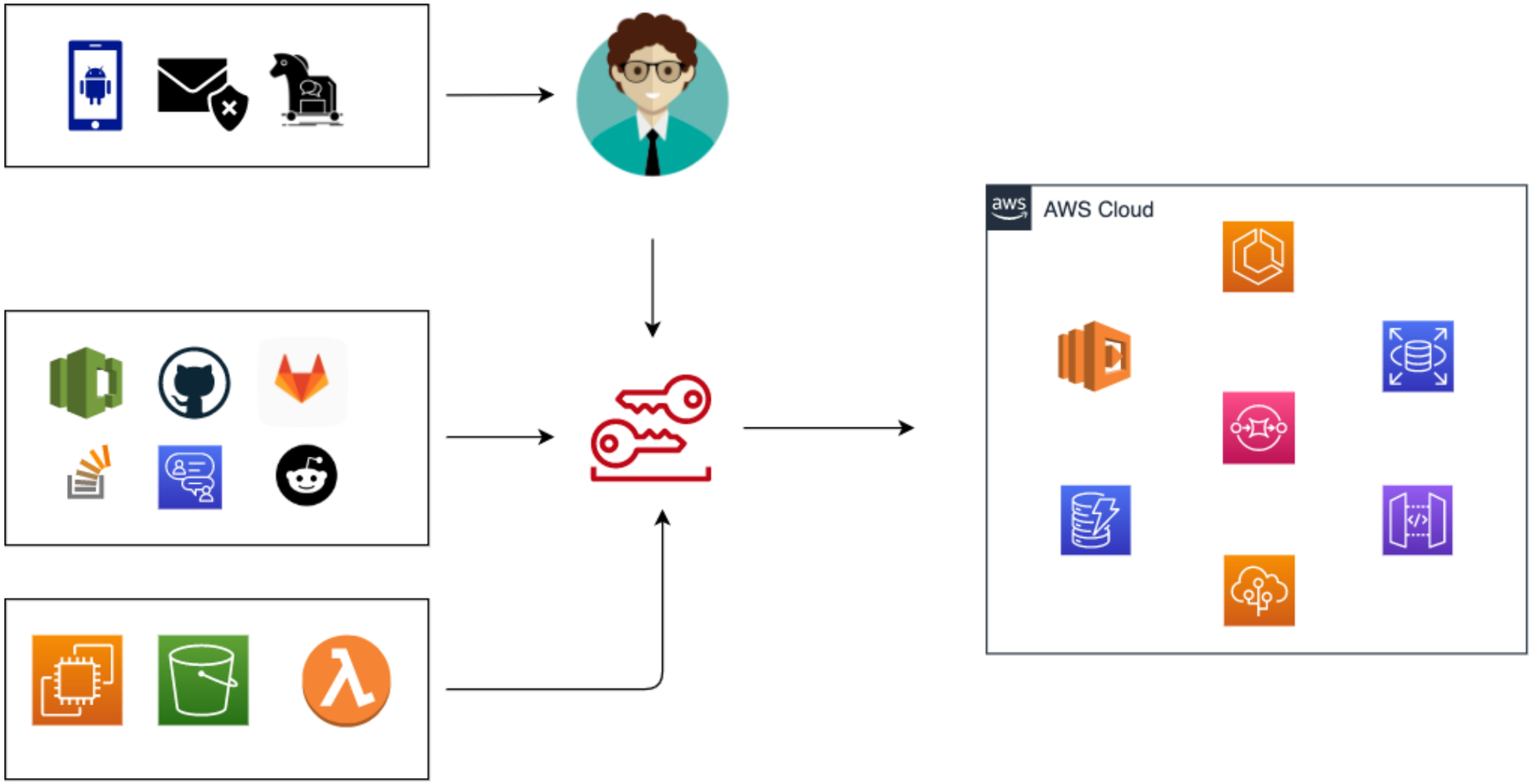
- Cloud Developer @ INE
- Master's Degree in Computer Science
- Two years of experience in software development and specializes in Java Frameworks





Threatscape

Threatscape



Motivation

- Training Needs
- Lack of Real World Azure Pentesting Environment
- Contribution from the open source community and security professionals
- Release of OWASP Top 10: 2021

Introducing AzureGoat

AzureGoat : A Damn Vulnerable Azure Infrastructure

- Mimics real-world infrastructure but with added vulnerabilities
- Multiple application stacks - Multiple exploitation/escalation paths
- Features OWASP Top 10: 2021
- Focused on Black-box approach
- Still in early stage
  - Module 1 : Blog Application

OWASP Top 10

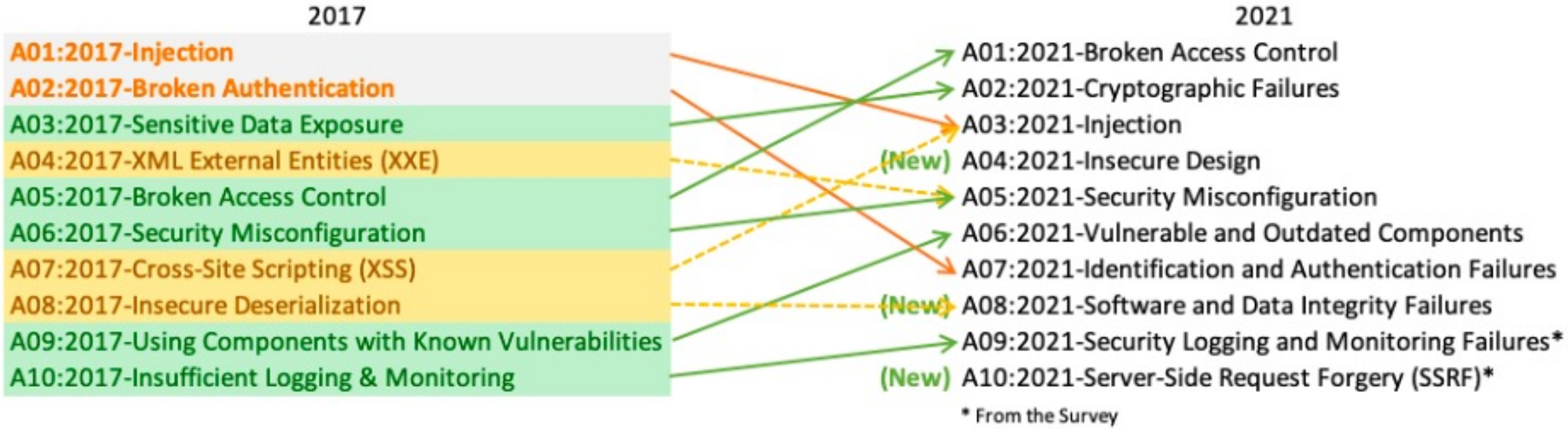
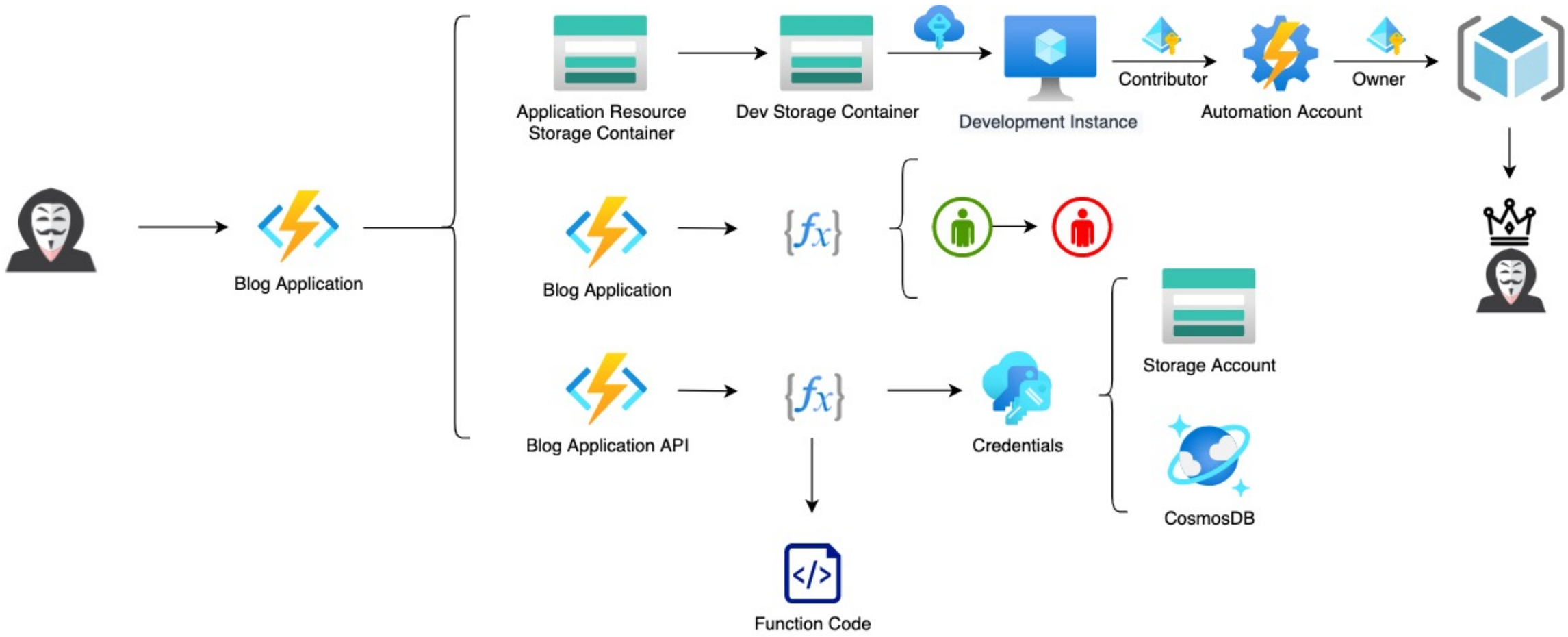


Image Reference: https://owasp.org/www-project-top-ten/

AzureGoat : Module 1 (Blog Application)

- A01: Broken Access Control
- A02: Cryptographic Failure
- A03: Injection
- A04: Insecure Design
- A05: Security Misconfiguration
- A07: Identification and Authentication Failures
- A10: Server Side Request Forgery

AzureGoat : Module 1 (Blog Application)



Building Realistic Insecure Application : Challenges

- Security Professional vs Seasoned Developers
- Mimicking Development Process
- Multiple Developer Environments
- Fast-paced development
- Lack of secure code practices

Project Family

Installation

- Repository: https://github.com/ine-labs/AzureGoat

- Requirements
  - AZ Utility
  - Terraform
  - Python
  - Git
- Commands
  - az login
  - git clone https://github.com/ine-labs/AzureGoat
  - terraform init
  - terraform apply

Installation

Attacking the Application

- Reflected XSS
- SQL Injection
- Insecure Direct Object Reference
- Server Side Request Forgery
- Sensitive Data Exposure
- Password Reset
- S3 Misconfiguration
- IAM Privilege Escalation

Exploitation

Server Side Request Forgery

- Reading the source code of the application

- Reading the environment variables
  - Storage Account Credential Strings
  - CosmosDB Credentials.
  - Escalate Privileges

- Enumerate other applications/instances in the network

Hunting Storage Accounts and Containers

- Globally unique
- Company-wide naming practices: Predictable names - based on departments/applications
- Misconfigured Storage Account - plethora of information

Privilege Escalation

Future Plans: Multiple Applications across Multiple Tenants

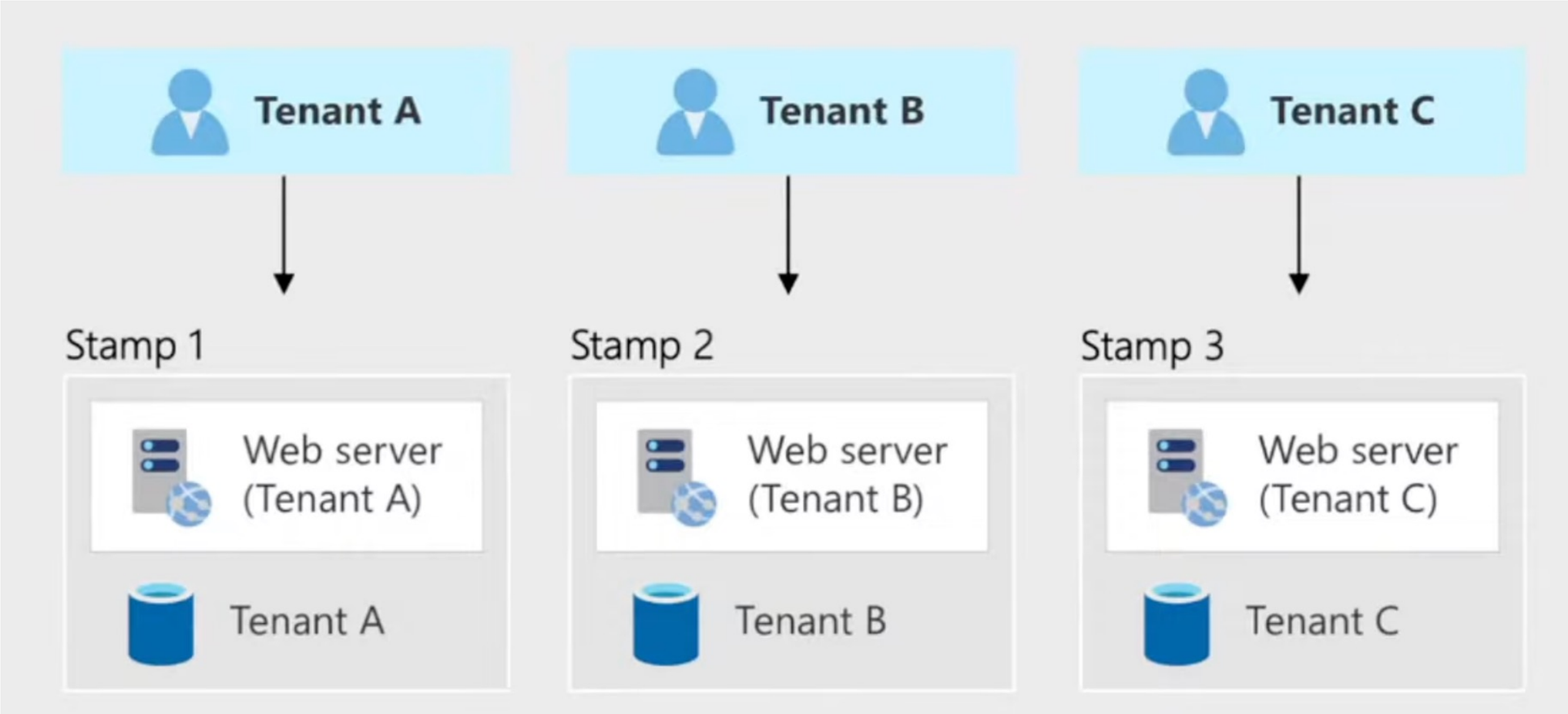


Image Reference: Architecting multitenant solutions on Azure

Future Plans

- More modules: Virtual Machine, Container Instances and AKS
- Multi Tenant infrastructure
- Working with the community
- IaC Misconfigurations
- Secure coding/deployment practices

Thanks

jmathai@ine.com