Bio:

Will Bengtson is senior security engineer at Netflix focused on security operations and tooling. Prior to Netflix, Bengtson led security at a healthcare data analytics startup, consulted across various industries in the private sector, and spent many years in the Department of Defense. Bengtson is on the BSidesSF and Bay Area OWASP leadership team. Bengtson contributes to numerous open source projects and has spoken on topics of security across the world.

Title: Security Through Immutability at Netflix

Abstract:

Cloud service adoption is increasing across organizations, from startups to enterprises. Manual manipulation and an inability to track changes throughout an environment causes a myriad of connectivity and complexity unknowns, resulting in significant difficulties in managing and auditing the security of cloud services. As config as code increases in popularity, detection of unwarranted changes or access to your system becomes more important. Skunky is an event driven system capable of marking compute resource dirty at scale while keeping a log of events in order to address this problem.

In this talk, we will take a look at what it means to have immutable instances and the benefits of deploying them. The idea and evolution of Skunky as a solution to immutability will be discussed as well as its use in a cloud deployment at scale.

Outline:

- Introduction
- Baking vs Frying (Immutable vs Mutable)
    - Frying
        - What does frying mean?
        - Examples of ways to fry: Chef, Puppet, Ansible, etc
            - Discuss master vs solo
    - Baking
        - What does baking mean?
        - Configuration as Code
            - Changes require a redeploy
            - Changes can be audited
            - Changes can be tested prior to release through an entire pipeline
        - Examples of ways to bake: Packer, custom

- The bakery at Netflix
    - Quick glance at what we call the bakery
- Requirements and Aim for solution
    - Identify a way to bake
    - Baking of server image
    - Deployment
    - Detect if configuration changes after boot and instance is not immutable anymore
- Skunky
    - What is Skunky?
        - Deployed as Lambda Function (doesn't have to be, - but our current choice)
        - How Skunky Works
            - SQS Queues
            - IAM Roles
            - Events
            - Database
        - What "dirty" means and why it matters
        - Marking resources dirty based on events:
            - SSH
            - File reads
            - Etc etc
        - Filters
            - Be able to filter out events if needed
                - Are we investigating a vulnerability
                - Do we have a script that runs continuously that - we trust
        - Watchers
            - Act on events
                - Do we place a Tag on an instance?
                - Do we send a slack message?
        - Skunky as an intel feed
        - Trials and tribulations
            - Evolution of Skunky
            - How to deploy
            - Future work
- Questions