HOW TO HAVE VISIBILITY AND SECURITY OF CICD ECOSYSTEM

Pramod Rana

@IAmVarchashva | https://github.com/varchashva

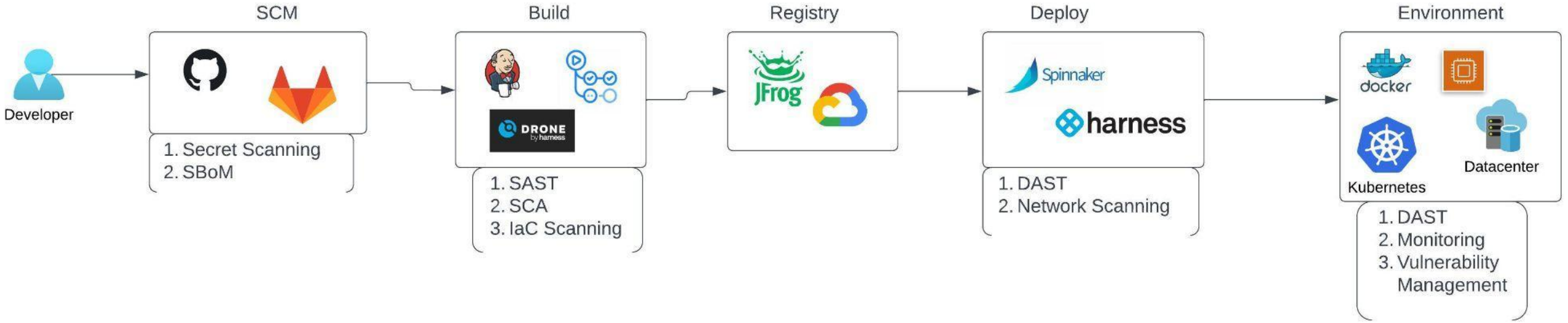
ABOUT ME

- Sr. Manager - Application Security Assurance @Netskope
- Author of three open source products:
  - Omniscient - LetsMapYourNetwork: a graph-based asset management framework
  - vPrioritizer - Art of Risk Prioritization: a risk prioritization framework
  - CICDGuard - SecurityOFCICD: Orchestrating visibility and security of CICD ecosystem
- Speaker @BlackHat | Defcon | OWASPGlobalAppSec | Insomnihack | HackInParis | nullcon | HackMiami | HITB | DevOpsDays
- OWASP Pune Chapter Leader | OSCP

AGENDA

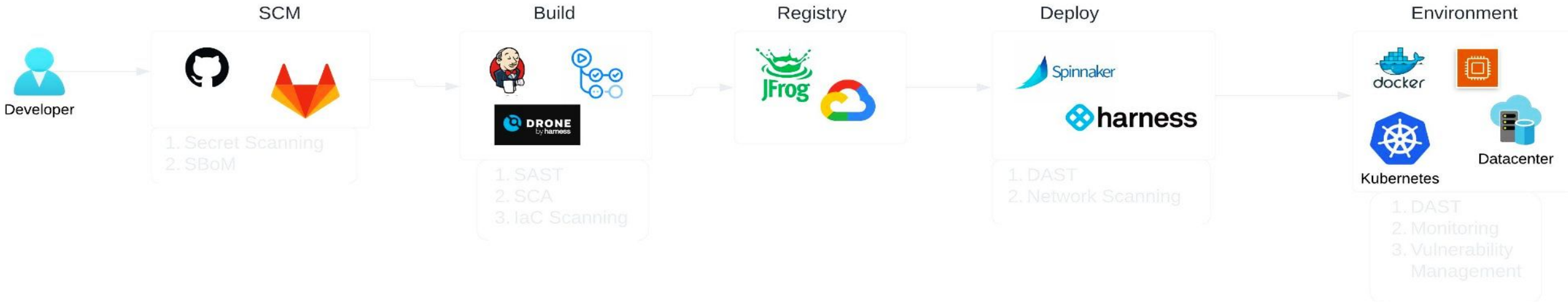
- Context
- Attack Surface
- Methodology
- Introduction to CICDGuard
- Architecture and Workflow
- Demo
- Going Forward

CONTEXT



- An oversimplified version of CICD ecosystem
- SecurityINCICD - as we all know it
- Wonderful topic but this talk is not about that

CONTEXT



- Secure the building blocks of CICD ecosystem. #Security**OF**CICD
- Compromise of one component impacts entire ecosystem
- Part of the problem is the lack of visibility into components & configurations and interconnection between different technologies

ATTACK SURFACE

Jenkins

Compromise of Jenkins console running jobs to build the binaries for end-user agent because of default/weak credentials, potentially leading to software supply chain scenario

Action

Malicious/vulnerable third-party Action running in self-hosted runners or public Actions are running in private runners leading to crypto-mining and similar attacks

GitHub

GitHub account compromised with no MFA with social engineering and thus leading to source code disclosure (IP theft)

JFrog

Compromise of JFrog user account who also has access to GitHub or Jenkins especially in case of single-sign-on

METHODOLOGY

- Focus on making technologies secure and robust, by default - Everyone needs to contribute in that
  - Implementing vetting process on organization level
  - Working with provider proactively to resolve the vulnerability
- How well we are implementing the solution in our environment
  - Default settings disabled
  - MFA enabled
  - Up-to-date plugins/apps/actions
- Are we monitoring adequately and can respond effectively, in case something happens

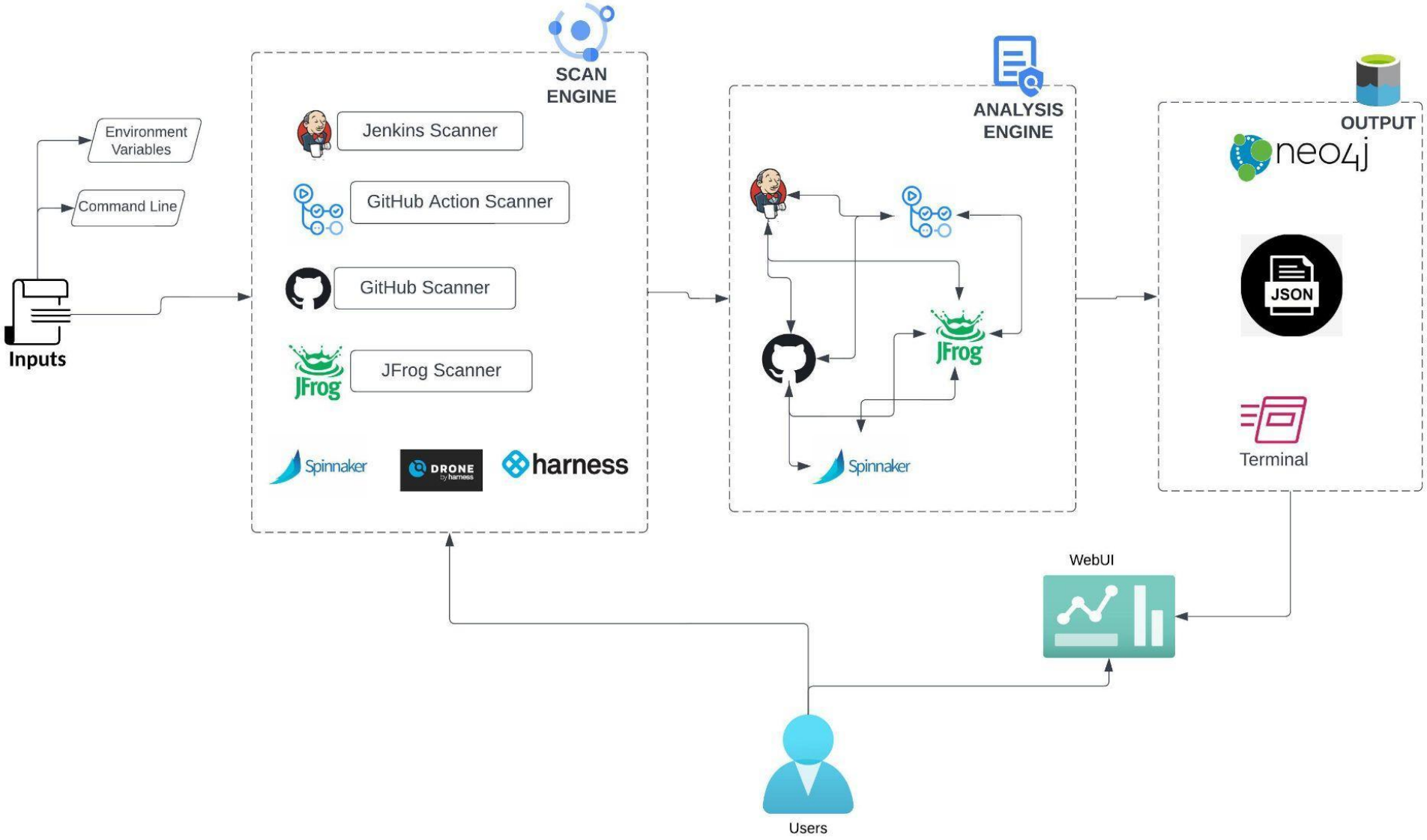
CICDGUARD INTRODUCTION

- CICDGuard represents each component of building blocks into graph
- Identifies security misconfiguration in the implementation
- Identifies relationship between different technologies and thus impact of insecurity in one technology to others. For e.g.
  - Changes in a particular repo triggering a particular Jenkins job
  - Are we using vetted version of external GitHub Action
  - Do we have common users between Jenkins and JFrog and GitHub and so on…

NODE DATABASE SCHEMA

GitHub	GitHub Action	Jenkins	JFrog
• Repository • Organization • User • Team	• Workflow • Job • Step • Command • Runner • Action	• Server • Node • Job • Build • User • Plugin	• User • Server • Group • Plugin

ARCHITECTURE & WORKFLOW



DEMO

GOING FORWARD

- Expansion of target technologies:
  - Spinnaker
  - Drone
  - Harness
  - GitLab and so on…
- Expansion of analysis engine, includes parsing of different components to determine relationship across technologies:
  - Correlation between different repositories
  - Build relating to repositories
  - Repositories and builds contributing to a particular micro-service
- More intuitive visualization

varchashva/CICDGuard

@IAmVarchashva

varchashva@gmail.com
rana.miet@gmail.com