HOW TO HAVE VISIBILITY AND SECURITY OF CICD ECOSYSTEM

Pramod Rana

@IAmVarchashva | github.com/varchashva

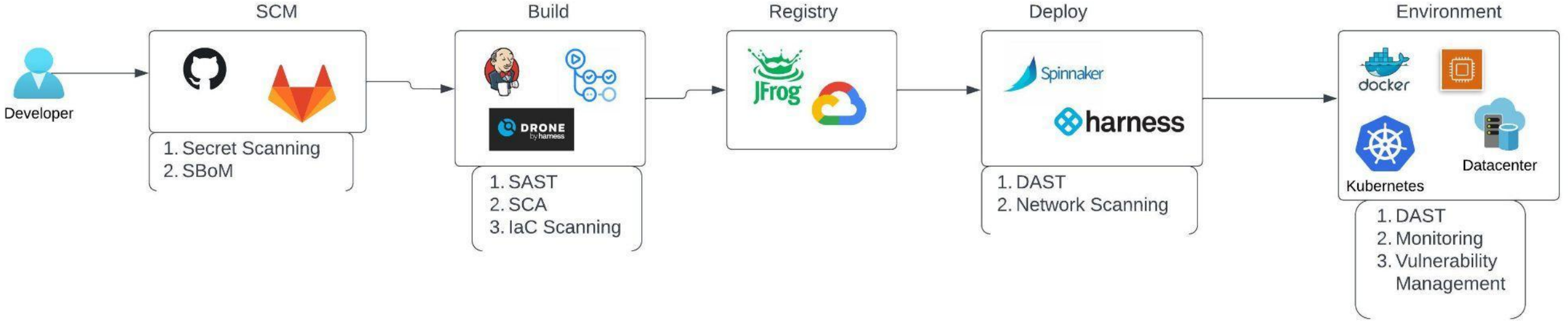
ABOUT ME

- Sr. Manager - Application Security Assurance @Netskope
- Author of three open source projects:
  - Omniscient [*Let's Map Your Network*]: Graph-based asset management framework
  - vPrioritizer [*Art of Risk Prioritization*]: Risk prioritization framework
  - CICDGuard [*Security OF CICD*]: Orchestrating visibility & security of CICD ecosystem
- Speaker @BlackHat | Defcon | HITB | OWASPGlobalAppSec | Insomnihack | HackInParis | nullcon | HackMiami | DevOpsDays | ACCMelbourne | rootcon
- OWASP Pune Chapter Leader | OSCP

AGENDA

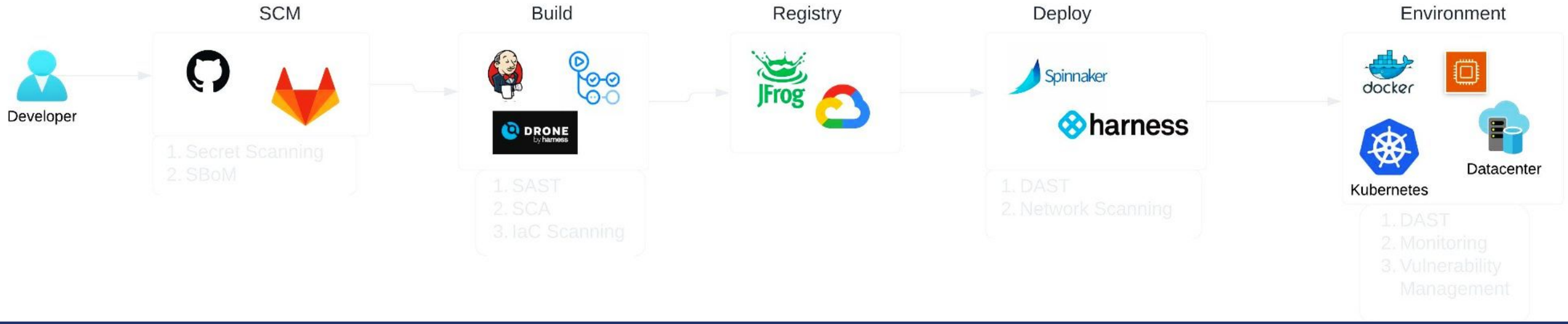
- Context
- Attack Surface
- Methodology
- Introduction to CICDGuard
- Architecture and Workflow
- Demo

CONTEXT



- An oversimplified version of CICD ecosystem
- Security **IN** CICD - as we all know it
- Wonderful topic but this session 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

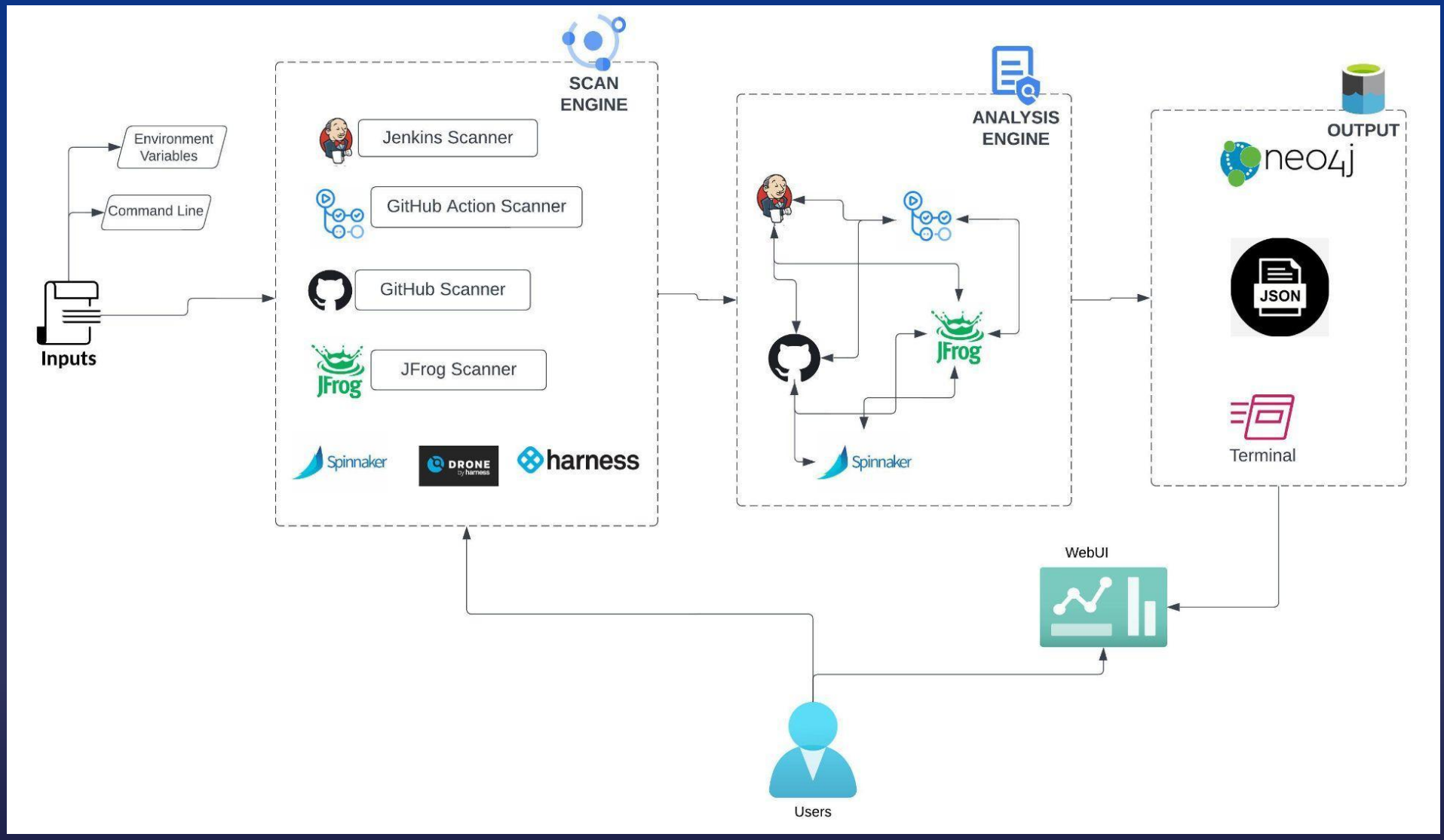
METHODOLOGY

1. Focus on making technologies secure and robust, by default - Everyone needs to contribute in that
    a. Implementing vetting process on organization level
    b. Working with provider proactively to resolve the vulnerability
2. How well are we implementing the solution in our environment
    a. Do we have default settings disabled/secured?
    b. MFA enabled for all users for all applications?
    c. Are we using up-to-date/non-vulnerable plugins/apps/actions?
3. 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…

ARCHITECTURE & WORKFLOW



DEMO

[varchashva/CICDGuard](#)

@IAmVarchashva

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