Secure DevOps
Summit & Training
Threat Modeling-as-Code with ThreatPlaybook
Nithin Jois
sans.org/SecureDevOps
Yours Truly

- Solutions Engineer, we45
- DevOps, Security Engineering and DevSecOps for Clients
- Co-lead Developer - ThreatPlaybook
- Core-Developer - Orchestron
- Trainer - we45 Trainings and Workshops
Today's Session

• Some Issues we see with Threat Modeling as it’s done today
• The “as-code” movement and Threat Modeling’s role in it
• ThreatPlaybook and philosophy behind it
• Completing the Automation Cycle with ThreatPlaybook
And a few demos....

Demo Gods! Please let this work
Inspirations and Thanks

• Adam Shostack for breaking down Threat Modeling into a consistent and repeatable process
• Jonathan Marcil for Threat Modeling Toolkit
• Chris Gates for metta (carnal0wnage) => MITRE ATT&CK Framework
Current Problems with Threat Modeling
Some Problems

LIVE LONG AND
READ THE SECURITY POLICY

I JUST CAN'T KEEP UP
WITH ALL THIS ACTIVITY!

"HIGH LEVEL"
What’s worse...

- Threat Modeling is usually undertaken at the beginning a project and then forgotten - Updated annually/not at all (usual case)
- Not integrated with the Agile SDLC
- No link with user stories/functionality
- Security teams often just do it themselves
- Threat Modeling (for many) has become largely about generating Diagrams and not actually modeling Threats
And Yet...

MUST START ENGINEERING

WITH THREAT MODELING YOU SHOULD
But....
The Result

NO VISIBILITY

WITH ENGINEERING & MANAGEMENT
Benefits of Threat Modeling

- Ties into enterprise risk/compliance
- Input for security incident management
- Input for table top exercises
- Defining mitigation strategies
- Input for static analysis
- Security in requirements & design
- Attack modeling: security test cases/abuse cases
- Security risk benchmarking
Security in DevOps

Plan

Code

Build

Test

Release

Deploy

Operate

Monitor

Threat modeling

SAST
Security - Composition

DAST
IAST

Security in
IaC

Security monitoring
& attack detection

Threat Modeling Inputs - Go here!
On the other hand....
“Spec” based Systems

• Frameworks that allow users to define deployments/delivery without having to write complex code

• Abstract the complexity away from the user

• Increase Cross-Functional workflows

• Make everything “As-Code”
Application Delivery by Spec
Basically...

- Continuous Integration - Code
- Infrastructure as Code
- Cloud Deployments as Code
- Container Orchestration as Code
- Security as Code
Our Philosophy for Threat Modeling

- We see Threat Models as “Playbooks” for Security
- More power to collaborative Threat Modeling
- Iterative Threat Modeling
- Manageable Threat Modeling
The Idea

- YAML Spec Based Orchestration Tools
- AppSec Automation
- Threat Modeling
Which is why...ThreatPlaybook!
The ThreatPlaybook way...
Build Threat Model
Functionality/User Stories
Abuser Stories
Threat Scenarios

Security Test Cases
Invoke Strategic Automation

DEVOPS
TO THE MAX
ThreatPlaybook (YAMLs) (define Threat Models)

ThreatPlaybook Robot Library

OWASP ZAP Robot Library

Other SAST/DAST/SCA Robot Library

Invoked in Robot Script (Automation)

Target App
Process - Threat Playbook

- Write Threat Models in YAML files:
  - Iterative and modular
  - Link (or not) to Security Test Cases
- Run Automation
- Generate Report with Vulnerabilities linked with Threat Models
Who can use ThreatPlaybook?

• Engineering Teams:
  • Develop and scale Iterative Threat Models
  • Run a Security Pipeline with Threat Modeling included

• Pentesting Teams:
  • Attack-driven Threat Modeling with Pentest Automation
Demo Time

Demo Gods! Please let this work
Hard Problems & Questions?

Look at our Iterative Threat Model + Linked Vulnerability Assessment Results!

Our Threat Models have Security Test Cases + Automation of those Test Cases in CI/CD!
Security Tools - API that we have developed

- **DAST**
  - OWASP ZAP
  - BurpSuit Pro
  - Arachni

- **SAST**
  - NodeJSScan
  - Brakeman
  - Bandit

- **Recon/Mapping**
  - Nmap
  - Wfuzz (Directory Bruteforce only)
  - SubList3r

- **SCA**
  - OWASP Dependency

- **Cloud**
  - Bucketeer
  - weirdAAL (Selected for Dev)
The future

- Objective: To make this the Kubernetes of Application Security
- Add a CLI for user interaction => Happening in final release 1.2
- Add capabilities for Trust Boundaries
- Look at a more comprehensive Diagramming API
- Contributors Welcome :)
Thank you!

Secure DevOps

Summit & Training

Twitter: @bondijois
Twitter: @we45
LinkedIn: bondijois
Github: github.com/we45

sans.org/SecureDevOps