# Maximizing the ROI of your Pentest

ISACA Denver
April 2025

Mark Hoopes

# Agenda

- Pentesting in the Testing Tool Landscape
- When is a pentest useful?
- What you can do to maximize value
  - Before the test
  - During the test
  - After the test
- How do you know if you got a good pentest?

Slides: https://www.merisec.com/blog

# Know your Bias

- Application Penetration Tester (12 years)
  - Network Penetration Tester (6 years)
  - Secure Development Trainer
- Web Development Project Manager (5 years)
- Major Incident Manager, Web hosting operator (6 years)

Slides: https://www.merisec.com/blog

# Pentesting in the Security Testing Landscape

# Definition of Terms

## What is a Penetration Test?

A simulation of a real-world attacker attempting to bypass security controls to access sensitive resources

- External Network – An assessment of your Internet facing controls
- Internal Network – An assessment of your Intranet controls (“Assume Breach”)
- Application – An assessment of the controls inside a specific application (web, mobile, desktop)
- Red Team – A long-term assessment of the entirety of your security controls
- “Continuous Penetration Test” == Vulnerability Scan

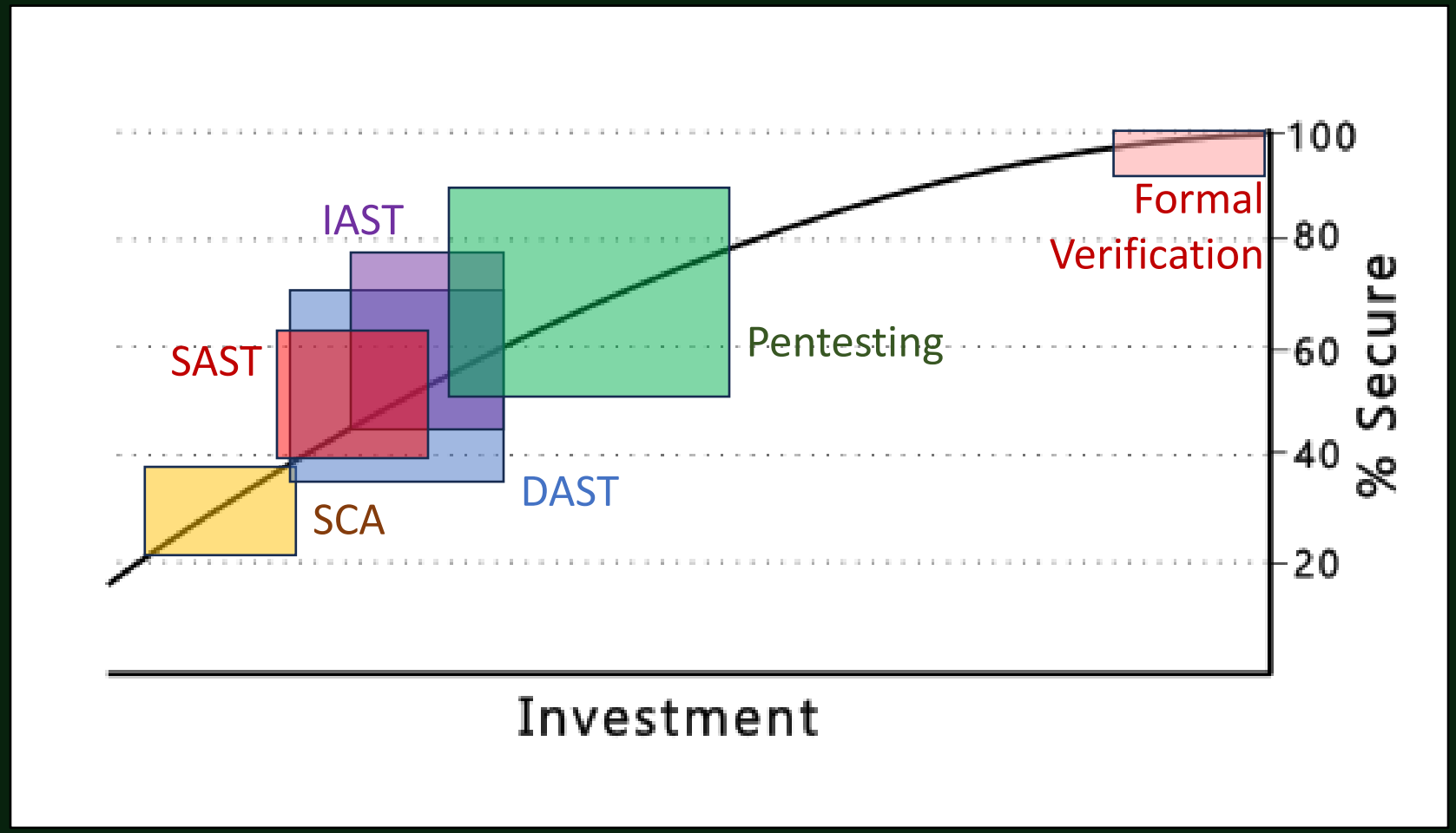
# Security Testing in the Application World

| Testing Type | Low End | High End | Strengths | Weaknesses |
|---|---|---|---|---|
| SCA | Finds published vulnerabilities | Finds published vulnerabilities | Simple to deploy | Blind to unique or unpublished vulns |
| SAST | Finds common patterns | Once finely tuned, cheaply finds coding errors | Low overhead to run. High potential for use across organization applications. | Tuning is arduous. Full code flow is resource intensive. Unlikely to find business logic and authorization issues. |
| DAST | Finds the most common vulnerabilities | Once tuned, finds a majority of vulnerabilities visible to users | Tests the actual attack surface. | Resource intensive. Spidering is error prone. Misses business logic and some authorization issues. |
| IAST | Finds common SAST and DAST vulnerabilities. | With exhaustive test cases, finds a majority of vulnerabilities | Tests the actual attack surface and underlying code. | Requires active use. Only catches BL and Auth issues if test case is written for it. |

# Security Testing in the Application World

| Testing Type | Low End | High End | Strengths | Weaknesses |
|---|---|---|---|---|
| Vulnerability Scanner | Finds externally visible published vulnerabilities | Finds user visible published vulnerabilities | Low overhead to run | Blind to unique or unpublished vulns |
| Penetration Test | Finds the kinds of vulnerabilities casual hackers will find | Finds the kinds of vulnerabilities expert hackers will find | Mimics your threat actors across all vulnerability classes. | Expensive and doesn't scale well. Relies on a sampling rather than an exhaustive search. |
| Red Team Engagement | Finds a single pathway through your defenses | Finds the gaps a minor nation state would find | Best simulation of a dedicated attacker | Expensive. Not focused on finding "all" weaknesses. |
| Formal Verification | No such thing. | Definitive proof of the security of a component of your environment | The only technique for certainty | Extremely expensive. Usable on small scopes only. |

# Mark's Anecdotal Graph of Application Testing Technique Effectiveness * ** ***

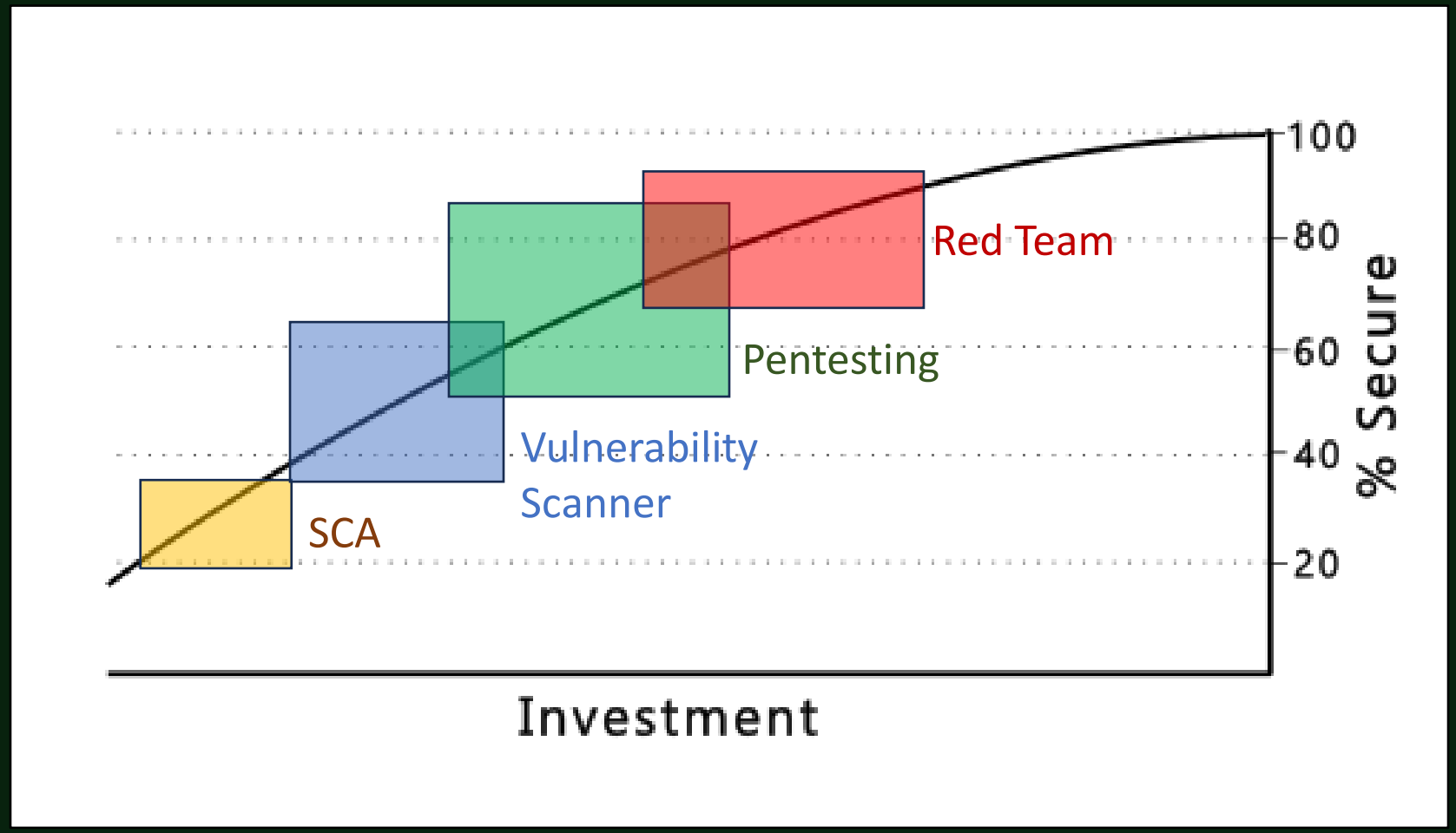



* Based only on experienced vibes

** Mixes purchase price and implementation costs

*** Does not reflect economies of scale across multiple applications

# Mark's Anecdotal Graph of Network Testing Technique Effectiveness * ** ***




* Based only on experienced vibes

** Mixes purchase price and implementation costs

*** Does not reflect economies of scale across multiple applications

# How Does AI Change The Landscape?


OR

Current AI is a co-pilot, not an auto-pilot

- Can the AI to tell you what your most critical business asset is?
- Can the AI tell you if a “super user” should be able to access that asset?

# Conclusion 1 – When to get a PenTest

1. If you're only going to do one thing…

2. If the compliance framework says so
   - If the customer says so

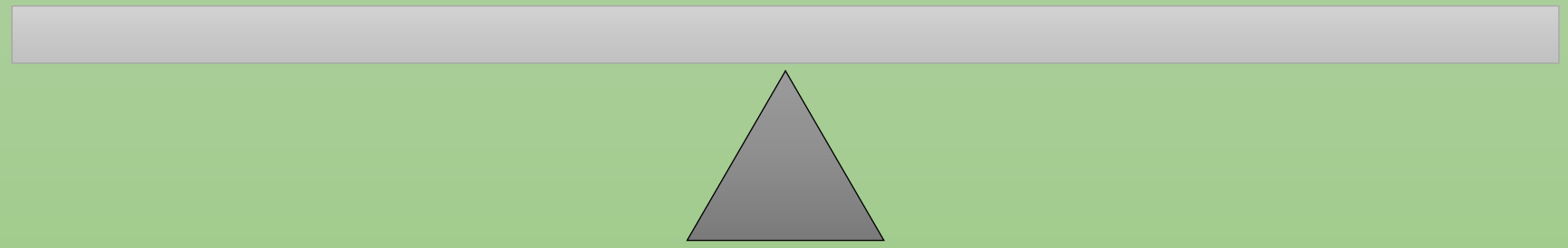
3. As a capstone to a mature secure development lifecycle
   - Red Team only if you think you've blocked everything else

# Preparing for Success

# How much do you tell your pentester?

A REAL attacker won't have any starting information

A pentester's time is limited



# Pentest Preparation Checklists

## Application

- ❑ Context
  - ❑ Business Purpose
  - ❑ Technology Stack
  - ❑ 3rd Party Services
- ❑ Scope
  - ❑ Source Code?
- ❑ Test Environment
- ❑ Roles / IDs
- ❑ Rules of Engagement
  - ❑ WAF

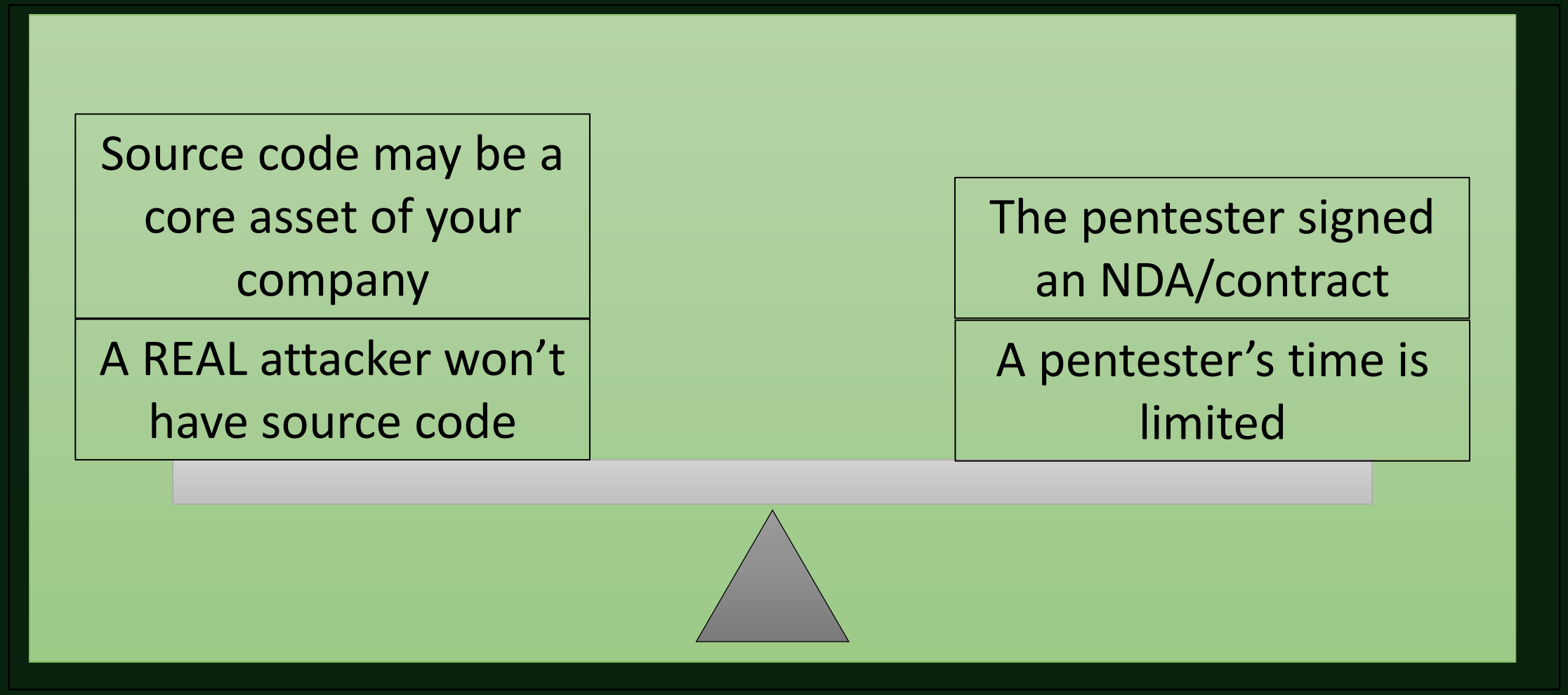
## Network

- ❑ Context
  - ❑ Business Purpose
- ❑ Scope
- ❑ Access Requirements
- ❑ Rules of Engagement
  - ❑ Out of scope systems
  - ❑ Out of scope techniques

# Defining the Scope

- External Network
  - A list of IP addresses, subnets, and/or domains to be tested
  - Direction to perform OSINT to discover IPs and assets
- Internal Network
  - What is NOT in scope.  Both techniques and specific devices
- Application
  - Web: Starting URL(s), in-scope/out-of-scope supporting services

# To give source code or not to give source code?



# Story Time – Testing in Production

| | |
|---|---|
| /api/v1/campaigns | List of all marketing campaigns for this client |
| /api/v1/campaign/7 | Details of marketing campaign 7 |
| /api/v1/donors | List of all client donors |
| /api/v1/donor/1001 | Details of donor 1001 |
| /api/v1/donation/4277 | Details of a specific donation, including CC info |
| /api/v1/donations | All donations for a client - CC INFO UNMASKED |

Was this a data breach?

# Testing in production?

## Production

- Typically lower setup effort
- Represents the actual surface an attacker will go against
- Might impact clients
  - Sometimes attacks live-on past the pentest
- If there is a vulnerability, you may have just caused a data breach

## Test/UAT/Dev/etc.

- May require setting up network access (VPN) to access
- May require adding realistic data
- May occupy an environment normally used for other things

# Testing in Production – Network Style

- You really don’t have a choice.
- Make sure there is language in the contract (SoW) that ensures the testing company will protect any data discovered.

# Pentest Preparation Checklists

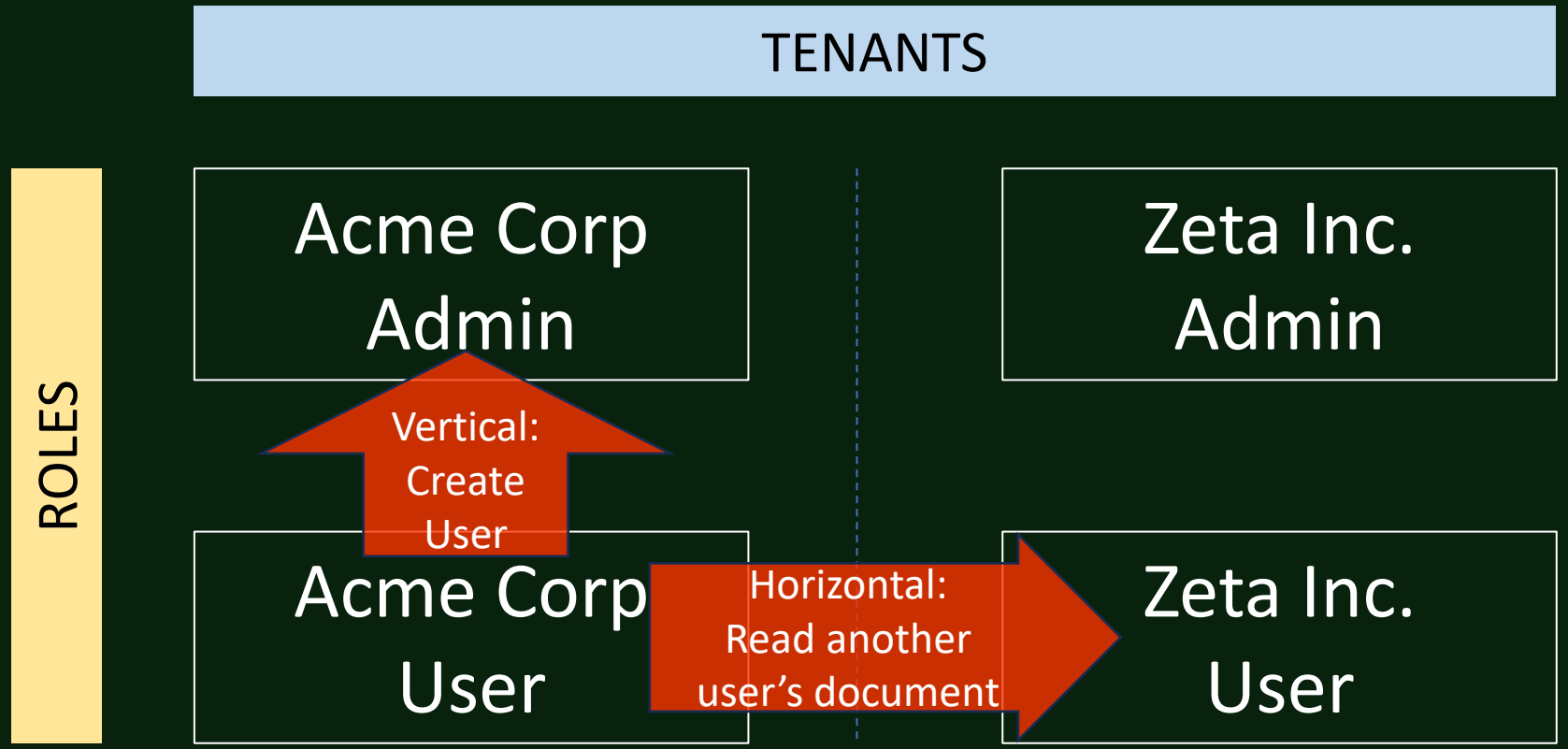
## Application

- ✓ Context
  - ✓ Business Purpose
  - ✓ Technology Stack
  - ✓ 3rd Party Services
- ✓ Scope
  - ✓ Source Code
- ✓ Test Environment
- ❑ Roles / IDs
- ❑ Rules of Engagement
  - ❑ WAF

## Network

- ✓ Context
  - ✓ Business Purpose
- ✓ Scope
- ✓ Access Requirements
- ❑ Rules of Engagement
  - ❑ Out of scope systems
  - ❑ Out of scope techniques

# Why to testers want so many IDs?

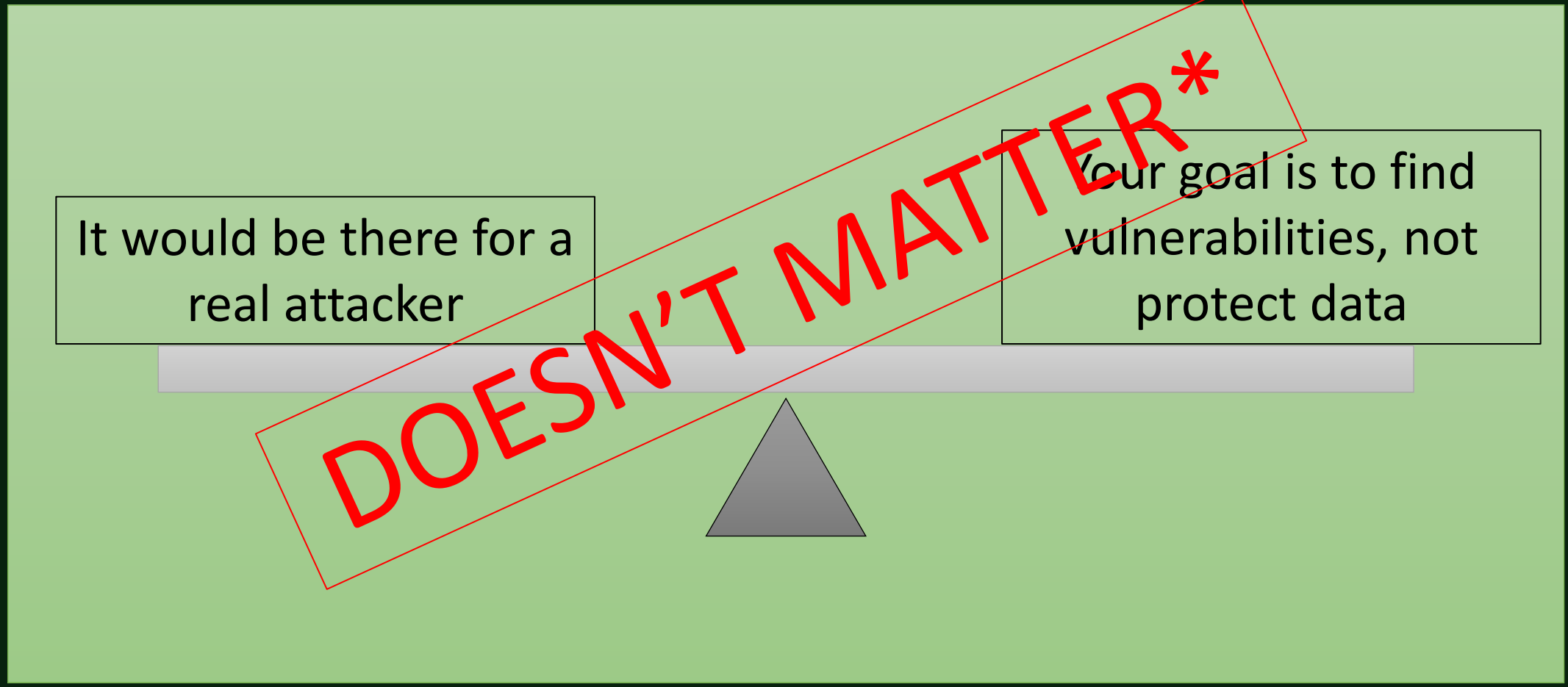


# Rules of Engagement

Common Elements

- ❑ Time of day
- ❑ Out-of-scope servers / domain
  - ❑ 3rd Party
  - ❑ Temperamental
  - ❑ Soon to be out of service*
- ❑ Out-of-scope techniques
  - ❑ Denial of Service
  - ❑ Social Engineering

# Do you turn off your WAF?



# Quick Example

Search

Capybara

NO DOCUMENTS FOUND

Search

Capybara” or “a”=“a

BLOCKED BY XYZ WAF

Search

Capybara” or “z”=“z

838 DOCUMENTS FOUND

# Set the Stage for Your Team

A mandatory security review of our application will be taking place from April 1 to April 10. As a reminder, any high or critical findings discovered will need to be remediated before this version can be released.

To support our common goal of protecting client and company data, an independent security review will be performed from April 1 to April 10. We will prioritize fixing any high or critical findings before we put the code in production.

# During the Test

# Keep the Test Environment Available

# Expect Questions

How does the "[*name*]" feature work?

Should [*user role*] be able to [*modify object*]?

Is [server] in scope?

Exploiting this vulnerability might [*potential risk*]. Should I try it?

# Expect Updates

| | |
|---|---|
| Critical / High Finding Updates | • Typically delivered “immediately” after discovery<br>• Should include vulnerability, impact, and replication steps<br>• Should be delivered over a secure channel |
| Periodic Updates | • Typically delivered between daily and weekly<br>• Should include a high-level description of what has been tested<br>• May include outstanding questions or roadblocks |
| Testing Start / Stop Updates | • May be useful if a SOC is ignoring alerts |

# After the Test

# Readout Calls – It's your dime

- Invite anyone who might have a question
  - The people who are going to have to fix the problem
  - Grumpy executive who thinks a problem is worse than it is
- Ask about what your tester has seen at similar companies
  - Resolution strategies, resolution timing, design decisions
- Report phrasing can be adjusted
- Report severities can be adjusted…sometimes


# Requesting Finding Severity Changes

- Good Reasons
  - An external business process mitigates the impact
  - There is a control that blocks the attack path that was not part of the test
    - IF it can be tested now, otherwise it just gets added as a note in the finding

- Bad Reasons
  - We can't pass compliance with anything higher than a medium severity
  - We can't fix it within the deadline that comes with that severity
  - We couldn't reproduce the exploit

# How do you know if you got a good pentest?

- Communication with pentester
  - They should know specific details of your environment / application
- Level of detail in the report
  - Findings should describe specific features
  - Replication steps should be included (not just screenshots)
  - Recommendations should be tailored
  - If few findings, the executive summary should describe

# Sample Executive Summaries

“ Coyote Corp. conducted a security review of the Acme Corp. internal network between April 1 and April 10, 2025. No vulnerabilities were found. ”

vs

“ Roadrunner Inc. conducted a security review of the Acme Corp. internal network between April 1 and April 10, 2025. The network demonstrated strong security practice including patch management, egress filtering, and network segmentation. No vulnerabilities were found.

Roadrunner attempted to capture and replay Windows hashes observed on the network, however SMB Signing was required on all Windows systems, blocking this common attack. ”

# Sample Finding

“Cross-site Scripting occurs when unvalidated input is inserted into a webpage. Coyote Corp was able to save a malicious payload into the user profile page as seen in the screenshot below.”

vs

“JavaScript Injection (commonly known as Cross-site Scripting) occurs when user-controlled input is incorporated into a webpage without proper validation or encoding. Roadrunner was able to save a JavaScript payload in the “title” field of the user profile page, and that payload was executed on the user profile, user search, and user management pages (viewable only by the administrator).

To replicate the vulnerability follow these steps…”

# Sample Replication Steps

“ Evidence
The screenshot below shows the Cross-site Scripting payload executing.”

vs

“ Replication Steps
1. Log in to the Acme site as a user in the “viewer” role.
2. Click on the profile picture in the top right and select “My Profile”
3. Click “Edit Profile” at the top of the page.
4. In the “Title” field, enter the following payload
   “><img src=bad onerror=“alert(‘XSS in Title’)”><x a=“

…

”

# Conclusions

**When to Pentest?**

- As your one security investment
- When required by customer or compliance
- As the capstone to your security program

**How Do You Know If You Got A Good Pentest?**

- The report should contain specific details about your environment
- Failed attacks and strong controls should be include in the executive summary

**How Do You Maximize the ROI Of Your Pentest?**

- **Eliminate barriers between the pentester and the vulnerabilities**

# Thank You!

Slides
https://www.merisec.com/blog

Mark Hoopes
mark@meristeminfosec.com
https://www.linkedin.com/in/markhoopes/