How to run a pentesting engagement

OWASP Day NZ
6 September 2024

AppSec
NZ
appsec.org.nz

OWASP
NEW
ZEALAND
owasp.org.nz

AUT
UNIVERSITY
TE WĀNANGA ARONUI O TAMAKI MAKAU RAU

BASTION

SECURITY GROUP

2°

DATACOM

aws

84.

PentesterLab

plexure

VERACODE

*** Matt Tompkins ***

Security Consultant (secure dev, security architecture, governance / risk / compliance)

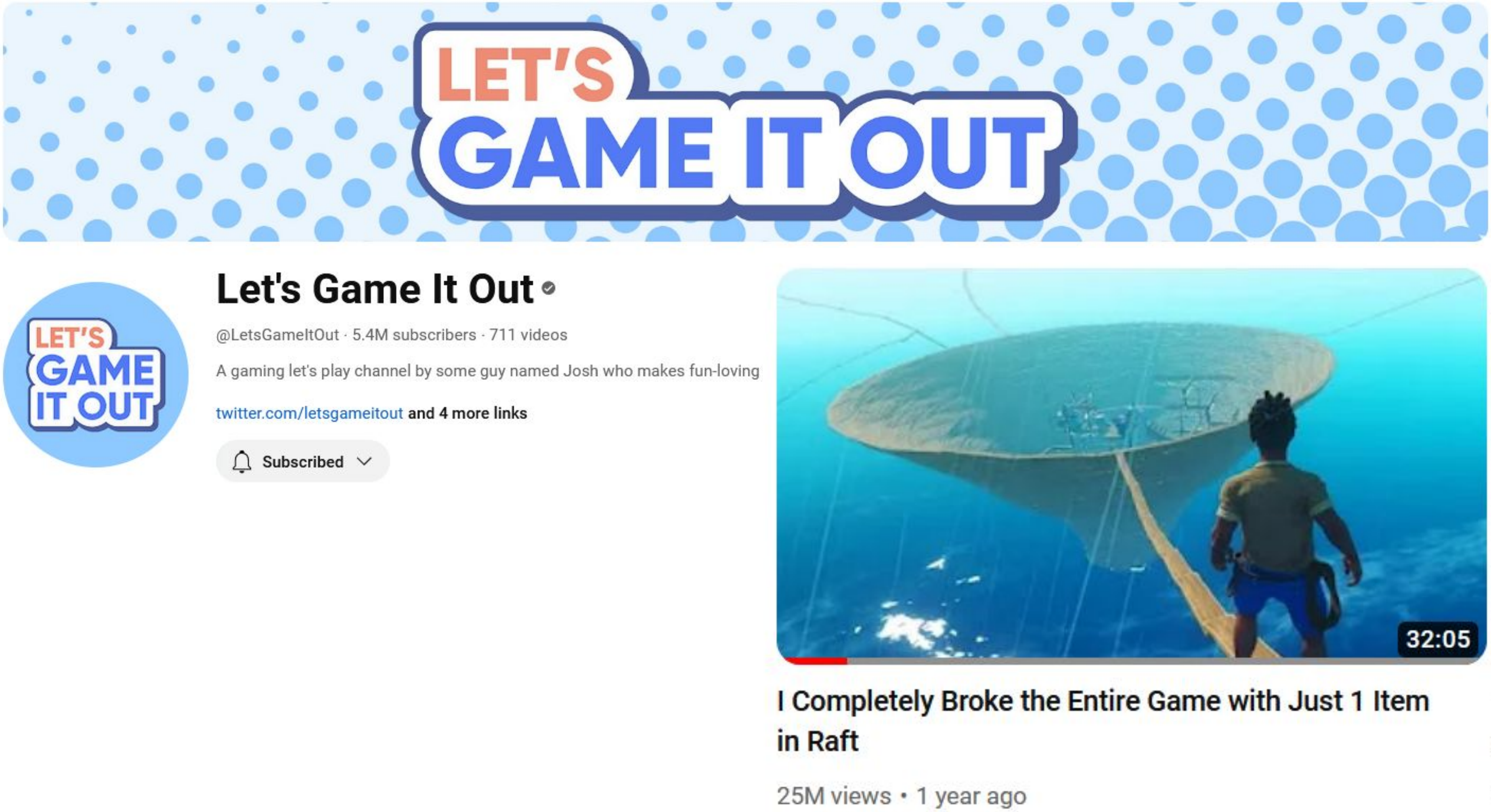
threads.net/@iobreakers

linkedin.com/in/matt-tompkins

* Agenda *

- What is hacking?
- What is pentesting?
- Debate whether pentesting is even a good idea
- Eight steps in a pentesting engagement

What is hacking?



Let's Game It Out

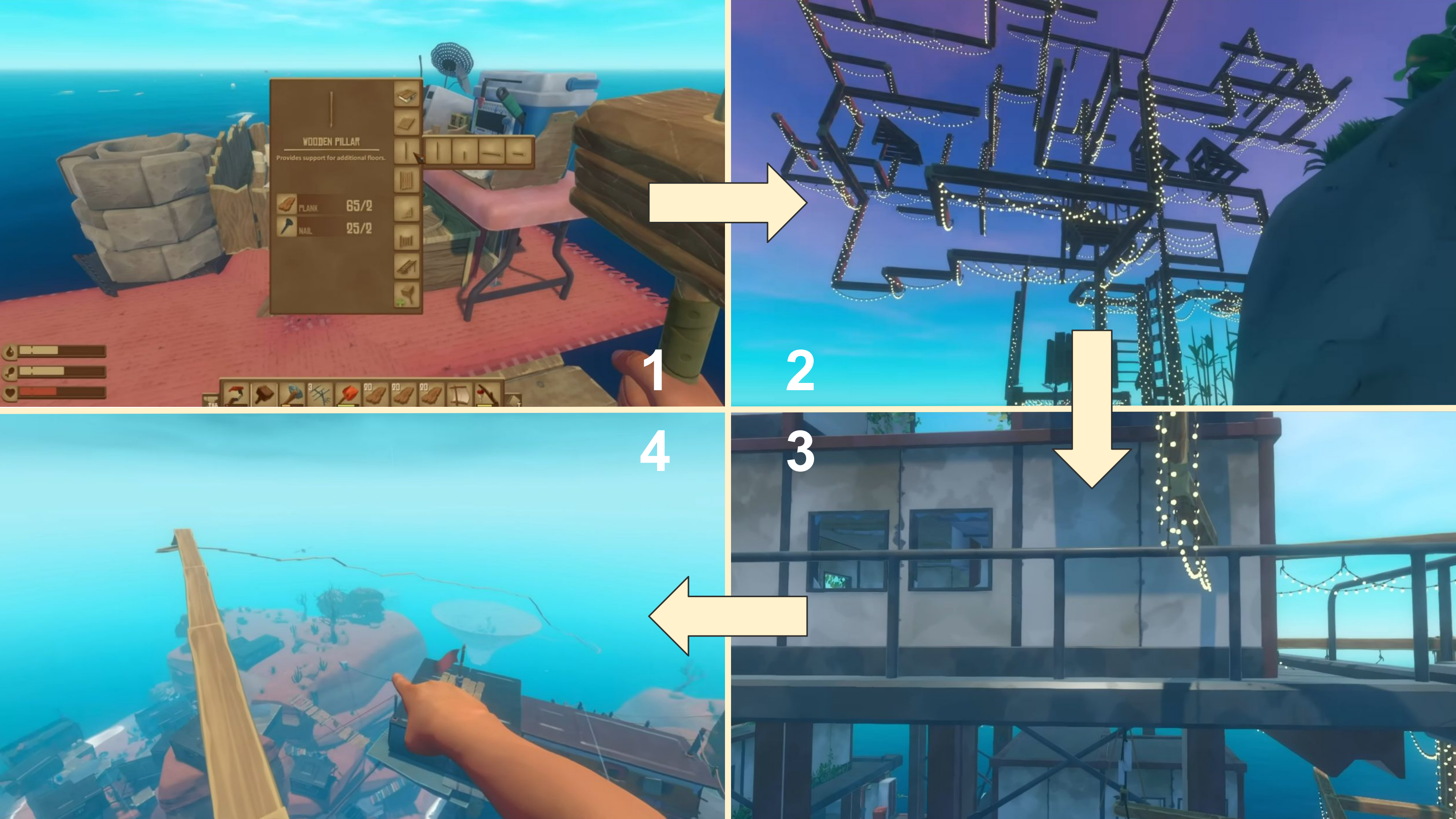
@LetsGameItOut · 5.4M subscribers · 711 videos

A gaming let's play channel by some guy named Josh who makes fun-loving

twitter.com/letsgameitout and 4 more links

I Completely Broke the Entire Game with Just 1 Item in Raft

25M views • 1 year ago

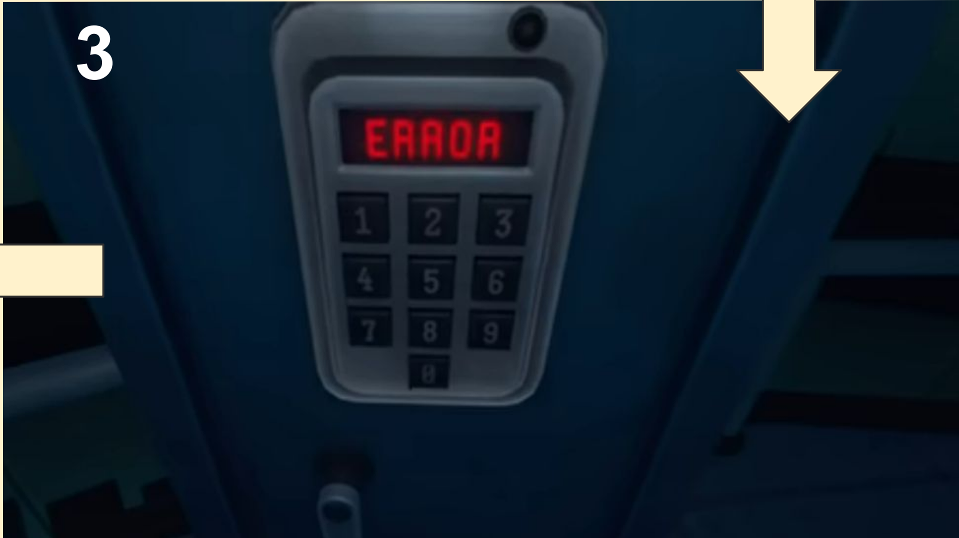

WOODEN PILLAR
Provides support for additional floors.
PLANK 65/2
NAIL 25/2
1
2
3
4

9 hours later...

1

2

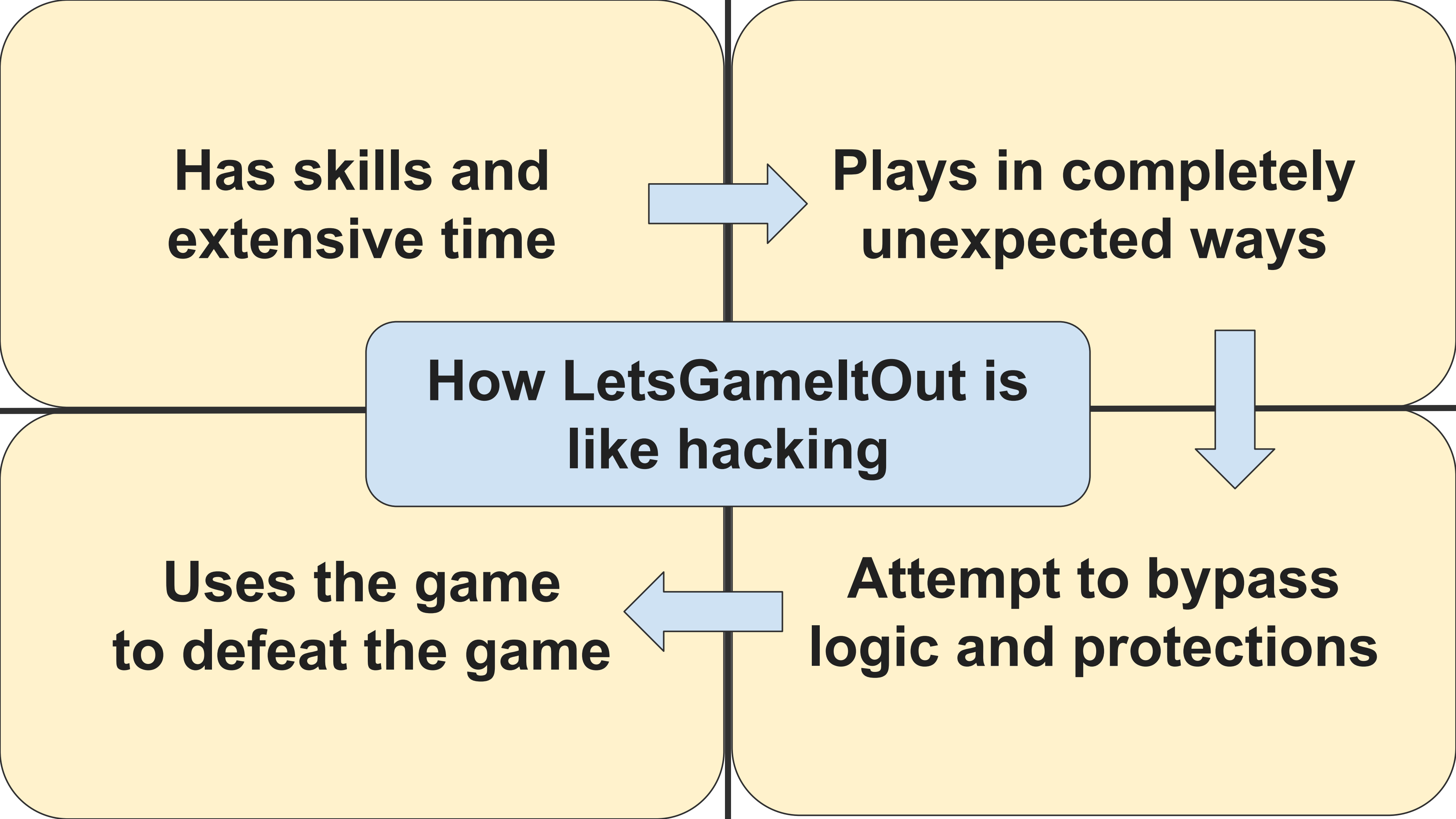
3



4

5,960+ tries later...

WATCH OUT FOR BEARS
1
2
3
4



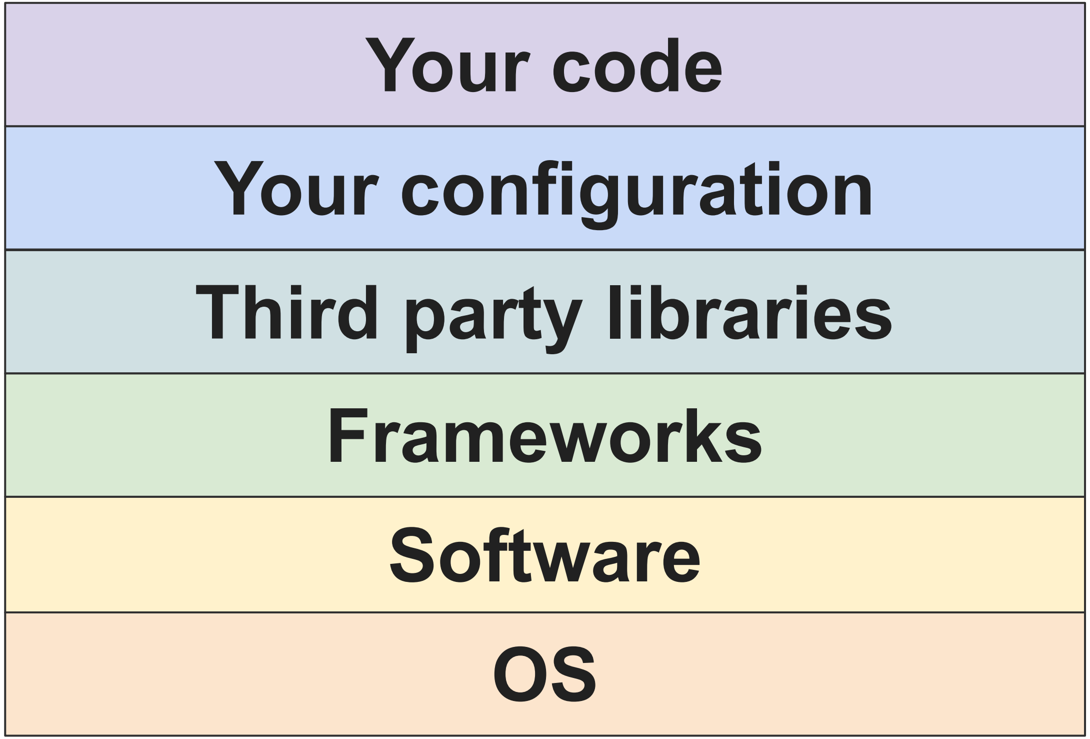
How LetsGameItOut is like hacking

Has skills and extensive time

Plays in completely unexpected ways

Attempt to bypass logic and protections

Uses the game to defeat the game



Your code

Your configuration

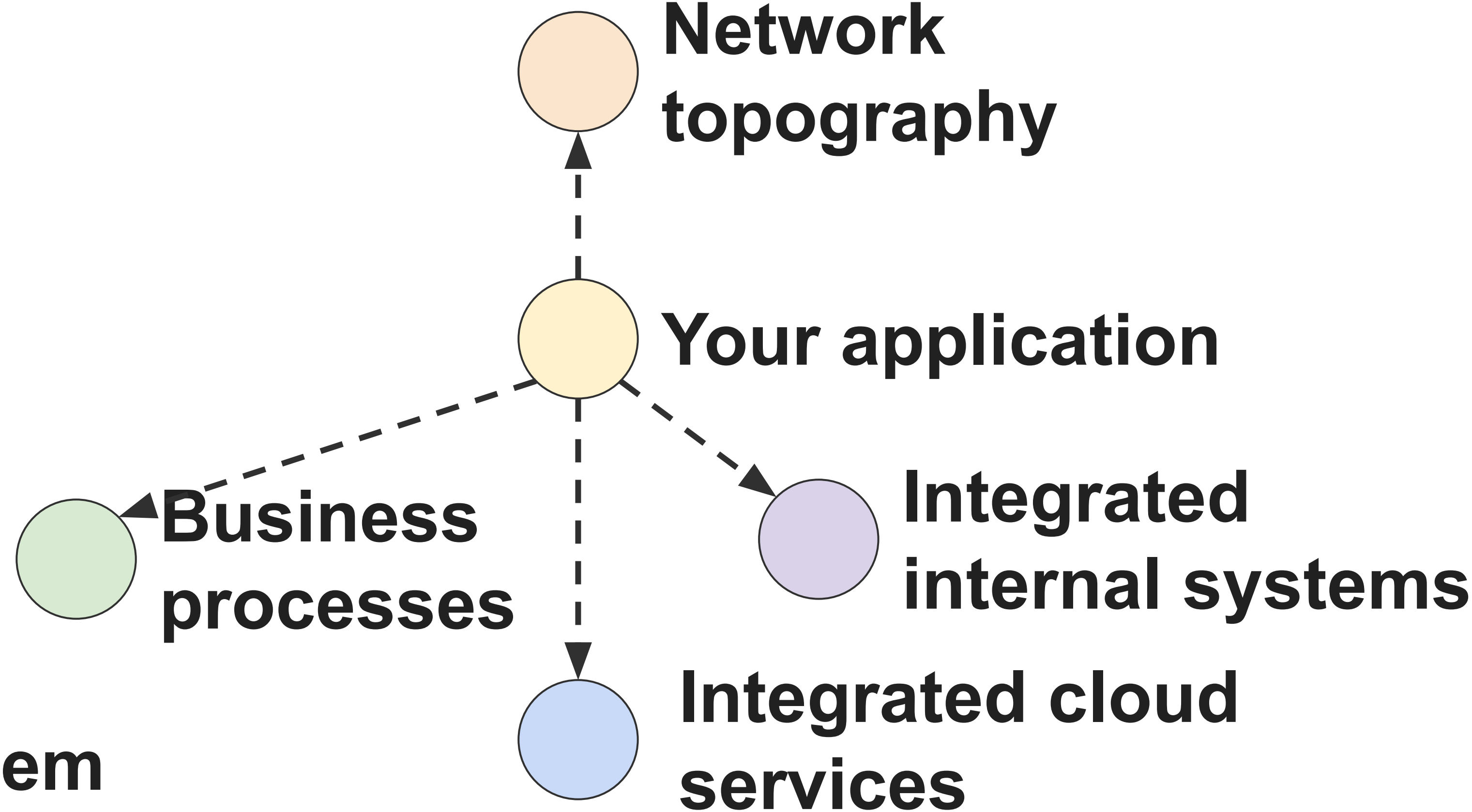
Third party libraries

Frameworks

Software

OS

Vulnerabilities in your application



Network topography

Your application

Business processes

Integrated internal systems

Integrated cloud services

Ecosystem

What is pentesting?

pci Security Standards Council®

PCI DSS: Definitions

Penetration tests simulate a real-world attack situation ... A penetration test differs from a vulnerability scan, as a penetration test is an active process that usually includes exploiting identified vulnerabilities.

§11.4.1 Methodology

A penetration testing methodology is defined, ... and includes:

- Testing from both inside and outside the network.

- Application-layer penetration testing.

- Network-layer penetration tests that encompass all components that support network functions.

§11.4.2 §11.4.3 Penetration testing

§11.4.2 Internal penetration testing.

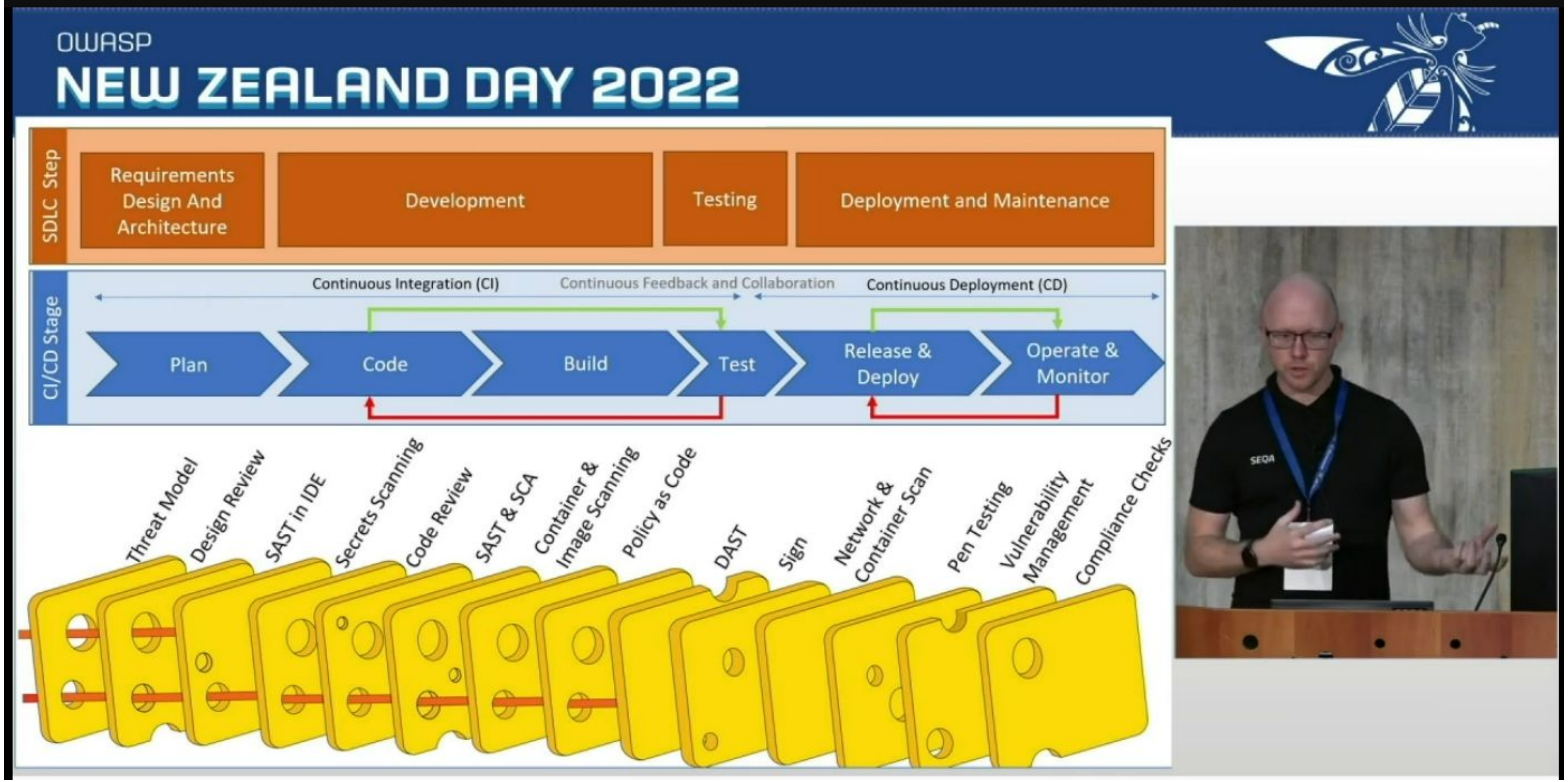
§11.4.3 External penetration testing.

- Per the entity's defined methodology.

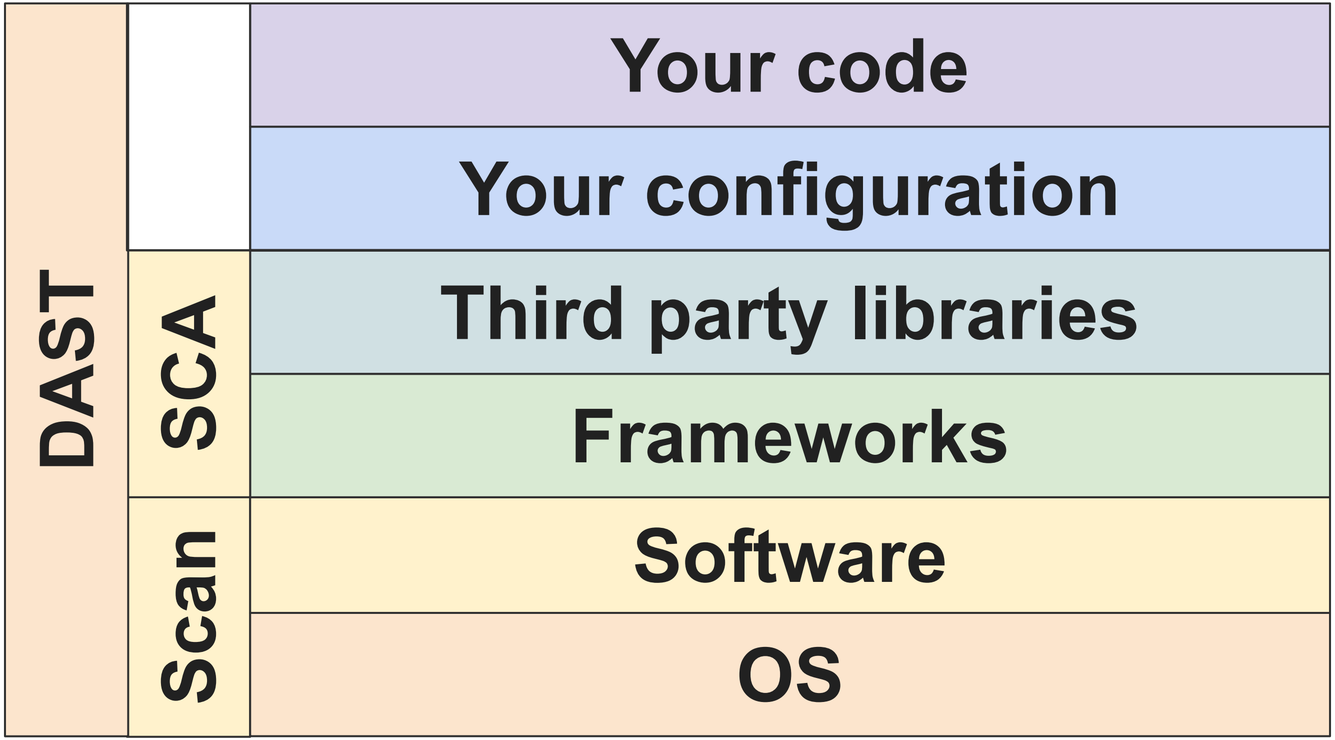
- By a qualified internal resource or qualified external third-party.

- At least once every 12 months.

- After any significant infrastructure or application upgrade or change.



Application Security Cheese - Steve Esler



DAST

SCA

Scan

Your code

Your configuration

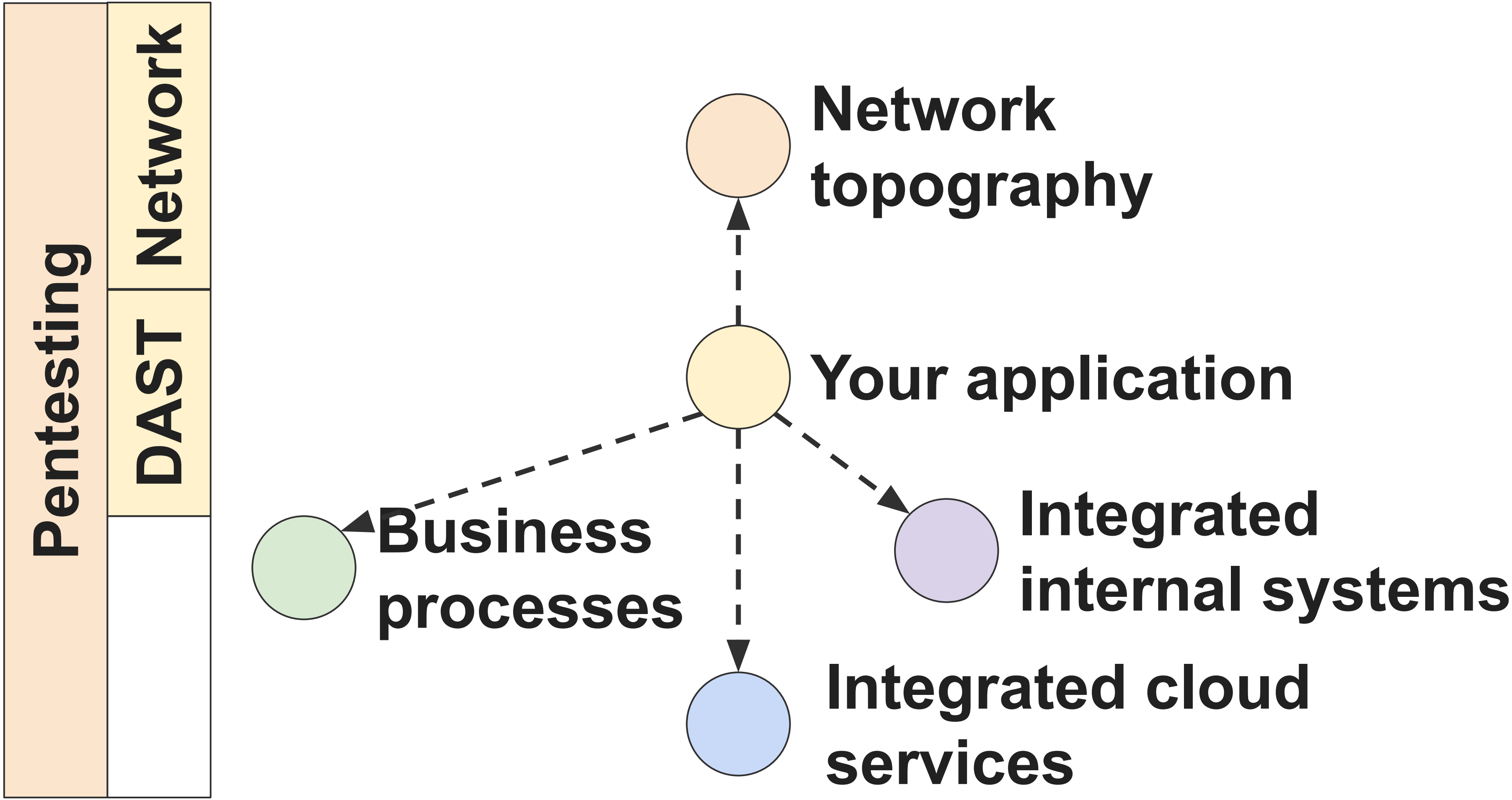
Third party libraries

Frameworks

Software

OS

Vulnerability scans for your application



Pentesting

Network

DAST

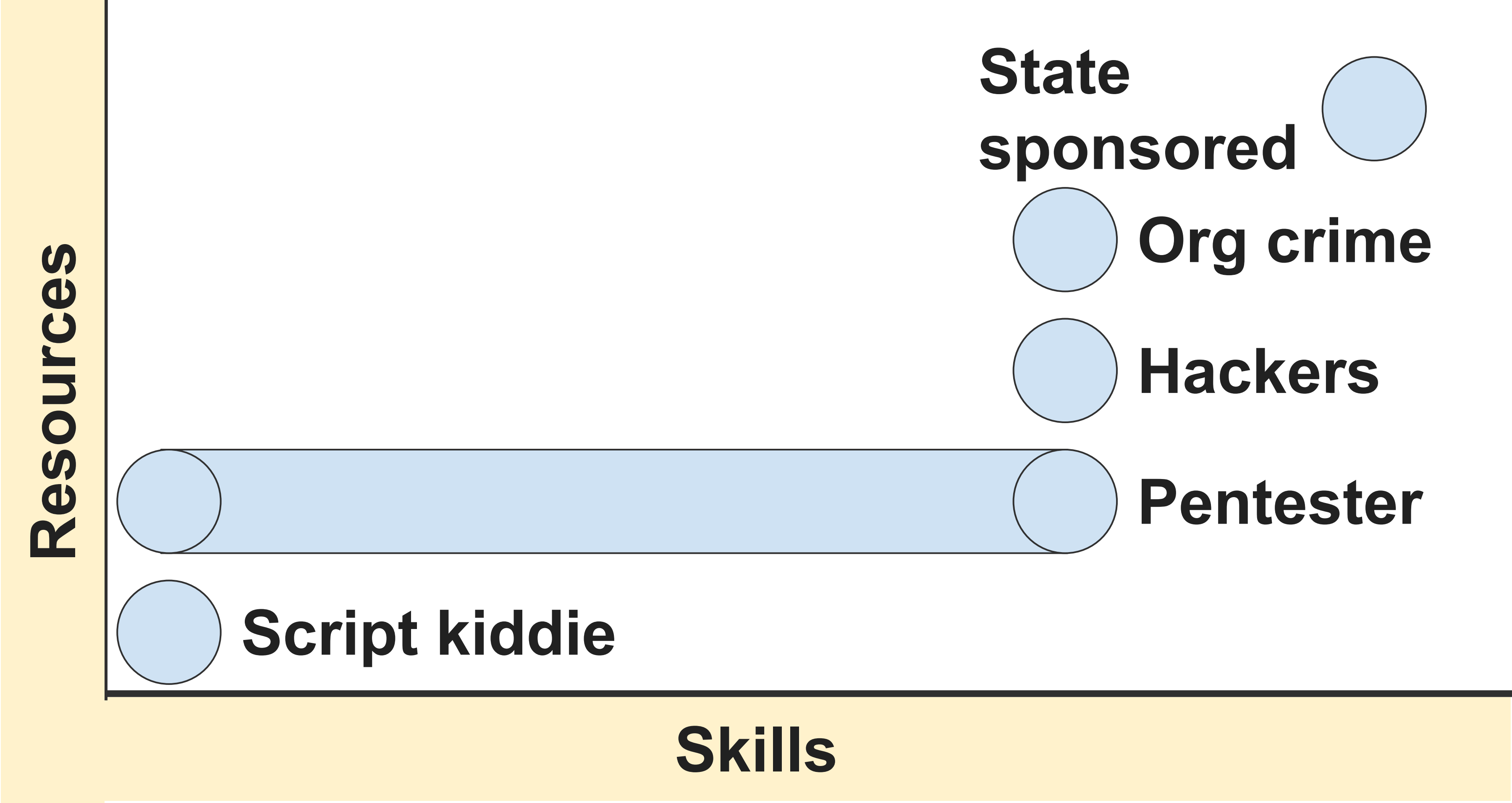
Network topography

Your application

Business processes

Integrated internal systems

Integrated cloud services


State sponsored
Org crime
Hackers
Pentester
Script kiddie
Resources
Skills

Debate whether pentesting is even a good idea

IRL

"Pentesting is like a relic from a bygone era"

Objection:

- Pentesting is "point in time".
- Pentesting is not "Agile".
- Financial cost of pentesting every 3, 6, or 12 months.
- Occupies Test / UAT.
- Pentesting is timeboxed, hackers are not constrained.
- Pentesting is does not provide 100% certainty.

Response:

- Pentesters use tactics, techniques, and procedures, like a friendly hacker.
- Pentesters look at your application as a whole.

"We already have secure dev practice"

Objection:

- We scan for all network and application vulnerabilities.
- We follow "Secure/Privacy by Design".
- We have change processes.
- We have the best developers with the best training.

Response:

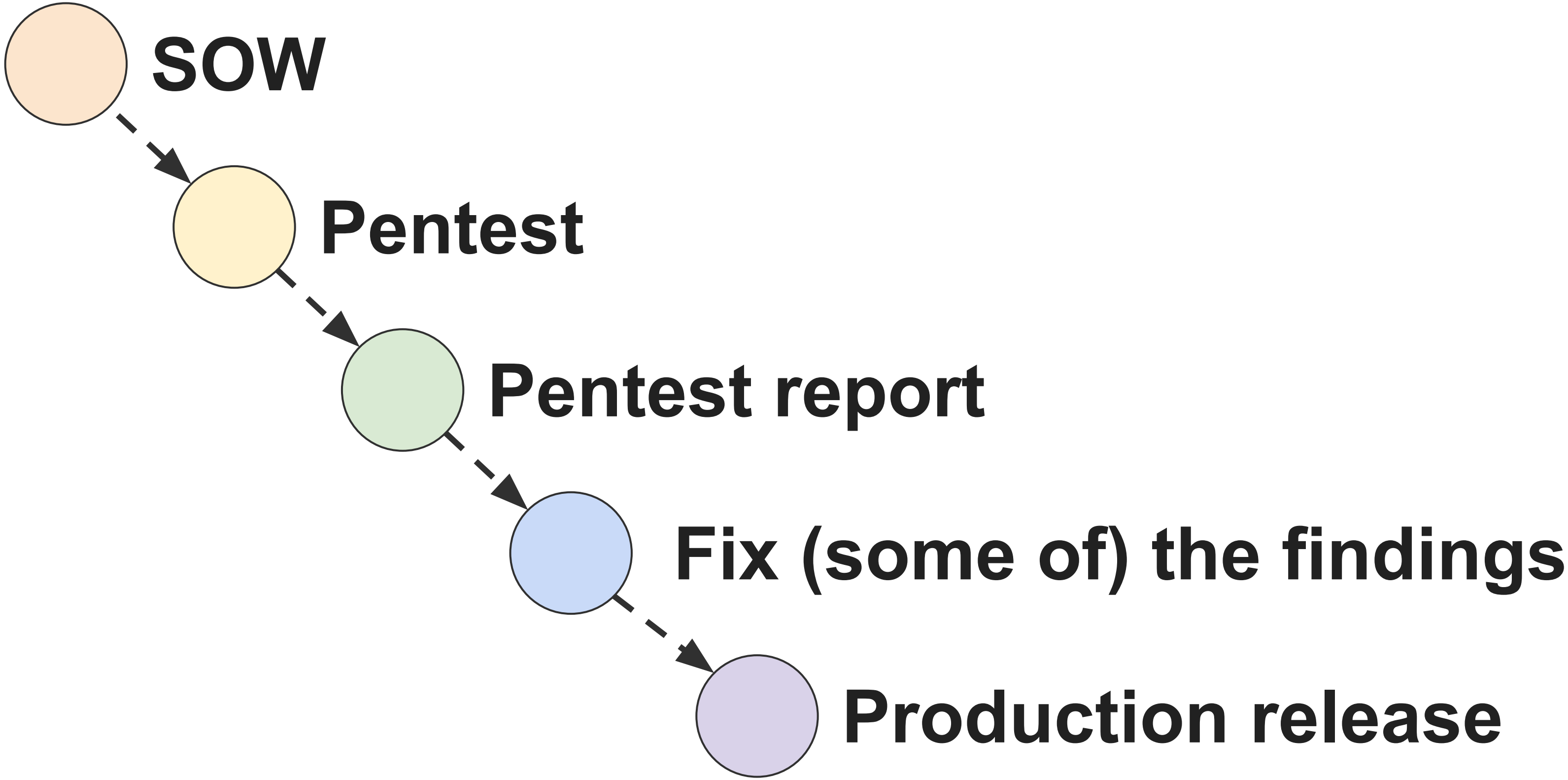
Pentesters are experts that look for vulnerabilities:

- More than the OWASP Top 10.
- The business logic and business processes.
- Bespoke code with unusual corner cases.
- Chain multiple vulnerabilities.

Some benefits of pentesting

- Test the risks and threats for your application.
- Test both inside and outside your network.
- Validates your secure dev practices.
- Validates your network and application vulnerability scanning.
- Prove it would take more than a Script kiddie to attack your application.

Eight steps in a pentesting engagement



SOW

Pentest

Pentest report

Fix (some of) the findings

Production release

Timelines and planning

IRL

Step 1: Talk to Security Ops

Your Security Ops team might just be "a security person", but they can provide:

- Technical considerations for access or setup.
- Scheduling considerations and conflicts.
- Would like to hear all about your application.

If you don't tell Security Ops, then it looks like you are really being hacked.

IRL

Step 2: Scoping your pentesting engagement

Out of scope:
- Hosting provider and cloud services.
- Technical controls, like WAF, IDS / IPS.

In scope:
+ How much of your application to pentest?
+ How much of the network topology?
+ How you integrate with, communicate with, and configure integrated internal systems and cloud services?

Step 3: Get the SOW [1/2]

Timeline: 2-8 weeks before engagement start date.

Shop around a few security companies:

- Let them know requirements, eg. set budget or timeline.
- The rule is not "the bigger the security company, the better the service".
- Agree on the scoping and scheduling.
- Legal / management to review and get sign-off.

Step 3: Get the SOW [2/2]

Provide information up front:

- Some business context.

- Security standards and risks / threats of concern.

- What is your application and how big is it?

- What's in the box and what is it integrated with?

- Security services: System hardening / config review, pentesting, and grant access to source code to make the most of the engagement.

IRL

Step 4: Prep for pentesting [1/2]

Timeline: 1-3 weeks before engagement start date.

Agree to changes to the SOW written in email.

Provide access to documents:

- Architecture and design documents and wiki pages.
- Repositories for source code, IaC, and config files.
- List of URLs / IPs for all in-scope endpoints.

Step 4: Prep for pentesting [2/2]

Environment as Production-like as possible:

- Deploy the latest release candidate and DB backup.

- Ensure integrated systems are up and running.

- Turn technical controls to detect / passive mode.

Access:

- Multiple user accounts and service accounts.

- Physical access for on-site engagements.

Step 5: Support during pentesting [1/2]

Timeline: 1-4 weeks to complete the pentesting.

Day 1: Meet with the pentesters to discuss:

- The SOW, Scoping document, and agreed changes.
- Walk-through of architecture and design.
- Security standards and risks / threats of concern.

Step 5: Support during pentesting [2/2]

Change freeze to the environment:

- No changes to code, no infrastructure deployments.
- No changes to DB structure or data.
- Ensure integrated systems remain up and running.

Provide points of contact:

- Resolve access issues and answer any questions.
- A technical lead, an architect, and a security person.

Step 6: Post pentesting

Timeline: 3-10 days until the report is prepared and published.

Release the environment:

- DB restore.

- Resume changes to the environment.

- Turn technical controls back to protect / block mode.

Step 7: Receive the pentesting report

Timeline: 2-8 weeks ahead of Production release.

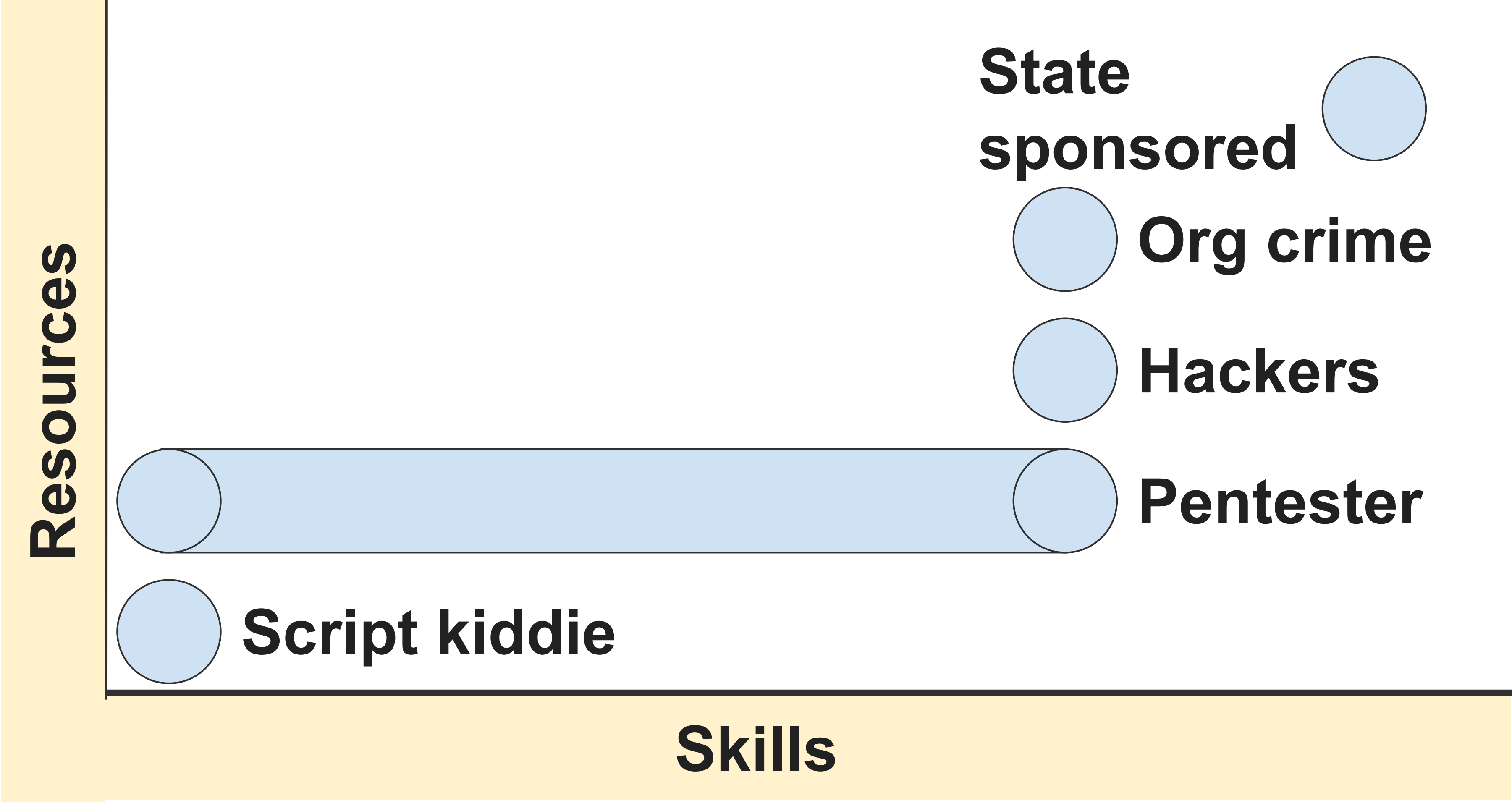
Pentesting report contains findings:

- Description of the finding.

- How to identify or reproduce the finding.

- Description of the risk rating.

- Recommendations how to fix the finding.


State sponsored
Org crime
Hackers
Pentester
Script kiddie
Resources
Skills

Step 3 (SOW): Top people ...

Step 5 (Pentest): Runs network vulnerability scan.

Not so good

1. Executive summary.
2. Results from a single scanning tool.
3. Multiple Low and Information risk findings.

Okay

1. Executive summary with business context.
2. Varied network and application findings.
3. Multiple Medium and Low risk findings.

Great

1. Executive summary with business context and security standards.
2. Chained, varied network and application findings.
3. Multiple High and Medium risk findings.

IRL

Step 8: Fix (some of) the findings

- Team meeting to review findings and prioritize what will be fixed prior to Production release.
- Compliance requirements might mandate that you fix all Medium risk findings prior to Production release.
- Allow time to perform root cause analysis, perform upgrades, and implement recommendations.

* My next presentation *

How to have a grown-up conversation about security risk and vulnerability management.

* Matt Tompkins *

Security Consultant (secure dev, security architecture, governance / risk / compliance)

threads.net/@iobreakers

linkedin.com/in/matt-tompkins