eficode

Threat Modelling in DevOps Platforms

And an investigation into the STRIDE threat model over time.

Darren Richardson

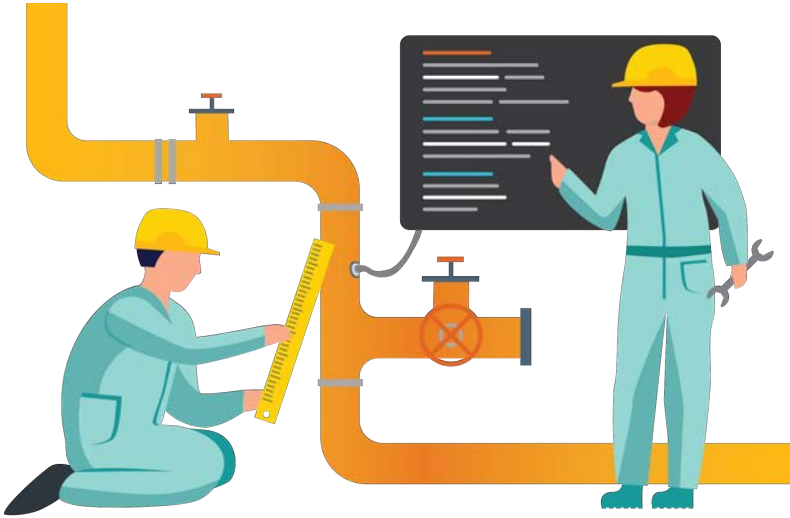
eficode

What is Threat Modelling?

eficode

What it is?

Threat Modelling works to identify, communicate and understand potential (cyber)security threats of a given application or platform.



eficode

What it isn’t.

Restricted.

Apply it anywhere to map threats!

eficode

Three Aspects of Threat Modelling

Map and diagram the dataflow

Before you can understand threats, you need to know your system landscape.

Break it down to trust boundaries

Understand control, ownership, and trust of data.

Apply the threat model

Apply the model to generate your list of threats.

[Threat Modelling is] an engineering technique you can use to help you identify threats, attacks, vulnerabilities, and countermeasures that could affect your application

Security Development Lifecycle

- Microsoft

eficode

STRIDE Analysis

And why it’s pretty good.

What is STRIDE?

eficode

STRIDE itself

- A Threat Model put out by Microsoft on April 1st, 1999, that is still in use today.
- Breaks down threats to into 6 distinct categories.

Other Interesting Models/Acronyms

- Process for Attack Simulation and Threat Analysis (PASTA)
- Visual, Agile, and Simple Threat (VAST)
- Damage, Reproducibility, Exploitability, Affected users & Discoverability (DREAD)

eficode

Spoofing

An attack against the property: **Authentication**

- Impersonating a user to log in.
- Generating false websites (googel.com, apple-comps.com, etc) to create the impression of authority.
- Masquerading as a legitimate process or function.

eficode

Tampering

An attack against the property: **Integrity**

Includes modifications made to:

- A system.
- An externally downloaded service/tool
- A codebase or binary storage.

eficode

Repudiation

An attack against the property: **Non-repduation**

In short, the ability to **plausibly deny** having performed an action.

- Editable logging.
- Untracked file changes.
- “No dear, I never visit *that* sort of website”

The STRIDE model

Information Disclosure

An attack against the property: **Confidentiality**

- Access to source code.
- Publishing of customer/client information.
- Read rights to other confidential information

eficode

Denial of Service

An attack against the property: **Availability**

- Deny access to a service through request spamming.
- Degrade the quality of a service through resource hogging.
- Even cutting cables could be considered DoS.

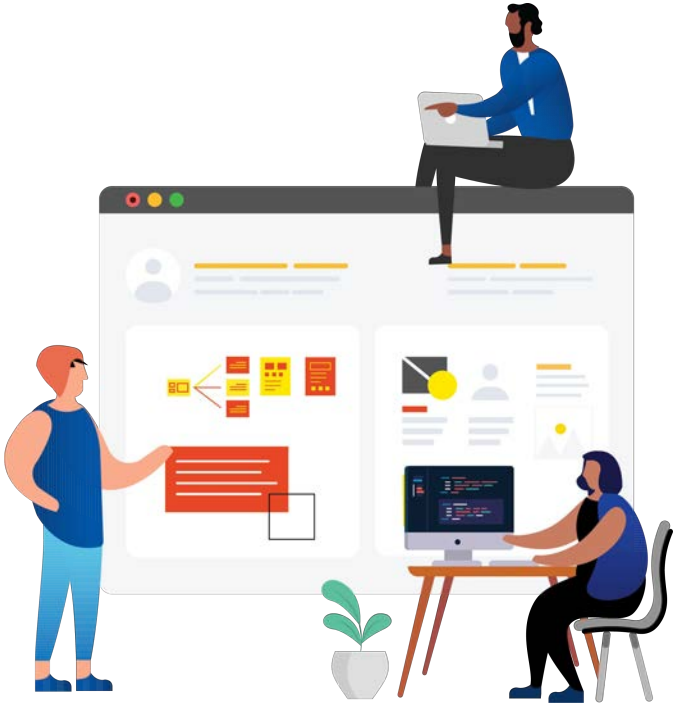
eficode

Elevation of Privilege

An attack against the property: **Authorization**

- Allowing the execution of remote code while unauthorized.
- Elevating from limited user to admin-level.
- Forcing open or door or picking a lock.

eficode



Let’s start with STRIDE, how it was originally intended.

Let’s head back to 1999.

eficode

Data Flow Diagram Circa 1999

Darren Richardson

darren.richardson@eficode.com

+358 40 753 0283

linkedin.com/in/greatbushybeard

eficode

A Caveat About the Speaker

Born in 1987, meaning:

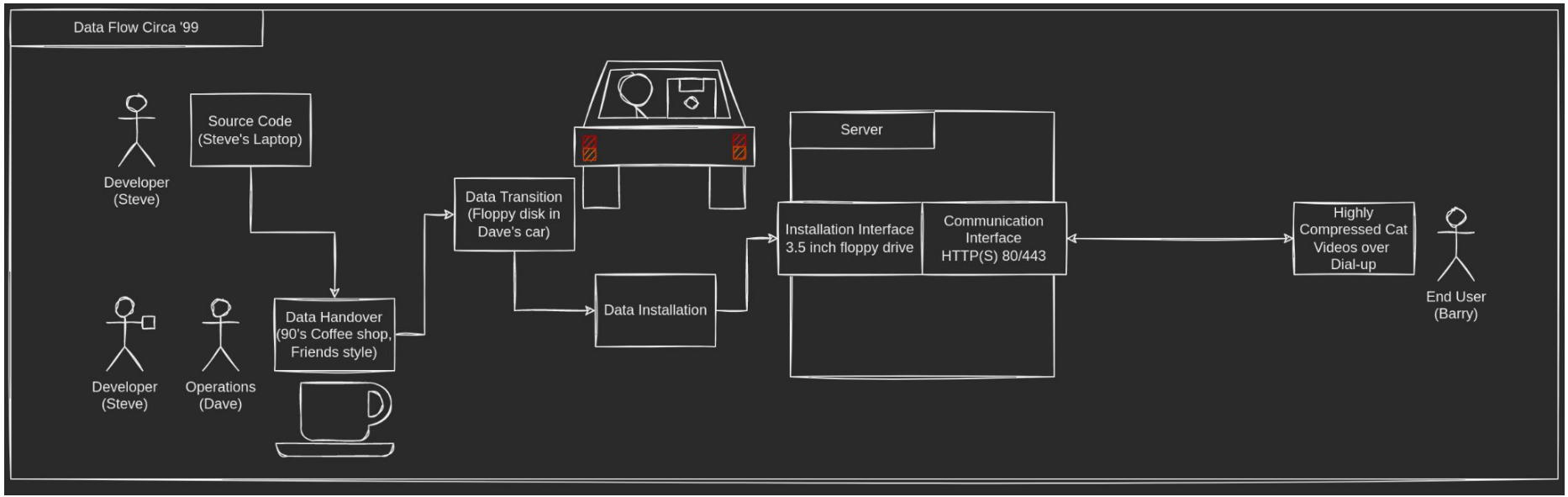
- ➔ I was 12 years old in 1999
- ➔ I was, therefore, not active in professional IT at the time.
- ➔ I was mostly skipping homework.
- ➔ And playing video games.

As such, this section will be based on wild speculation and assumptions about the IT business in the late 90s!

My Cultural Touchpoints

eficode

Data Flow Diagram



eficode

Trust Boundaries Circa 1999

eficode

What's a Trust Boundary?

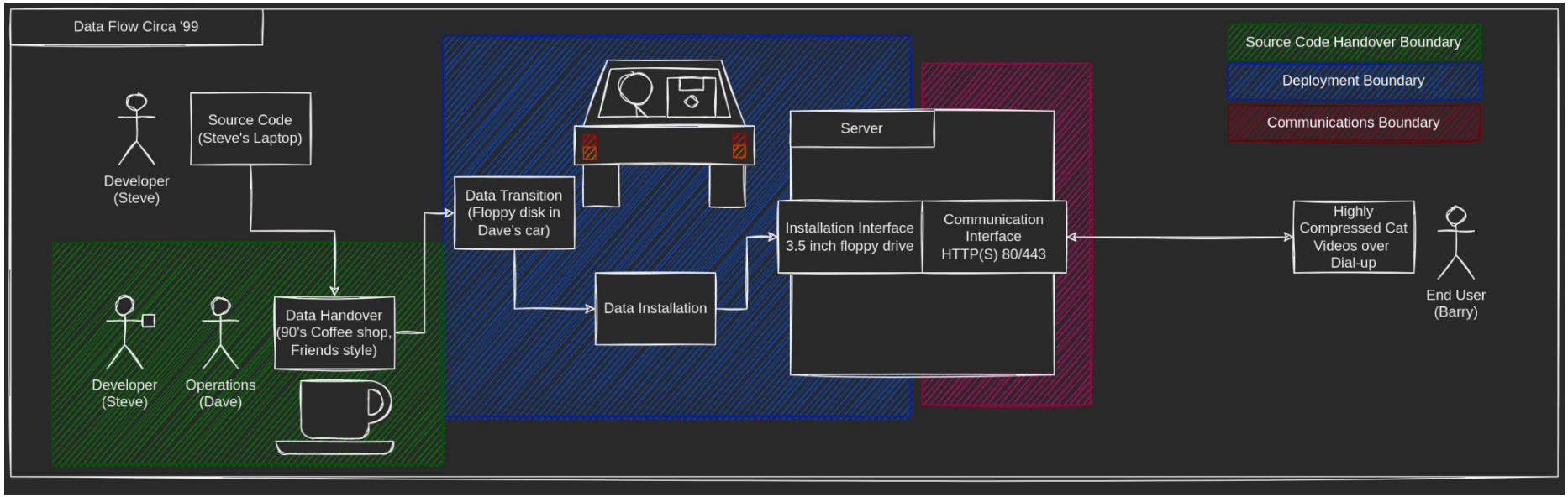
The Wiki Definition

Trust boundary is a term used in computer science and security which describes a boundary where program data or execution changes its level of "trust," or where two principals with different capabilities exchange data or commands.

What it means in practise:

1. In most practical places, it means data transit.
2. But if modelling against a single-platform system, can also be process-level control.

Threat Boundaries



eficode

STRIDE Analysis Circa 1999

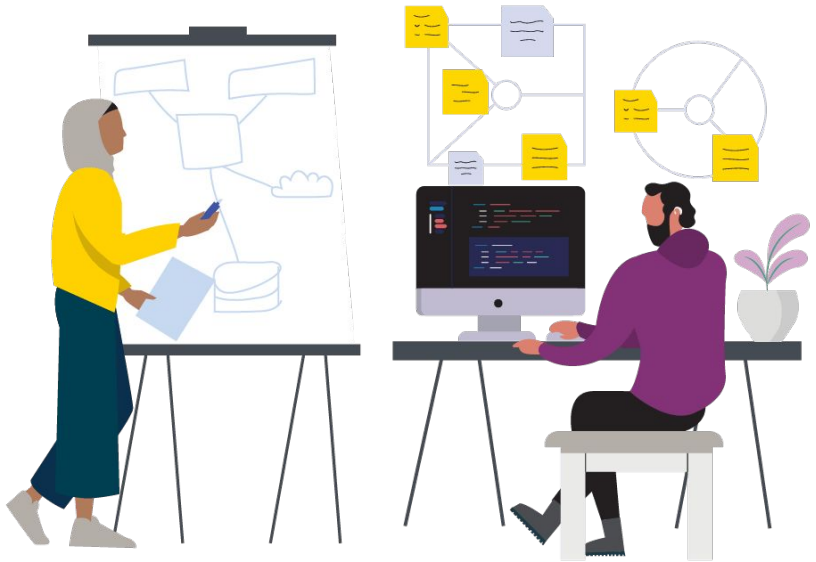
eficode

Applying STRIDE (STRIDEing?)

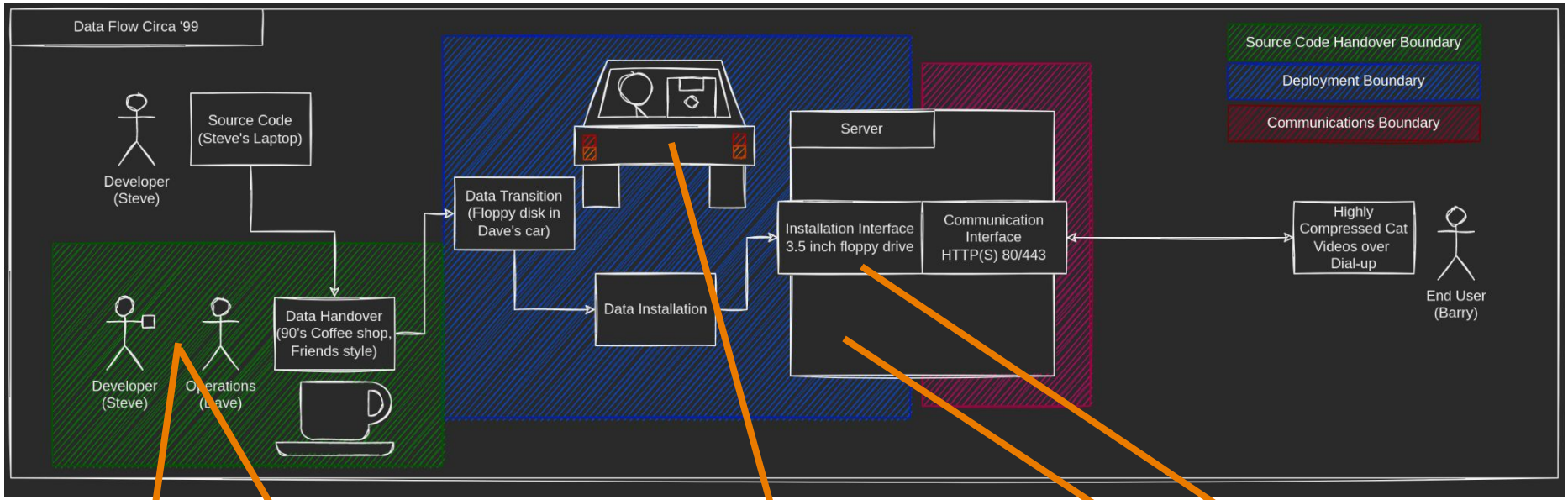
Here's where we apply the threat model.

For each threat boundary we should:

1. Look at inputs / outputs
2. Consider and note potential threats based on the STRIDE categories
3. Plan mitigations for the devised threats.



Threat Boundaries



Spoofing

Can identity be established between Developer and Operations?

Information Disclosure

Eavesdropping? Mitigation: Move to secure, private location.

Tampering

Does OPS go to install software directly? Can disks be switched while in the car?

Repudiation

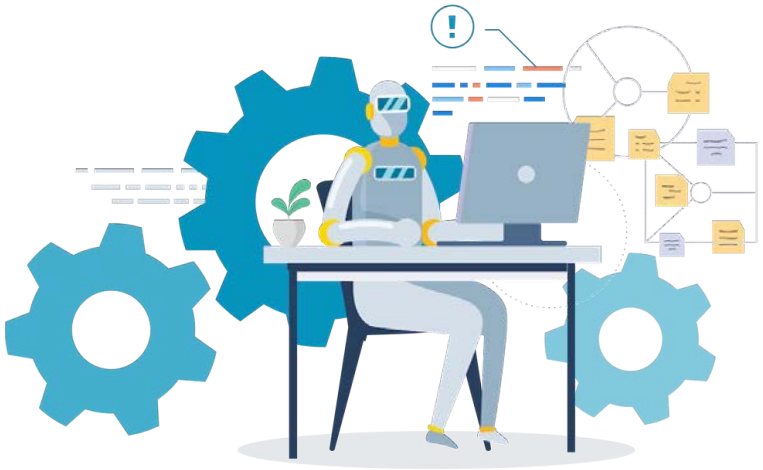
Are logs centrally stored or remain on the server?

Denial of Service

Failure to insert “Disc 2 of 8” corrupts installation. Service down.

eficode

Data Flow Diagram Circa 2022



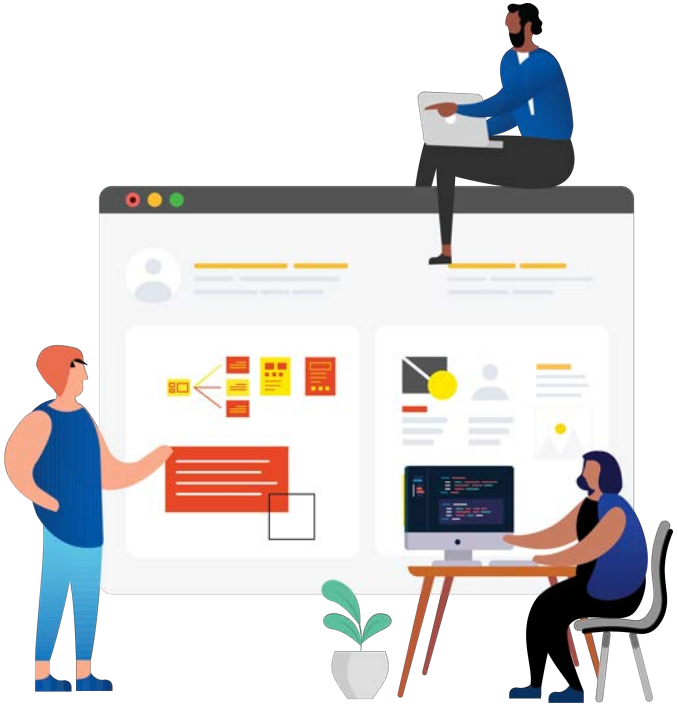
23 years later

Now we're here.

It is impossible to sum up how security has changed over the last two decades.

How does this change the application of the STRIDE model?

eficode

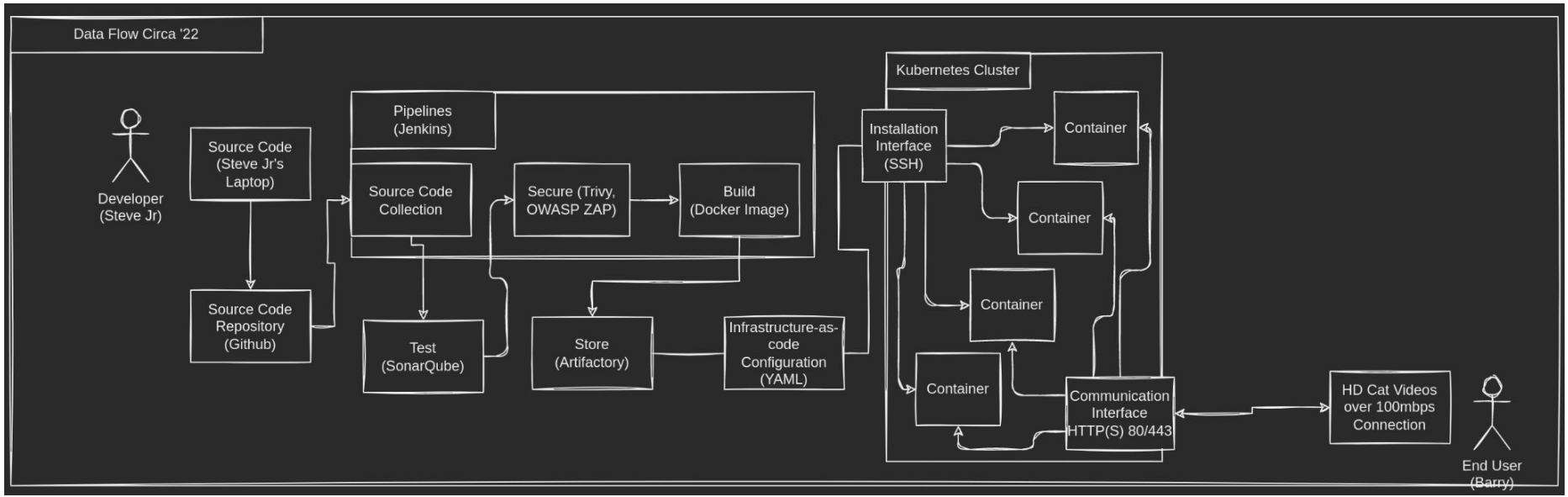


It Doesn’t.

It just complicates matters slightly.

eficode

Data Flow Diagram



eficode

Trust Boundaries Circa 2022

eficode

What’s a reminder of Trust Boundaries?

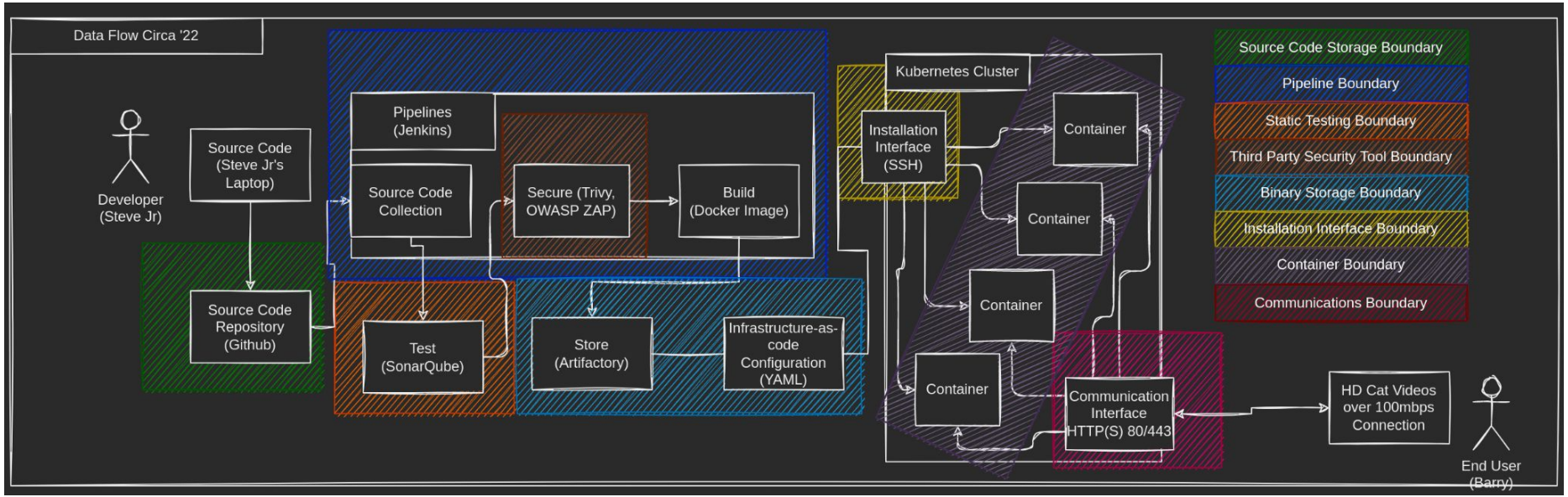
The Wiki Definition

Trust boundary is a term used in computer science and security which describes a boundary where program data or execution changes its level of "trust," or where two principals with different capabilities exchange data or commands.

What it means in practise:

1. In most practical places, it means data transit.
2. But if modelling against a single-platform system, can also be process-level control.

Trust Boundaries

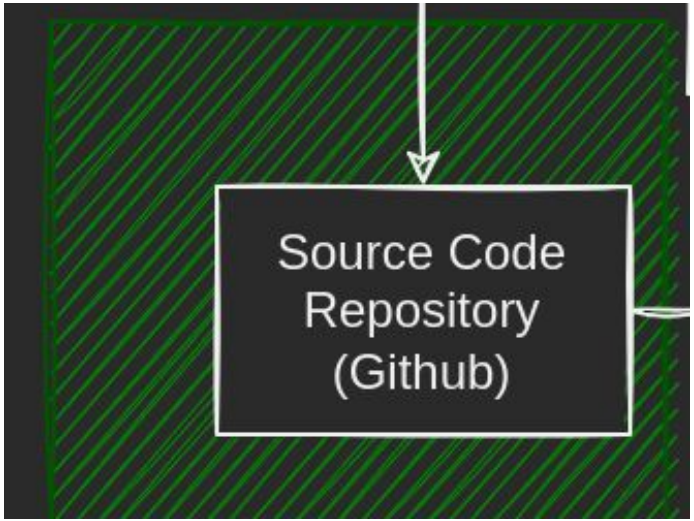


eficode

STRIDE Analysis Circa 2022

eficode

Source Code Repository



Contains source code data. Potential IPR. Thousands of man-hours of work, most likely.

Spoofing	Authentication spoofing, allowing erroneous logins.
Tampering	Modification of source to insert malicious code.
Repudiation	Access logs can be deleted, making code changes untrackable.
Information Disclosure	IPR leakage in the form of lost code, leading to reputation damage.
Denial of Service	Service loss prevents builds and/or GIT commits.
Elevation of Privilege	Unauthorized admin access leads to code-base modification (Tampering) or data leakage (Information Disclosure).

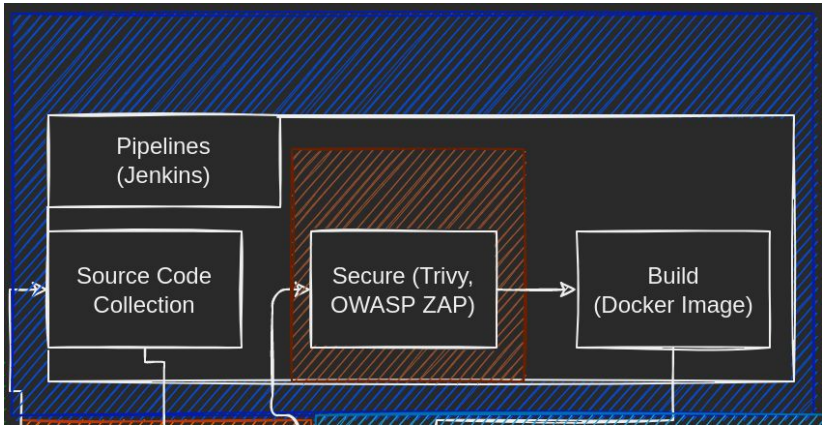
eficode

Pipelines

Pipeline platform and the pipelines themselves

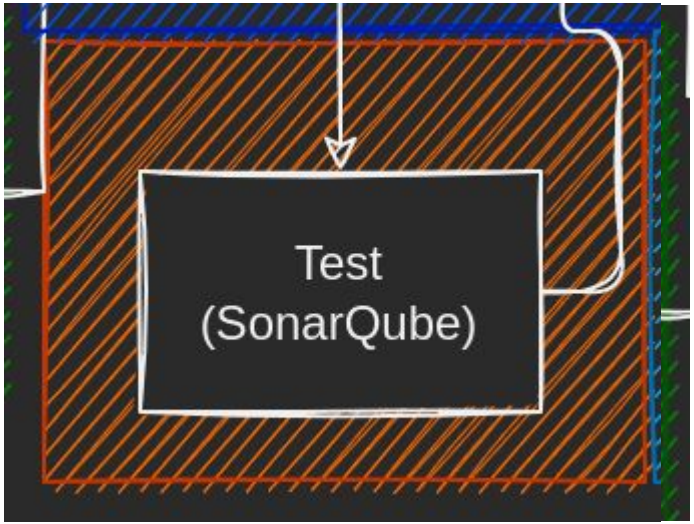
Contains pipeline configuration. Has access rights (READ) to GIT and (WRITE) to binary storage.

Spoofing	Spoofing the code-base storage location could lead to injected code in the final product.
Tampering	Modification of the pipelines could build pretty much anything instead of our code.
Repudiation	Always stays the same. Log everything centrally.
Information Disclosure	Same risks as Source Code Repo boundary. IRP loss.
Denial of Service	In acute situations, could prevent the deployment of emergency patches.
Elevation of Privilege	Alter the build pipelines, disable/skip tests, escaping pipeline, etc.



eficode

Static Code Analysis Tool



Receives the code to perform analysis. Returns only pass or fail values.

Spoofing	Access to the platform?
Tampering	Tests skipped or altered.
Repudiation	Log everything.
Information Disclosure	Leakage of test data, may lead to loss of reputation.
Denial of Service	Testing not possible. Pipeline interrupted or testing bypassed.
Elevation of Privilege	More access to the platform?

eficode

In Pipeline Security Tools

Such as Trivy for Docker Image scanning or OWASP ZAP for web app vulnerabilities

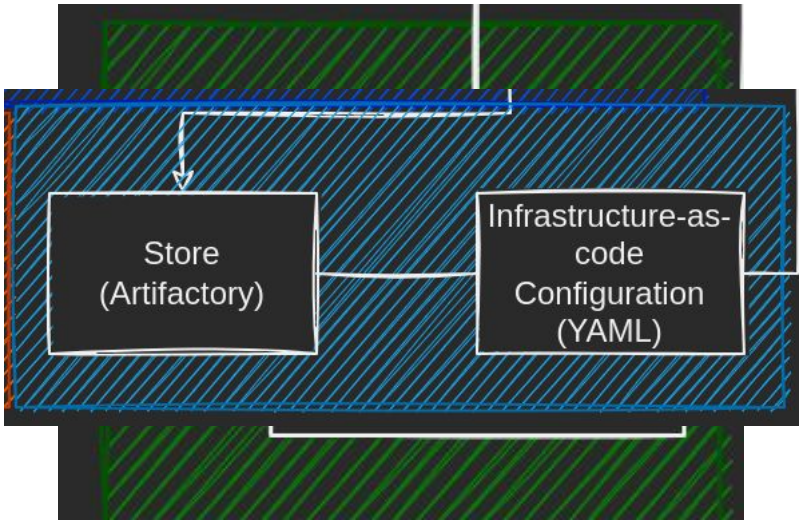
Reports security data for the images. Passes or fails pipelines based on this information.

Spoofing	External tools. Risk of spoofing those domains and downloading the wrong tools.
Tampering	Security checks disabled or corrupted due to alteration of the open source software.
Repudiation	False security check information.
Information Disclosure	Security check information, such as vulnerabilities, could be disclosed.
Denial of Service	Prevent security checks through lack of availability.
Elevation of Privilege	Same as Pipeline Boundary.

eficode

Storage

Stores compiled binaries or built Docker Images. May also have infrastructure as code configuration for system deployments.



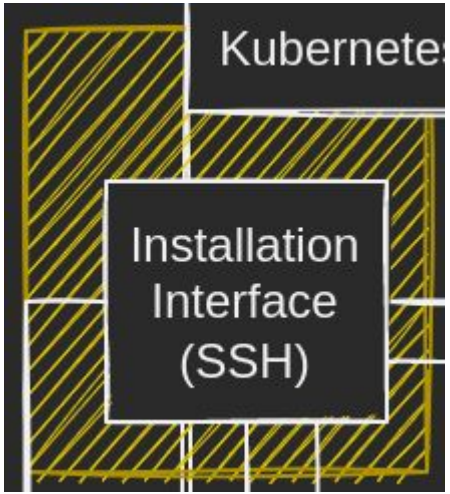
Spoofing	Spoofed credentials allow for upload / download of images which could be distributed to systems.
Tampering	Changes / replaced images or binaries which could be distributed.
Repudiation	Log everything.
Information Disclosure	Leaked images reverse engineered leading to loss of IPR.
Denial of Service	Inaccessible service prevents updates in acute situations.
Elevation of Privilege	Re-upload of altered images via illicit admin credentials.

eficode

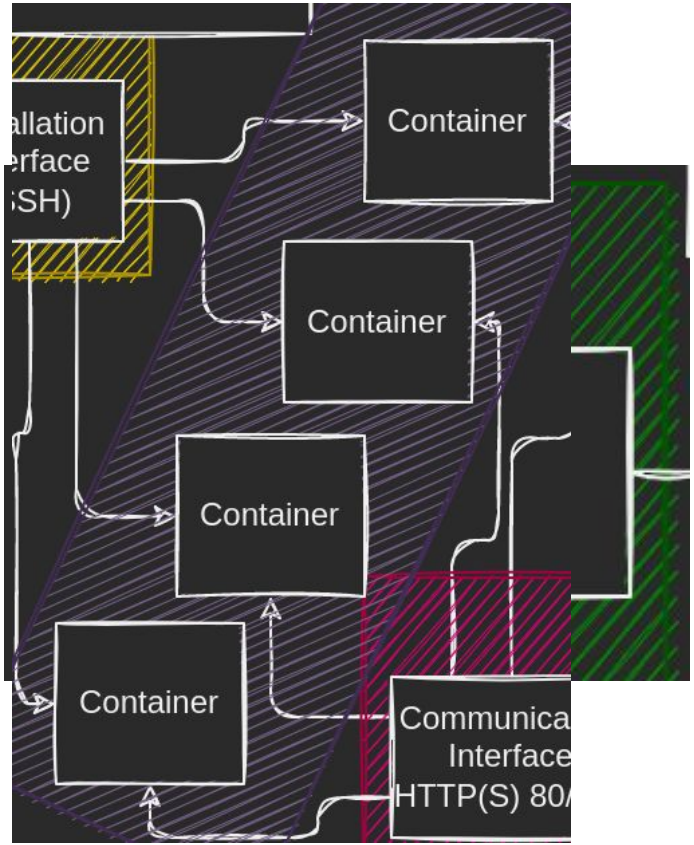
Installation Interface

SSH access to the system as the sole source of installation access.

Spoofing	Spoofed SSH credential, granting access.
Tampering	Editing of SSH config, allowing for root logins or password authentication
Repudiation	No logging -> No auth log -> No clear access trail
Information Disclosure	SSH version, server info leakage.
Denial of Service	Kill the service, prevent installation or admin/maint access.
Elevation of Privilege	Login as low level user legitimately and elevate privilege locally.



eficode



Containers

Actual workloads, running in containers inside the Kubernetes cluster.

Spoofing	Wrong image used to create container, perhaps coin mining ops via spoofed DNS, etc.
Tampering	Different container downloaded and spawned. Container edited.
Repudiation	Log everything.
Information Disclosure	Leaked images reverse engineered leading to loss of IPR.
Denial of Service	Loss of service in the containers -> Loss of function of the platform.
Elevation of Privilege	Containers running as root user when not required. Container escape, etc.

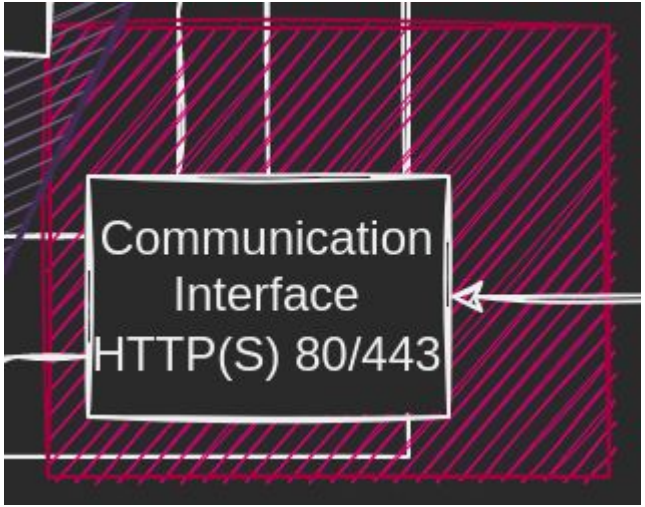
eficode

Communication Interface

Primary means of accepting traffic from the general public

Accessible to the whole internet.

Spoofing	Spoofed traffic, logins, service requests. With public access, the sky is the limit.
Tampering	Could be tampered to serve malware, illicit materials or (god forbid) dog videos.
Repudiation	Remember: Log everything.
Information Disclosure	Robots.txt, info on the web server, other information that should be private.
Denial of Service	Access point, whole service can be taken down here.
Elevation of Privilege	Is there a login? If so: Login elevation.



eficode

STRIDE Analysis

The Fly in the Ointment

STRIDE: The Problem

STRIDE is extremely effective in certain situations.

But may be missing M&M.

Mishandling & Malice

eficode

Mishandling

- ➔ There's no clear place for misconfiguration, mishandling and general mismanagement.
- ➔ Can potentially lead to all STRIDE sections. Should be included everywhere?
- ➔ Estimated 82% of attacks are caused by the "human element".
- ➔ But STRIDE faces specifically outwards.

Malice (Internal)

- ➔ What about an internal employee with an axe to grind?
- ➔ Should this be relegated to Elevation of Privilege?
- ➔ Information Disclosure too? Surely all boundaries would require this same threat.
- ➔ Breach of Principle of Least Privilege should be in a section about Misconfiguration.

eficode

What next?

Iterative Process

According to Microsoft's SDL, threat modelling covers the first three steps of the process:

1. Defining security requirements.
2. Creating an application diagram.
3. Identifying threats.
4. Mitigating threats.
5. Validating that threats have been mitigated.

90s Sitcom Style

Do it all over again in:
The Next Season of Threat Modelling

eficode

Thank you!

Got Questions or Comments?

- ➔ I'll be lurking around Conf42's discord for a while.
- ➔ Send me a message on LinkedIn

Darren Richardson

darren.richardson@eficode.com

+358 40 753 0283

linkedin.com/in/greatbushybeard

eficode

Fin

Now I don’t know what to do with those tossed salads and scrambled eggs.