goto;

GOTO
AMSTERDAM 2023

#GOTOams

goto;

GOTO
Guide

LET US HELP YOU

Ask questions through **the app**

★★★★★

also remember to rate session

THANK YOU!

Download on the App Store

GET IT ON Google Play

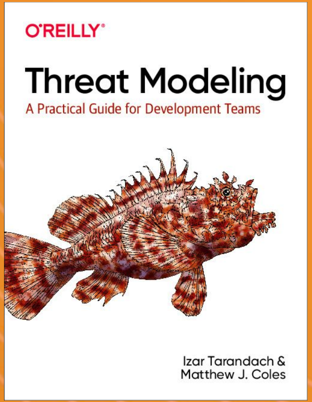
#GOTOams

The One Were We Threat model During Development

Izar Tarandach @izar_t
GOTO; Amsterdam 2023

No TV shows have been harmed in the making of this presentation; the presenter will NOT be using TV-show themed motives to illustrate it. Breathe.

About Me


O'REILLY

Threat Modeling

A Practical Guide for Development Teams

Izar Tarandach &
Matthew J. Coles

Izar Tarandach

- Sr Staff Engineer, Datadog
- Doing the security thing since the 90's
- Poking at everything SSDLC-related
- Lead dev for pytm

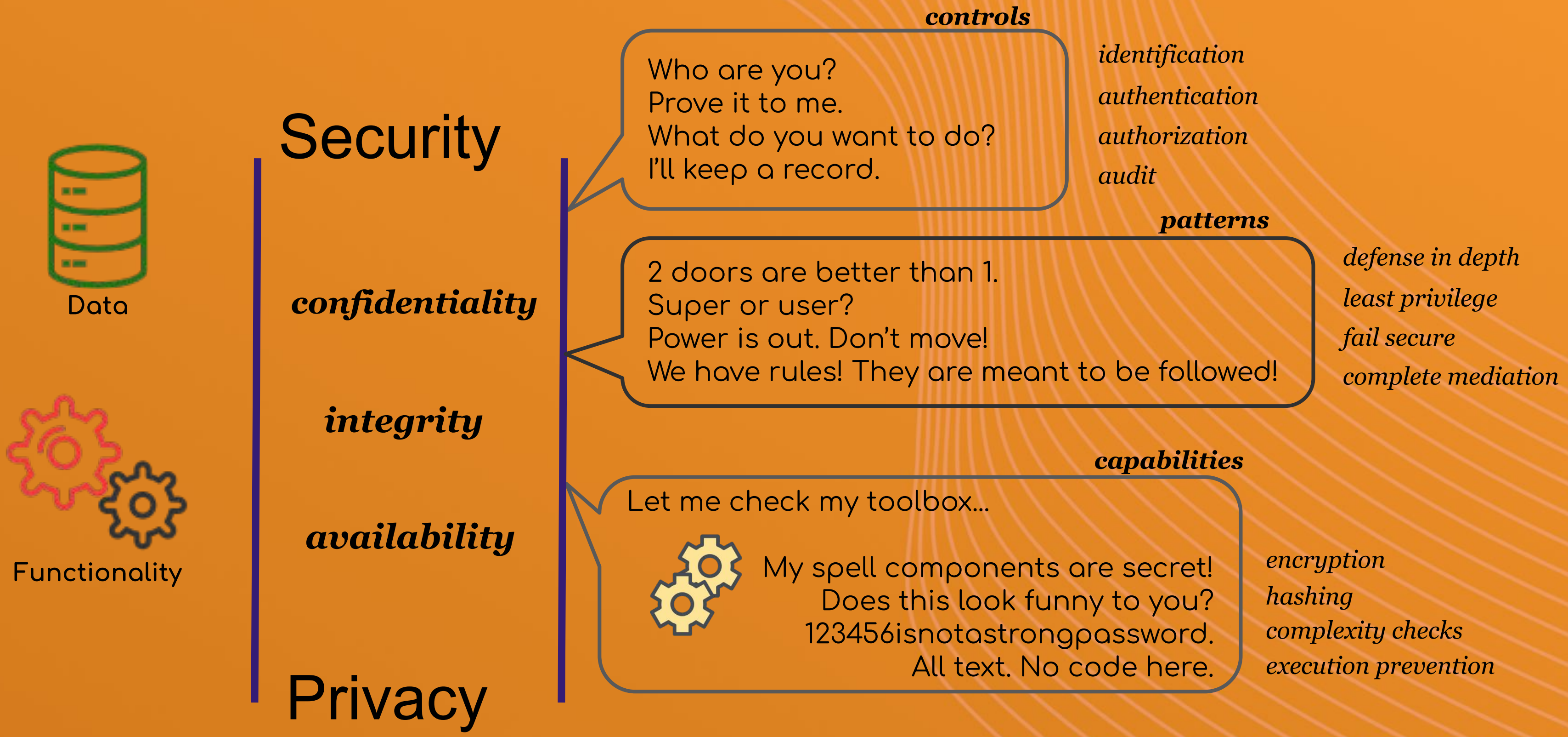
Co-authored "Threat Modeling: A Practical Guide For Development Teams", O'Reilly, 2020

Member of the Threat Modeling Manifesto Working Group, https://threatmodelingmanifesto.org

Standard disclaimer applies:

Agenda

- A quick security and threat modeling primer
- Threat Modeling as a Developer
    - CTM - Continuous Threat Modeling
    - Pytm - the pythonic way of threat modeling
- Questions



Data

Functionality

Security

confidentiality

integrity

availability

Privacy

controls

Who are you?
Prove it to me.
What do you want to do?
I'll keep a record.

identification
authentication
authorization
audit

patterns

2 doors are better than 1.
Super or user?
Power is out. Don't move!
We have rules! They are meant to be followed!

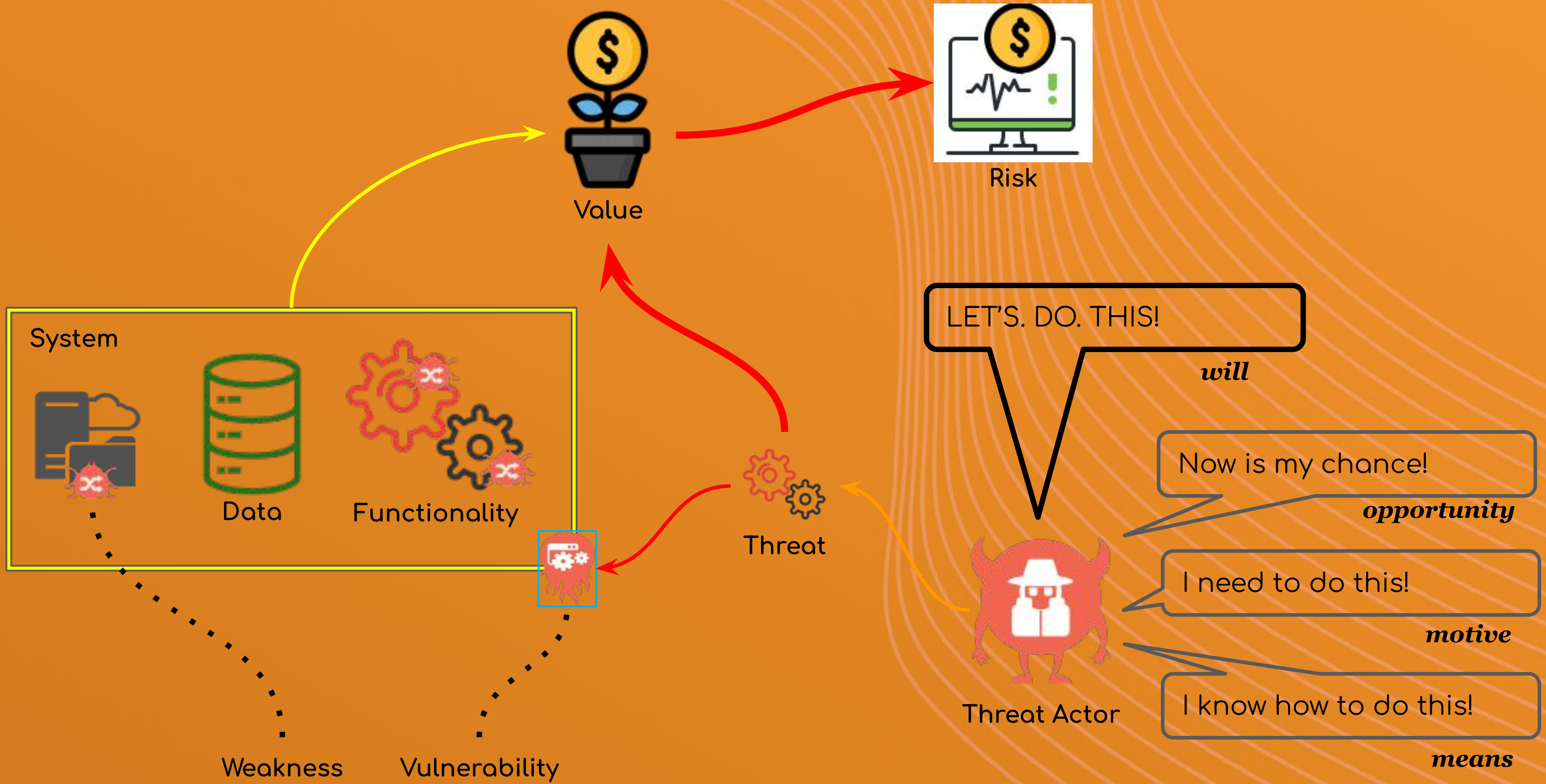
defense in depth
least privilege
fail secure
complete mediation

capabilities

Let me check my toolbox…

My spell components are secret!
Does this look funny to you?
123456isnotastrongpassword.
All text. No code here.

encryption
hashing
complexity checks
execution prevention



Value

Risk

System

Data

Functionality

Threat

Weakness

Vulnerability

LET'S. DO. THIS!

will

Now is my chance!

opportunity

I need to do this!

motive

I know how to do this!

means

Threat Actor

What is the process of threat modeling our systems?

```
Threat Model = f(System Representation (model), Threat Elicitation)
```

What is the process of modeling our systems?

```
Threat Model = f(System Representation (model), Threat Elicitation)
```

=

[Elements

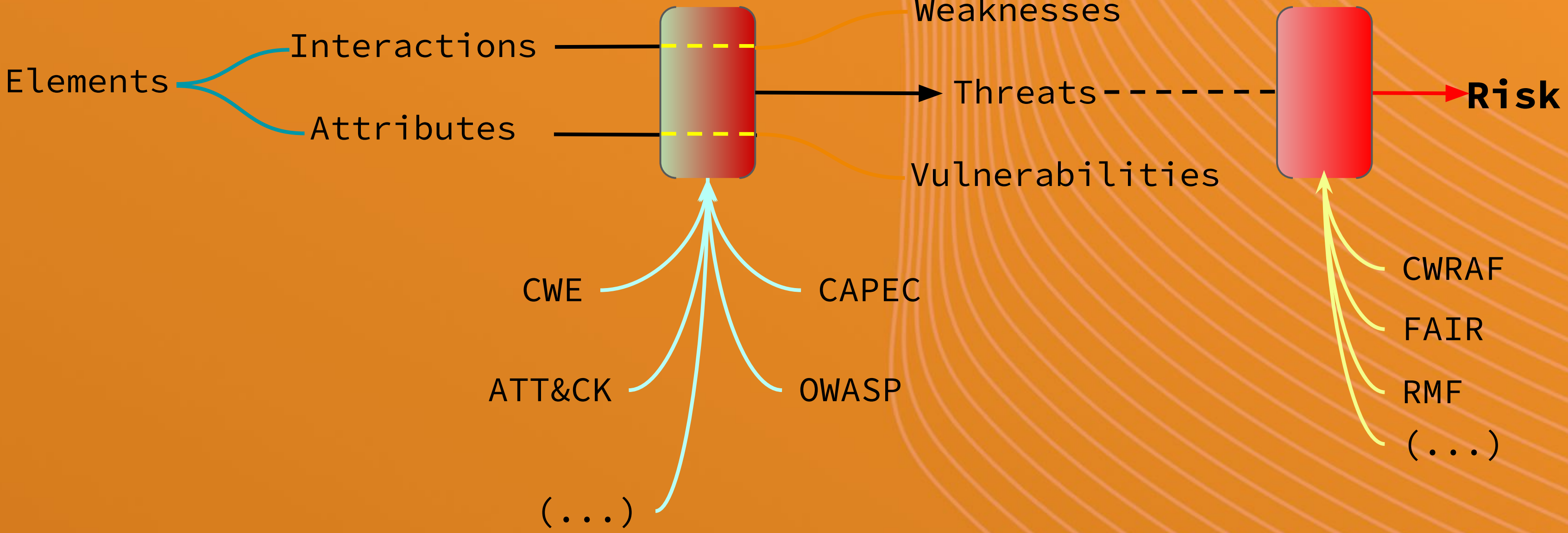
+

Interactions

+

Attributes]

What is the process of threat elicitation?



THREAT MODELING MANIFESTO

Working group consisted of 15 experienced threat modeling practitioners, theorists and academics

Zoe Braiterman
Matthew Coles
Avi Douglen
Marc French
Robert Hurlbut

Jonathan Marcil
Alyssa Miller
Irene Michlin
Chris Romeo
Brook S.E. Schoenfield

Fraser Scott
Adam Shostack
Izar Tarandach
Stephen de Vries
Kim Wuyts

Behind-the-Scenes

https://podcast.securityjourney.com/the-threat-modeling-manifesto-part-1/

https://podcast.securityjourney.com/the-threat-modeling-manifesto-part-2/

THREAT MODELING MANIFESTO

First we needed a consensus of what Threat Modeling *is*:

> *"Threat modeling is*
> ***analyzing** representations of a system*
> *to highlight concerns about*
> *security and privacy characteristics."*

THREAT MODELING MANIFESTO

The most basic Threat Modeling *process* can be summarized to 4 questions:

1. **What are we working on?**
2. **What can go wrong?**
3. **What are we going to do about it?**

…

4. **Did we do a good enough job?**

https://github.com/adamshostack/4QuestionFrame

THREAT MODELING MANIFESTO

The Threat Modeling Manifesto is structured based on the Agile Manifesto

- VALUES
- PRINCIPLES
  - Affirming Patterns
  - Anti-patterns

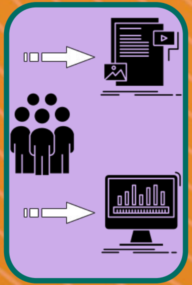
https://www.threatmodelingmanifesto.org/

Values

“THIS”	over	“THAT”
A culture of finding and fixing design issues		checkbox compliance
People and collaboration		methodologies, and tools
A journey of understanding		a security or privacy snapshot
Doing threat modeling		talking about it
Continuous refinement		a single delivery

Principles

- The best use of threat modeling is to *improve* the security and privacy of a system through early and frequent analysis.
- Threat modeling must *align* with an organization’s development practices and follow design changes in iterations that are each scoped to manageable portions of the system.
- The outcomes of threat modeling are *meaningful* when they are *of value* to stakeholders.
- *Dialog* is key to establishing the common understandings that lead to value, while documents record those understandings, and enable measurement.



Patterns

Systemic Approach

Apply knowledge in a structured way.

Informed Creativity

Use the force, or at least craft AND science.

Varied Viewpoints

Cross-functional collaboration is key.

Useful Toolkit

Use tools that improve productivity.

Theory into Practice

Use field-tested techniques modified by local needs.

Anti-Patterns

Hero Threat Modeler

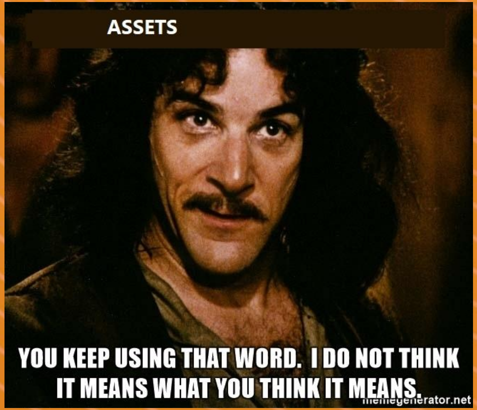
Anyone can threat model.

Admiration for the Problem

Beware analysis-paralysis. Find solutions.

Tendency to Overfocus

There is more to threat modeling than adversaries and assets.

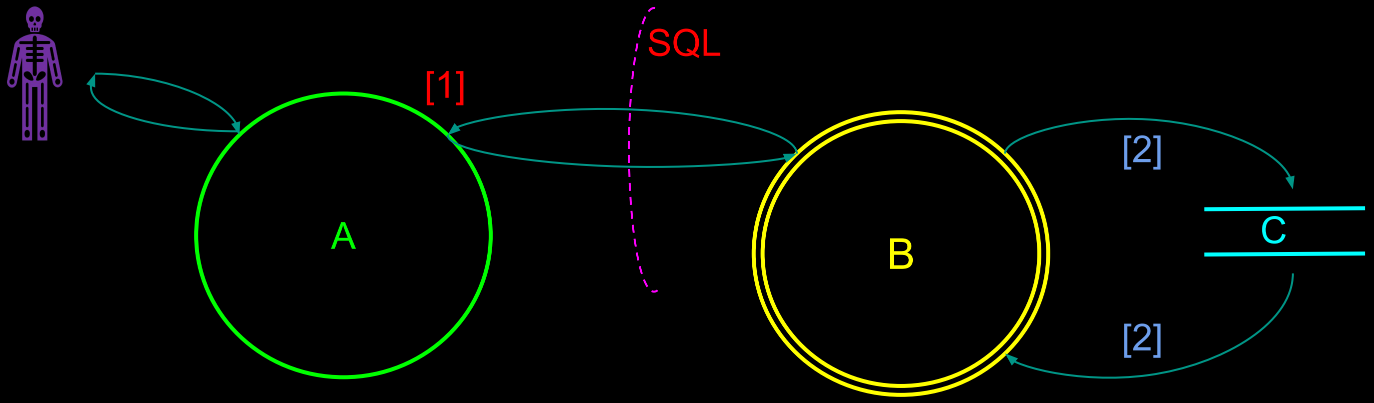

ASSETS
YOU KEEP USING THAT WORD. I DO NOT THINK IT MEANS WHAT YOU THINK IT MEANS.
memegenerator.net

Perfect Representation

There is no single ideal view.

No Perfect Representation - DFD

(Data Flow Diagram)



Process A
* is a web server.
* authenticates users.
* exposes HTTPS only.
* runs on frontend server.

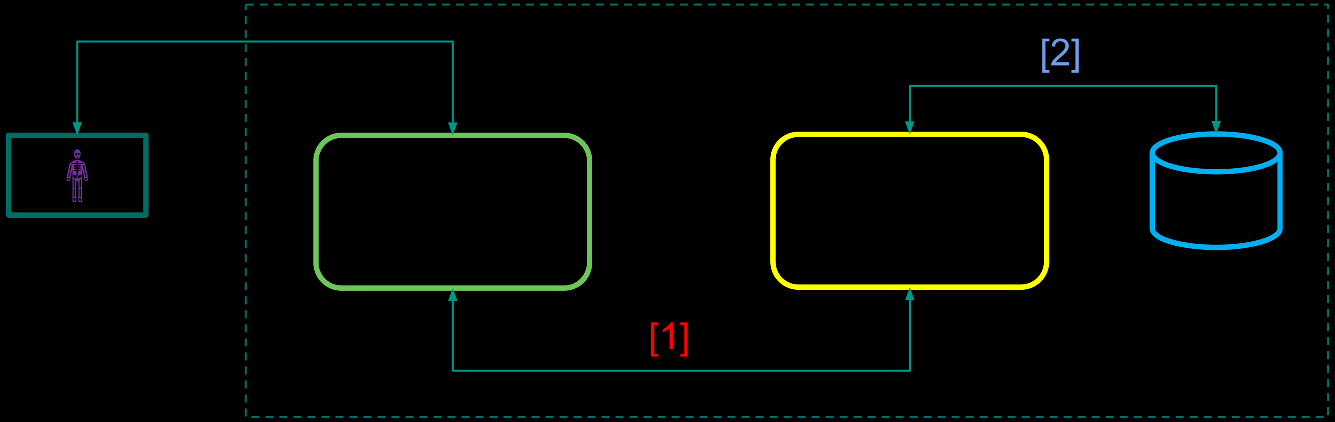
Process B
* is a database server.
* exposes port 1521.
* written in java.
* runs on backend server.
* runs privileged.

Datastore C
* is xml based.
* not encrypted.
* rw-rw--w-

[1]
a. odbc sql queries
b. Sends username and password, gets session token.

[2] Reads/writes data to file

No Perfect Representation – DFD3



Process A
* is a web server.
* authenticates users.
* exposes HTTPS only.
* runs on frontend server.

Process B
* is a database server.
* exposes port 1521.
* written in java.
* runs on backend server.
* runs privileged.

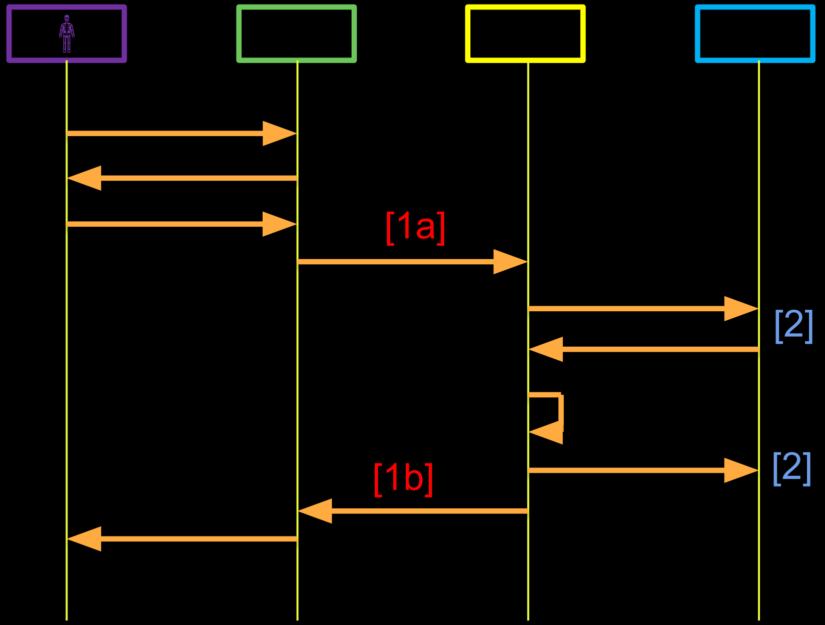
Datastore C
* is xml based.
* not encrypted.
* rw-rw--w-

[1]
a. odbc sql queries
b. Sends username and password, gets session token.

[2] Reads/writes data to file

https://github.com/adamshostack/DFD3

No Perfect Representation – Sequences



Process A
* is a web server.
* authenticates users.
* exposes HTTPS only.
* runs on frontend server.

Process B
* is a database server.
* exposes port 1521.
* written in java.
* runs on backend server.
* runs privileged.

Datastore C
* is xml based.
* not encrypted.
* rw-rw--w-

[1]
a. odbc sql queries
b. Sends username and password, gets session token.

[2] Reads/writes data to file

No Perfect Representation - Methodologies

TARA

Threat
Assessment &
Remediation
Analysis

Focus on Assets vs adversary Tactics, Techniques, and Procedures (TTPs)
Uses catalogs for TTPs and Countermeasures

STRIDE

Spoofing
Tampering
Repudiation
Information Disclosure
Denial of Service
Escalation of Privilege

Security focused

LINDDUN

Linkability
Identifiability
Non-repudiation
Detectability
Disclosure of Information
Unawareness
Non-compliance

Privacy focused

CTM

Continuous
Threat
Modeling

An approach geared towards Agile practitioners
Uses IFTTT-lists for threats and remediations

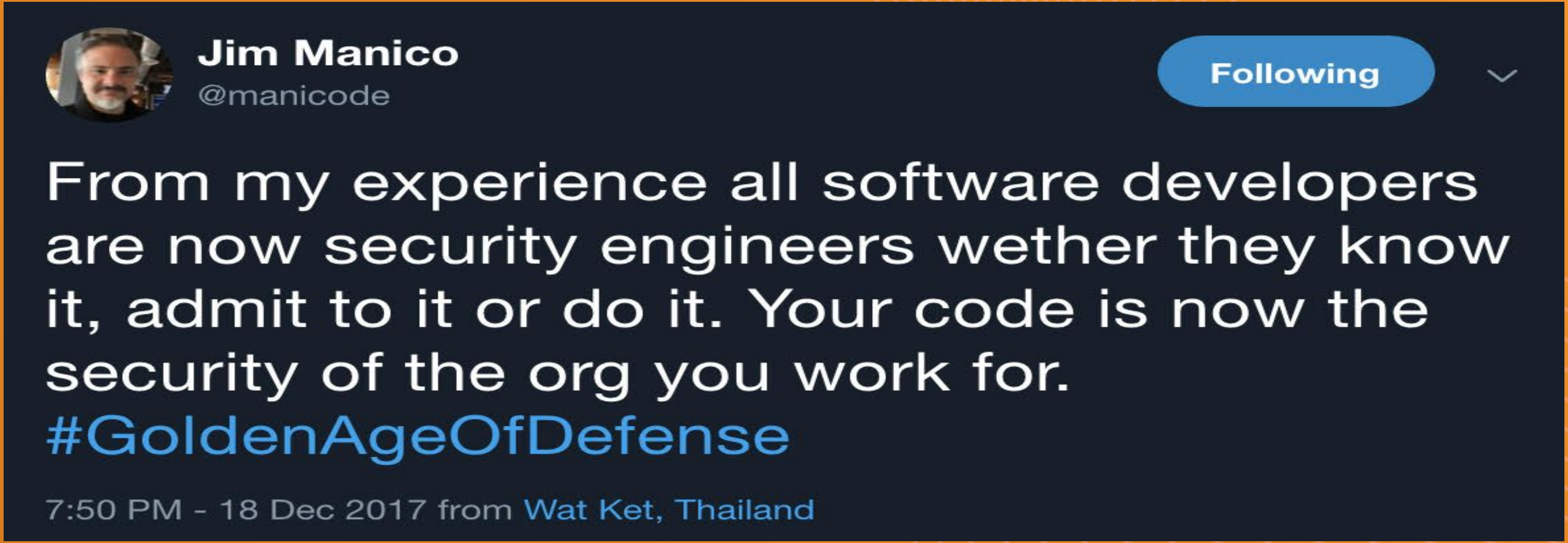
Show and tell - CTM

Continuous Threat Modeling

- Works with DevSecOps!
  - Developers are the new architects
  - Design and implementation happen together, cyclically, at different resolutions
  - Training is not enough - needs focus
  - Shorten the flaw-to-fix killchain
  - Up-to-date threat models are great documentation and test harnesses

https://github.com/Autodesk/continuous-threat-modeling

The Case For Continuous TM

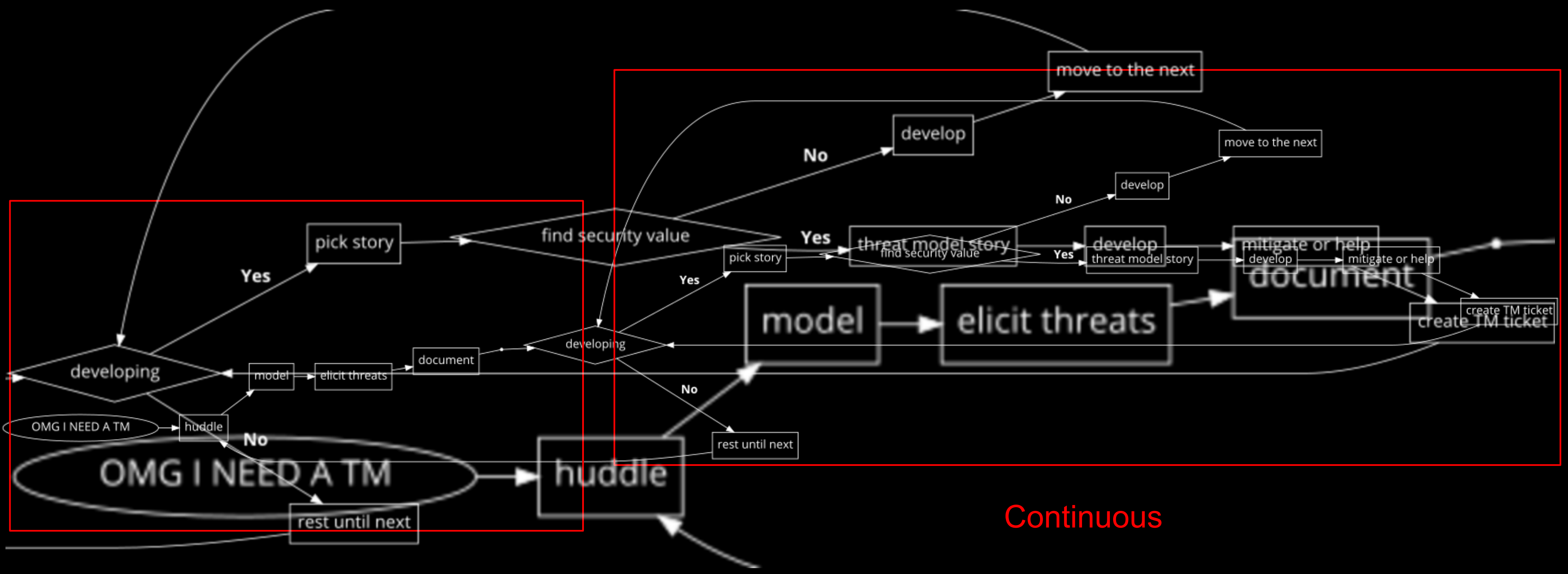


Jim Manico
@manicode

From my experience all software developers are now security engineers wether they know it, admit to it or do it. Your code is now the security of the org you work for. #GoldenAgeOfDefense

7:50 PM - 18 Dec 2017 from Wat Ket, Thailand

Continuous Threat Modeling in a pinch



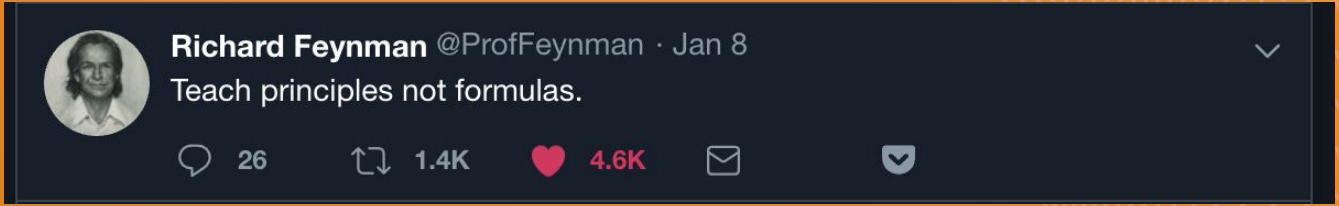
Continuous

Baseline

Threat Model Every Story

- build a baseline - involving everyone. Use whatever technique works for your team. If you don't know how, use CTM's subject based list of points of interest
- designate one or more "threat model curators" who will be responsible for maintaining the canonical threat model document and the findings queue
- instruct your developers to evaluate each one of their stories with focus on security:
    - if the story has no "security value", continue as usual
    - if the story generates a security "notable event", either fix it (and document as a mitigated finding) or pop it up as a "threat model candidate finding" for the curator to take notice of (at Autodesk we are doing this using labels on JIRA tickets)
- make sure your curators are on top of the finding and candidate finding queues

But…how do my developers know what has “security value”?



Richard Feynman @ProfFeynman · Jan 8

Teach principles not formulas.

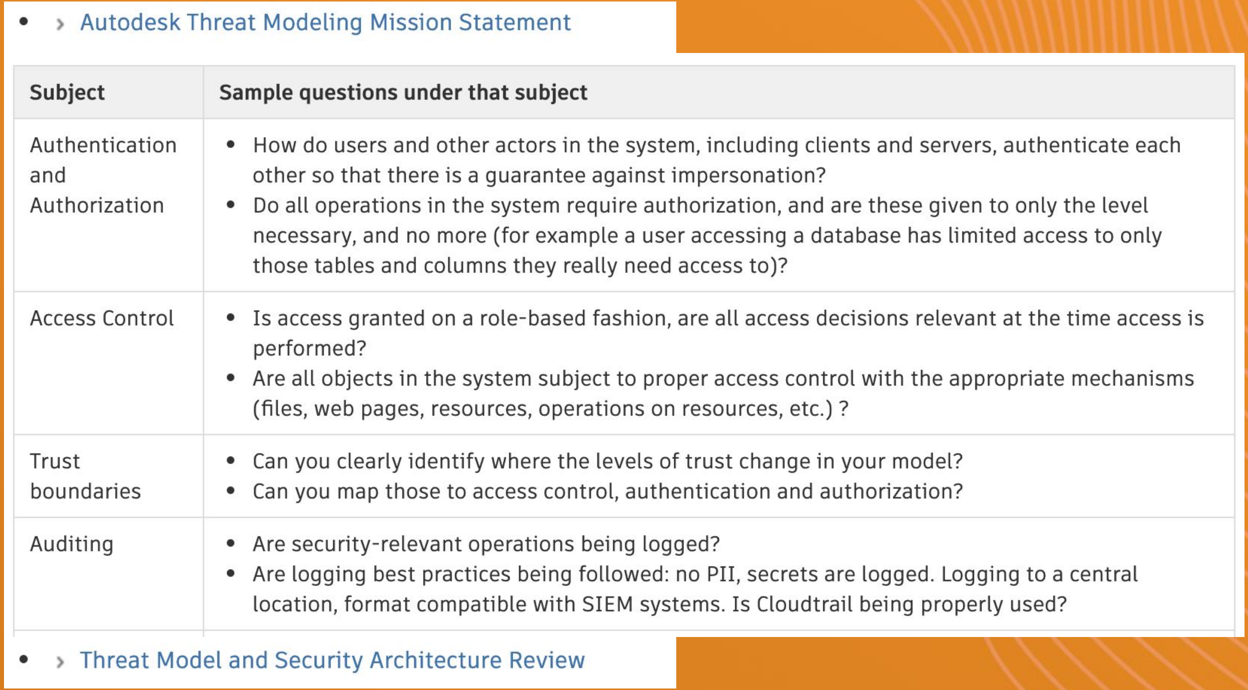
Subject areas

- Question and then continue questioning during “official design time” or when building a baseline

Checklist

- Verify that the principles have been followed at implementation time

Handbook and Subject areas

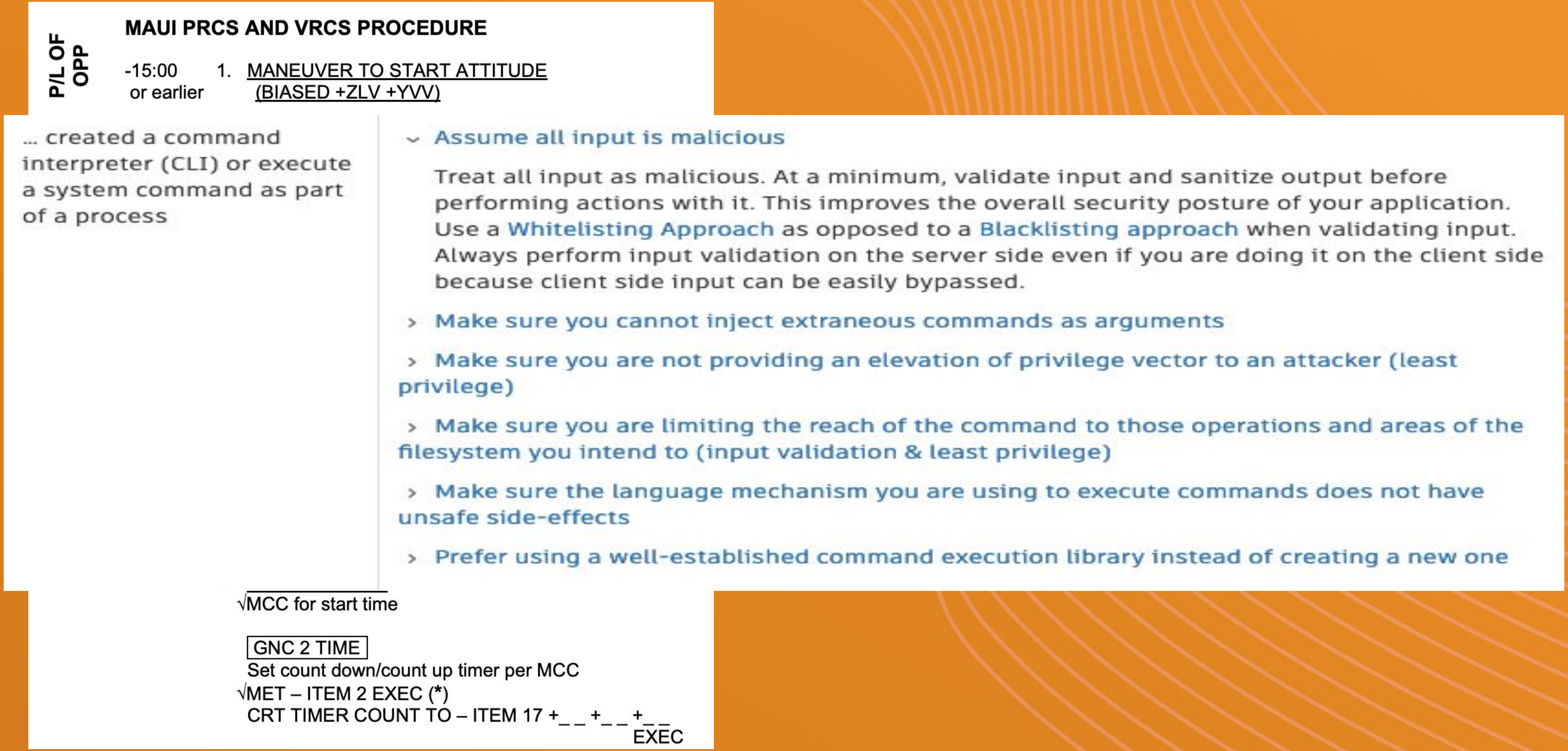


- › Autodesk Threat Modeling Mission Statement

Subject	Sample questions under that subject
Authentication and Authorization	• How do users and other actors in the system, including clients and servers, authenticate each other so that there is a guarantee against impersonation? • Do all operations in the system require authorization, and are these given to only the level necessary, and no more (for example a user accessing a database has limited access to only those tables and columns they really need access to)?
Access Control	• Is access granted on a role-based fashion, are all access decisions relevant at the time access is performed? • Are all objects in the system subject to proper access control with the appropriate mechanisms (files, web pages, resources, operations on resources, etc.) ?
Trust boundaries	• Can you clearly identify where the levels of trust change in your model? • Can you map those to access control, authentication and authorization?
Auditing	• Are security-relevant operations being logged? • Are logging best practices being followed: no PII, secrets are logged. Logging to a central location, format compatible with SIEM systems. Is Cloudtrail being properly used?

- › Threat Model and Security Architecture Review

Principles Checklist



P/L OF OPP

MAUI PRCS AND VRCS PROCEDURE

-15:00 or earlier 1. <u>MANEUVER TO START ATTITUDE (BIASED +ZLV +YVV)</u>

… created a command interpreter (CLI) or execute a system command as part of a process

⌄ Assume all input is malicious

Treat all input as malicious. At a minimum, validate input and sanitize output before performing actions with it. This improves the overall security posture of your application. Use a Whitelisting Approach as opposed to a Blacklisting approach when validating input. Always perform input validation on the server side even if you are doing it on the client side because client side input can be easily bypassed.

› Make sure you cannot inject extraneous commands as arguments

› Make sure you are not providing an elevation of privilege vector to an attacker (least privilege)

› Make sure you are limiting the reach of the command to those operations and areas of the filesystem you intend to (input validation & least privilege)

› Make sure the language mechanism you are using to execute commands does not have unsafe side-effects

› Prefer using a well-established command execution library instead of creating a new one

√MCC for start time

GNC 2 TIME

Set count down/count up timer per MCC

√MET – ITEM 2 EXEC (*)

CRT TIMER COUNT TO – ITEM 17 +_ _ +_ _ +_ _ EXEC

Threat Model Every Story - recap

- build a baseline - involving everyone. Use whatever technique works for your team. CTM provides a “subject based” list of points of interest - they’re starting points, not a checklist!
- designate one or more “threat model curators” who will be responsible for maintaining the canonical threat model document and the findings queue
- instruct your developers to evaluate each one of their stories with focus on security:
  - if the story has no “security value”, continue as usual
  - if the story generates a security “notable event”, either fix it (and document as a mitigated finding) or pop it up as a “threat model candidate finding” for the curator to take notice of (at Autodesk we are doing this using labels on JIRA tickets)
- make sure your curators are on top of the finding and candidate finding queues

Reactions from product teams

- “Uh...what?”
- “This is still too heavy”
- “But how do I know I did everything?”
- “I never saw a room of architects excited about threat modeling before”

Caveat Emptor: This Is Not Perfect

- Difficult to convince teams that the Subject List is not a threat library and developers that the Checklist is not a requirements list – not exhaustive, just a starting point
- The resulting TM won’t be perfect – evolutionary
- A SME or security group may still be necessary for education
- GIGO – garbage-in, garbage-out

Show and tell - pytm

Works with Agile, DevOps, DevSecOps,...

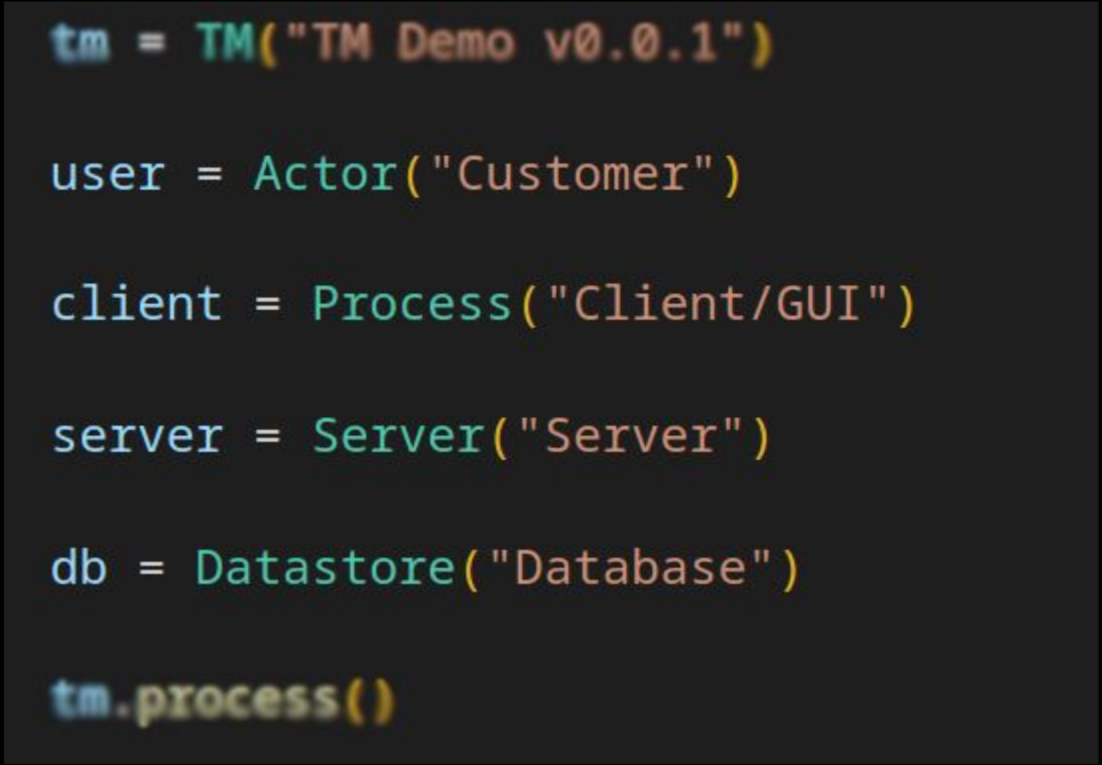
- *“A coder needs a diagram like a fish needs a bicycle”* - Charles S. Harris, paraphrased - helps developers where they live and play
- Supports CTM but doesn’t depend on it
- Express your system as elements in code with attributes
- Get baseline threats
- Get diagrams
- Get a report
- TM and code live and evolve together!

https://github.com/izar/pytm

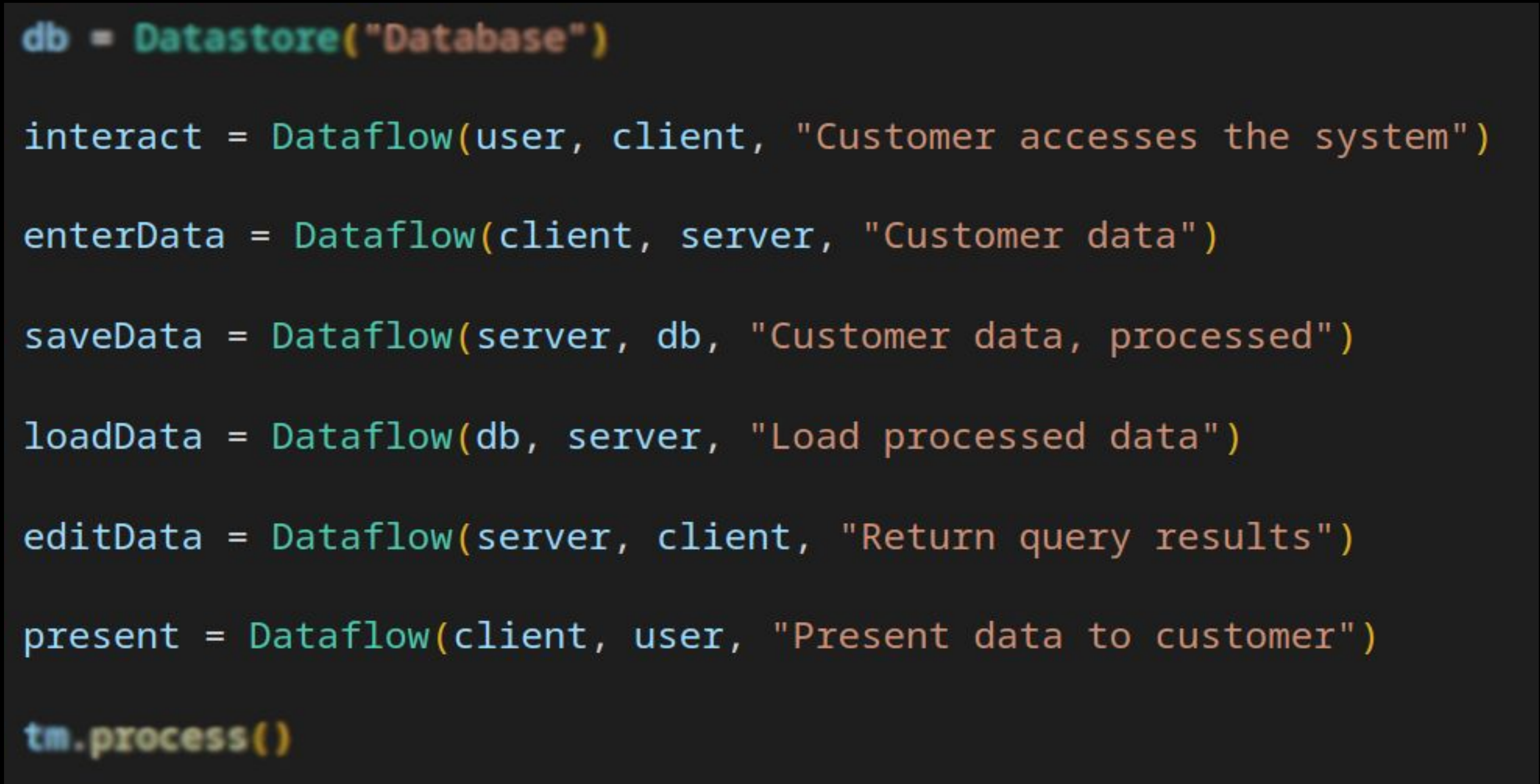
Using pytm

1. Define the components of the model and their relationships (dataflows)
2. Generate a dataflow diagram or a sequence diagram
3. Annotate the components with their attributes
4. Generate a report with the threats identified as a function of component and dataflow attributes

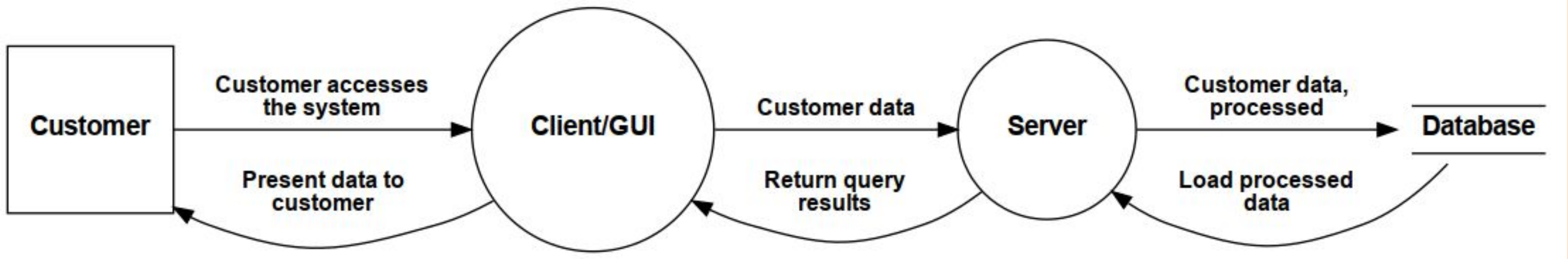
```python
#!/usr/bin/env python3

from pytm import (
    TM, Actor, Boundary, Classification, Data,
    Dataflow, Datastore, Process, Server
)

tm = TM("TM Demo v0.0.1")


...

tm.process()
```

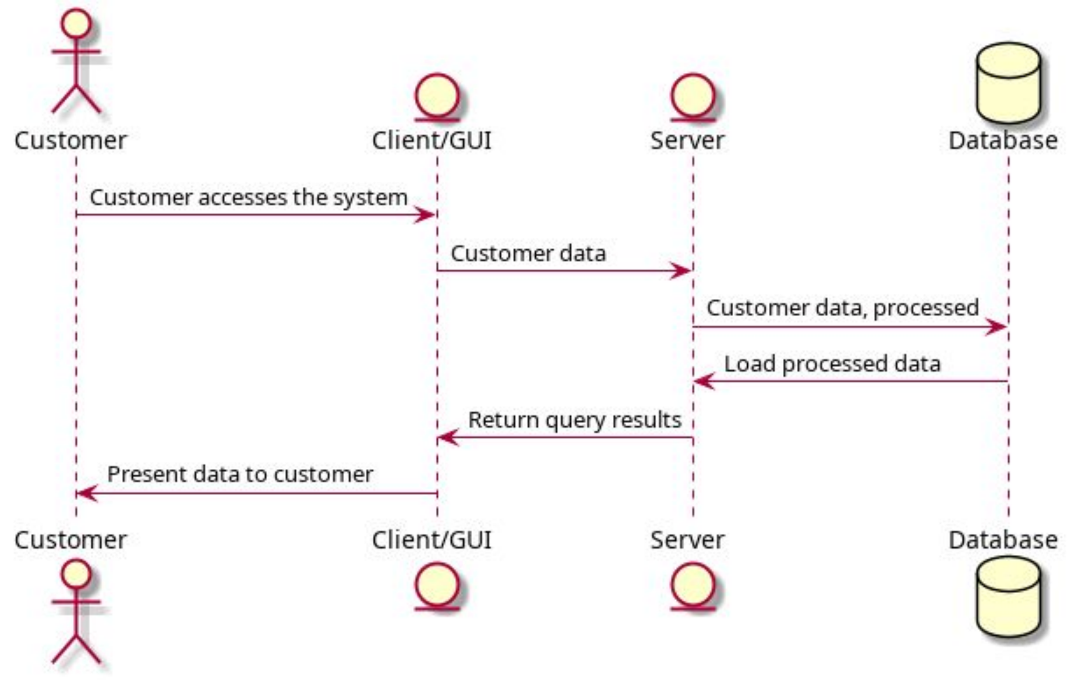


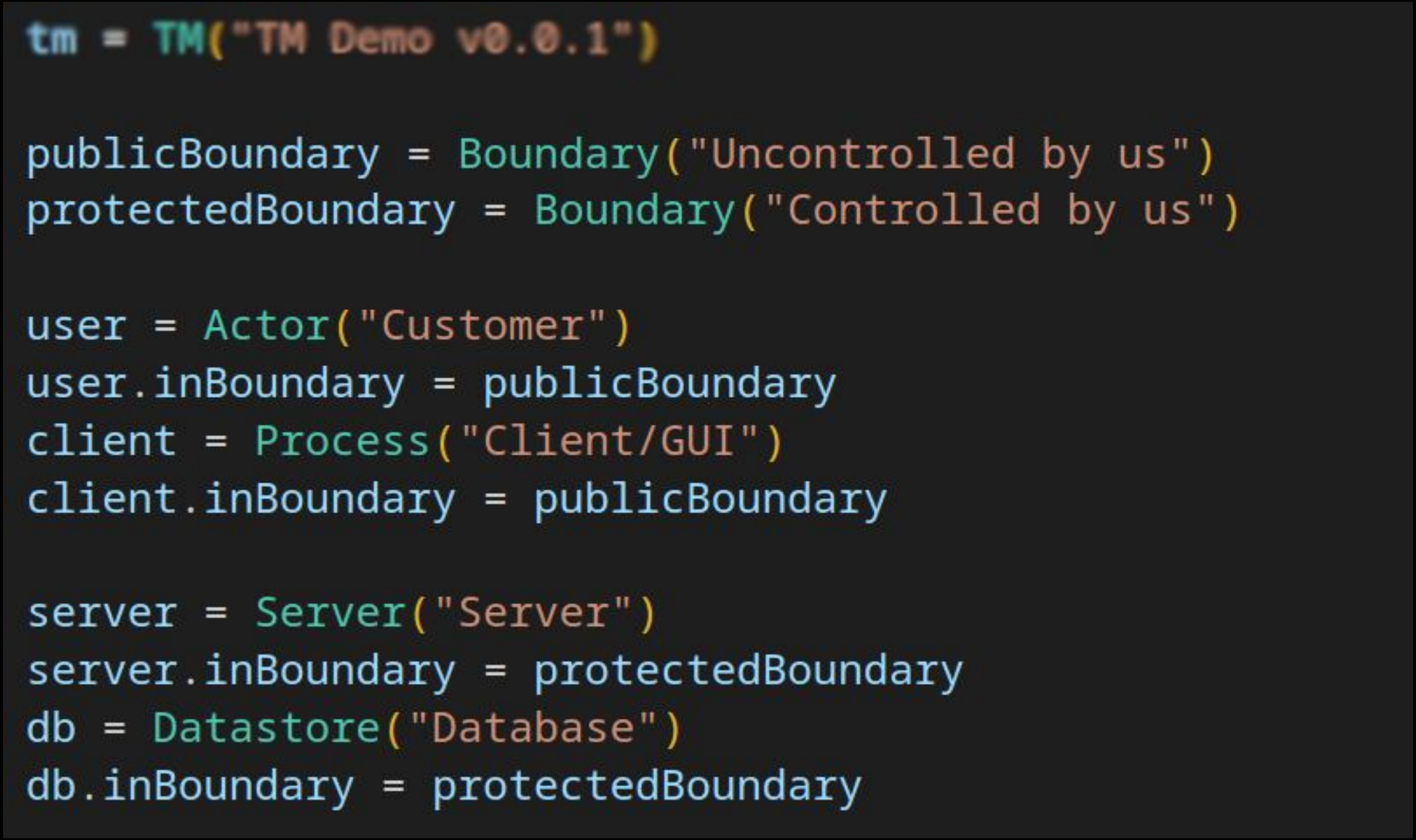
```
tm = TM("TM Demo v0.0.1")

user = Actor("Customer")

client = Process("Client/GUI")

server = Server("Server")

db = Datastore("Database")

tm.process()
```



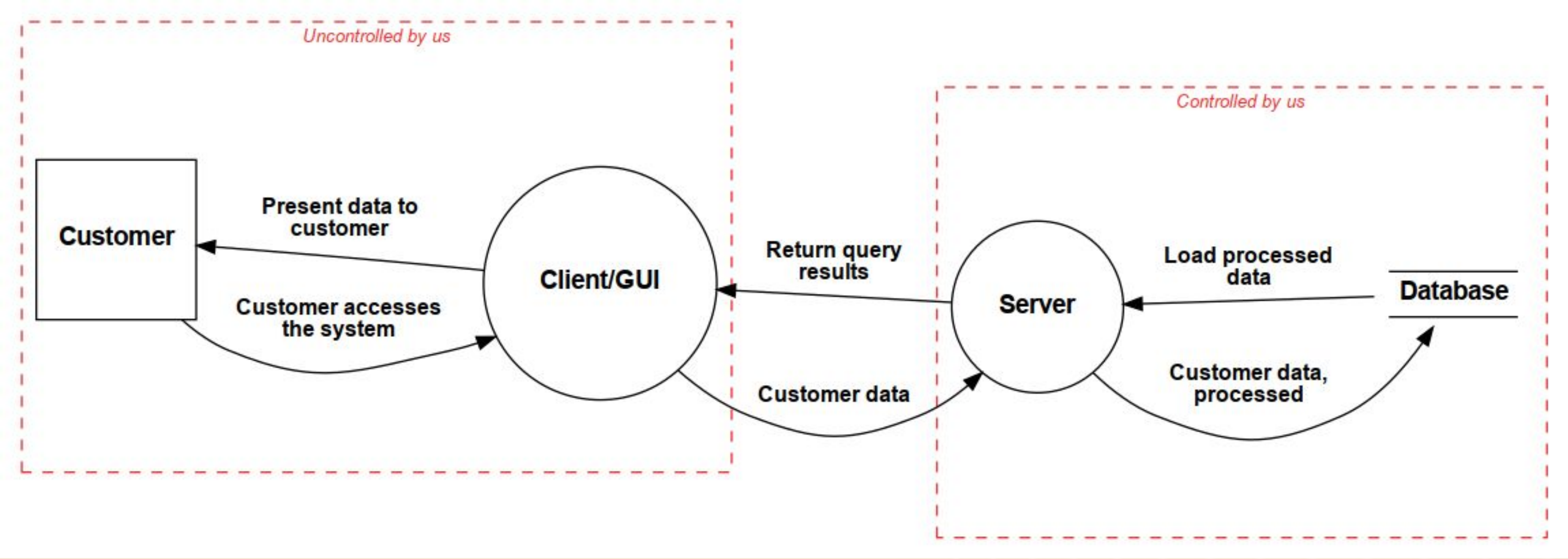
```
db = Datastore("Database")

interact = Dataflow(user, client, "Customer accesses the system")

enterData = Dataflow(client, server, "Customer data")

saveData = Dataflow(server, db, "Customer data, processed")

loadData = Dataflow(db, server, "Load processed data")

editData = Dataflow(server, client, "Return query results")

present = Dataflow(client, user, "Present data to customer")

tm.process()
```

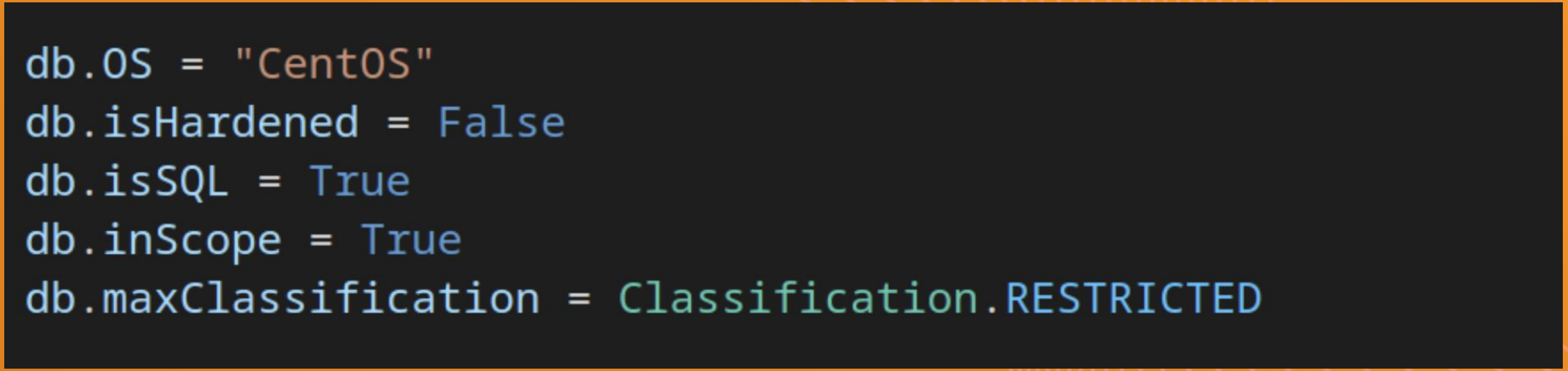

Customer
Customer accesses the system
Client/GUI
Customer data
Server
Customer data, processed
Database
Present data to customer
Return query results
Load processed data


Customer
Client/GUI
Server
Database
Customer accesses the system
Customer data
Customer data, processed
Load processed data
Return query results
Present data to customer
Customer
Client/GUI
Server
Database

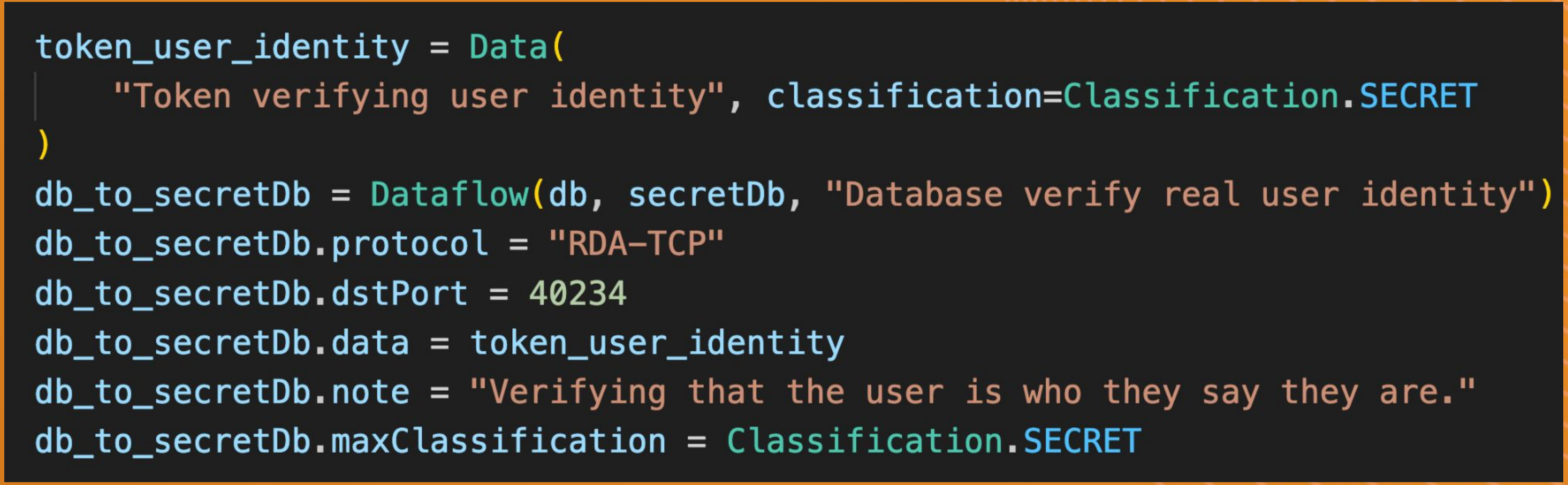


```
tm = TM("TM Demo v0.0.1")

publicBoundary = Boundary("Uncontrolled by us")
protectedBoundary = Boundary("Controlled by us")

user = Actor("Customer")
user.inBoundary = publicBoundary
client = Process("Client/GUI")
client.inBoundary = publicBoundary

server = Server("Server")
server.inBoundary = protectedBoundary
db = Datastore("Database")
db.inBoundary = protectedBoundary
```

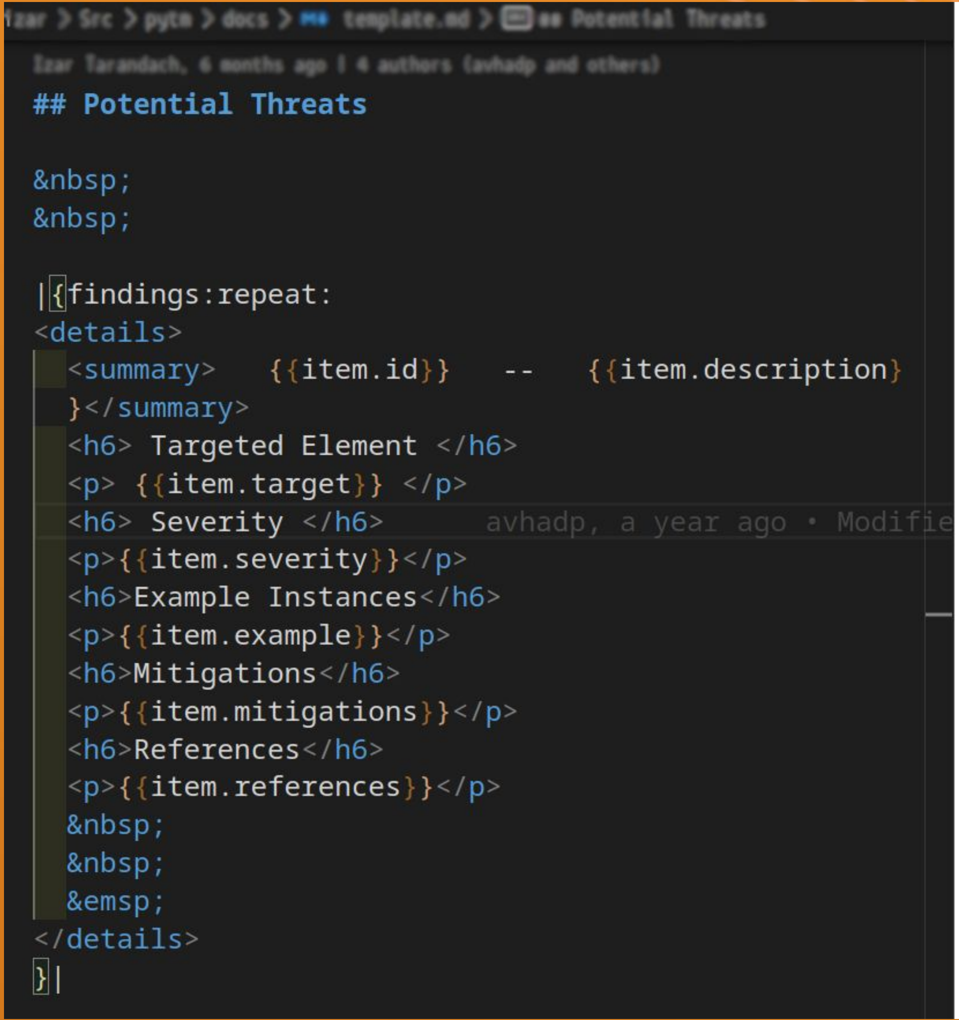

Uncontrolled by us
Controlled by us
Customer
Present data to customer
Customer accesses the system
Client/GUI
Return query results
Customer data
Server
Load processed data
Customer data, processed
Database



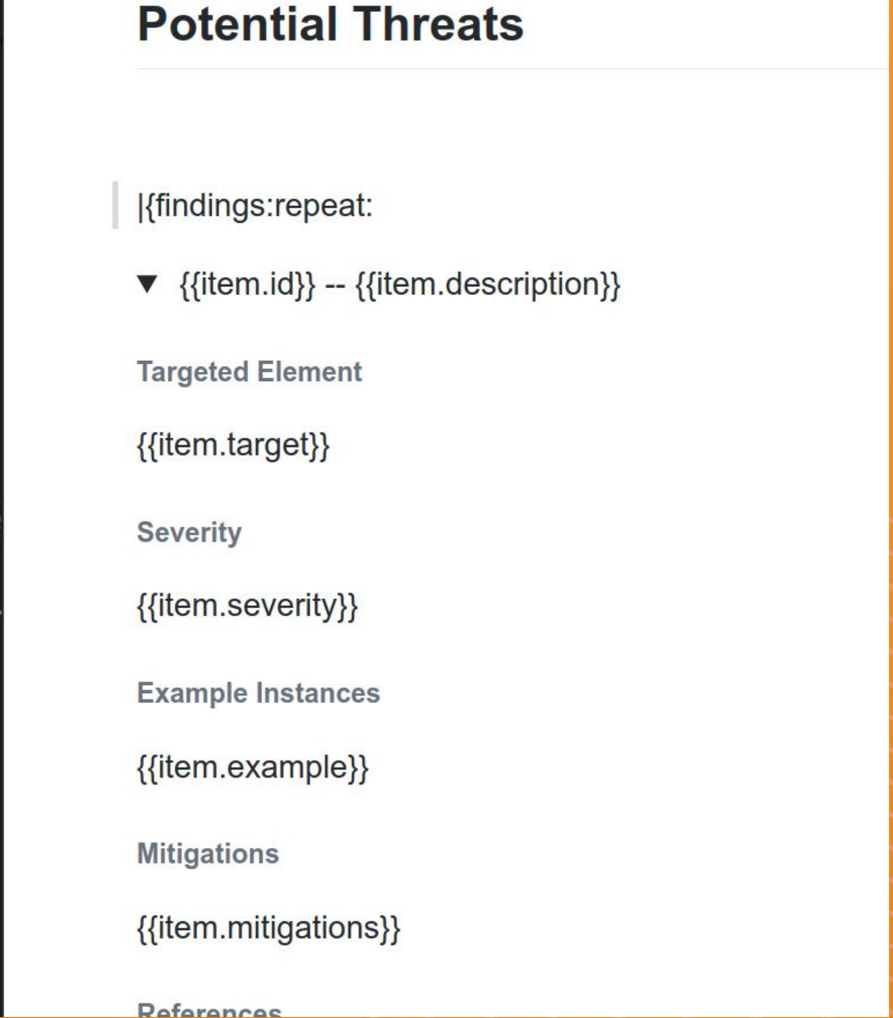
```
db.OS = "CentOS"
db.isHardened = False
db.isSQL = True
db.inScope = True
db.maxClassification = Classification.RESTRICTED
```



```
token_user_identity = Data(
    "Token verifying user identity", classification=Classification.SECRET
)
db_to_secretDb = Dataflow(db, secretDb, "Database verify real user identity")
db_to_secretDb.protocol = "RDA-TCP"
db_to_secretDb.dstPort = 40234
db_to_secretDb.data = token_user_identity
db_to_secretDb.note = "Verifying that the user is who they say they are."
db_to_secretDb.maxClassification = Classification.SECRET
```


izar > Src > pytm > docs > template.md > ## Potential Threats
Izar Tarandach, 6 months ago | 4 authors (avhadp and others)
avhadp, a year ago • Modifie

```
## Potential Threats

&nbsp;
&nbsp;

|{findings:repeat:
<details>
  <summary>   {{item.id}}   --   {{item.description}
  }</summary>
  <h6> Targeted Element </h6>
  <p> {{item.target}} </p>
  <h6> Severity </h6>
  <p>{{item.severity}}</p>
  <h6>Example Instances</h6>
  <p>{{item.example}}</p>
  <h6>Mitigations</h6>
  <p>{{item.mitigations}}</p>
  <h6>References</h6>
  <p>{{item.references}}</p>
  &nbsp;
  &nbsp;
  &emsp;
</details>
}|
```



Potential Threats

|{findings:repeat:

▼ {{item.id}} -- {{item.description}}

Targeted Element

{{item.target}}

Severity

{{item.severity}}

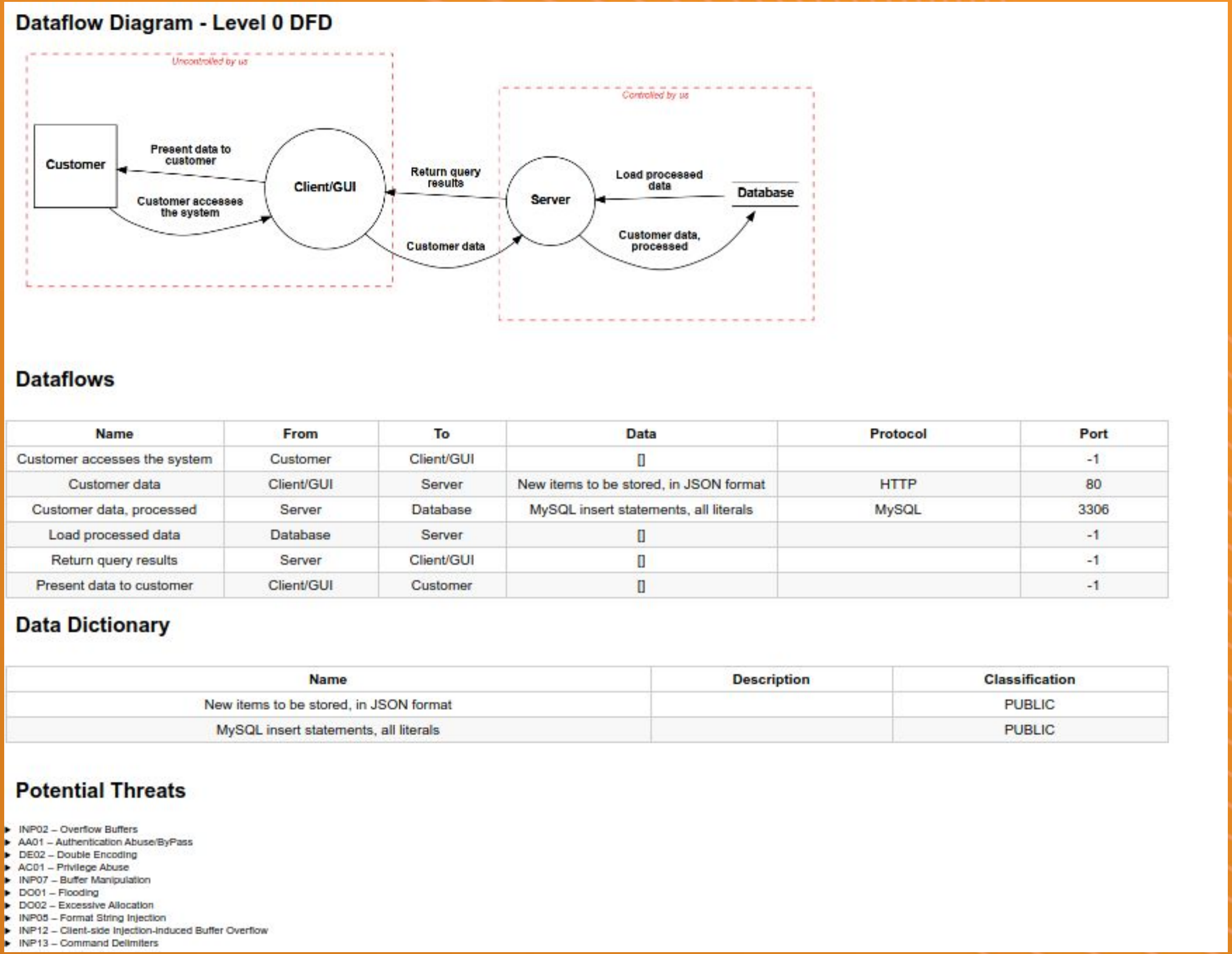
Example Instances

{{item.example}}

Mitigations

{{item.mitigations}}

References



Dataflow Diagram - Level 0 DFD

Dataflows

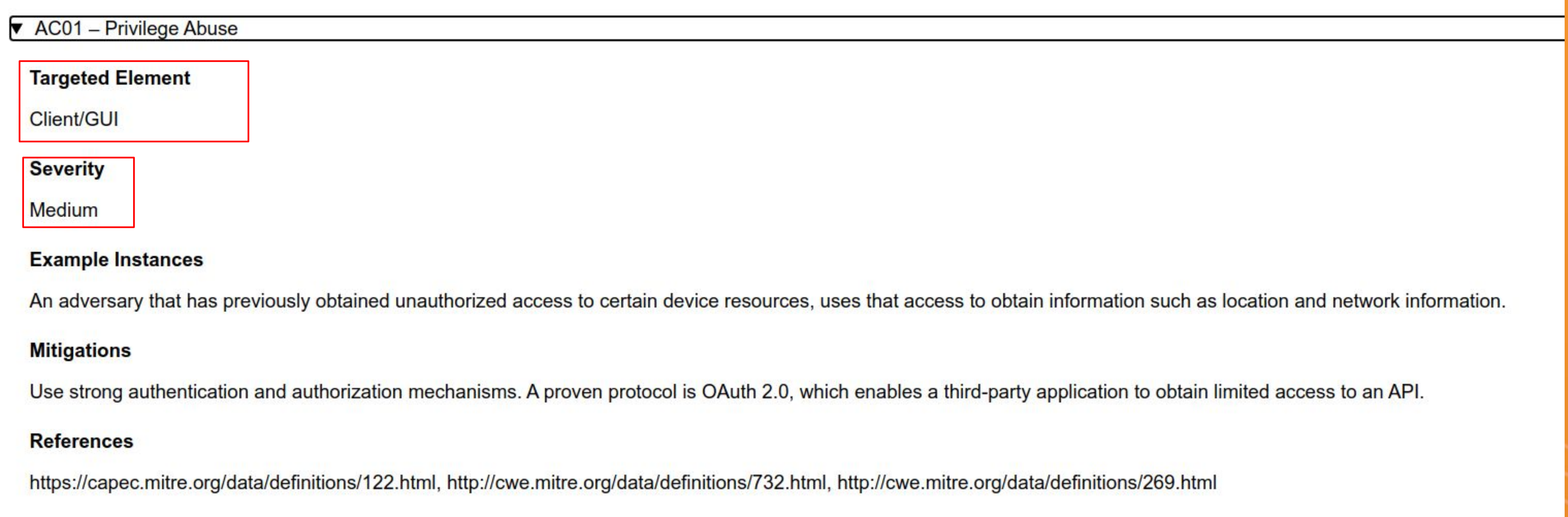
Name	From	To	Data	Protocol	Port
Customer accesses the system	Customer	Client/GUI	[]		-1
Customer data	Client/GUI	Server	New items to be stored, in JSON format	HTTP	80
Customer data, processed	Server	Database	MySQL insert statements, all literals	MySQL	3306
Load processed data	Database	Server	[]		-1
Return query results	Server	Client/GUI	[]		-1
Present data to customer	Client/GUI	Customer	[]		-1

Data Dictionary

Name	Description	Classification
New items to be stored, in JSON format		PUBLIC
MySQL insert statements, all literals		PUBLIC

Potential Threats

- INP02 – Overflow Buffers
- AA01 – Authentication Abuse/ByPass
- DE02 – Double Encoding
- AC01 – Privilege Abuse
- INP07 – Buffer Manipulation
- DO01 – Flooding
- DO02 – Excessive Allocation
- INP05 – Format String Injection
- INP12 – Client-side Injection-induced Buffer Overflow
- INP13 – Command Delimiters



▼ AC01 – Privilege Abuse

Targeted Element

Client/GUI

Severity

Medium

Example Instances

An adversary that has previously obtained unauthorized access to certain device resources, uses that access to obtain information such as location and network information.

Mitigations

Use strong authentication and authorization mechanisms. A proven protocol is OAuth 2.0, which enables a third-party application to obtain limited access to an API.

References

https://capec.mitre.org/data/definitions/122.html, http://cwe.mitre.org/data/definitions/732.html, http://cwe.mitre.org/data/definitions/269.html

Resources

@izar_t

- The Threat Modeling Manifesto
  https://threatmodelingmanifesto.org

- "Threat Modeling: A Practical Guide for Development Teams"
  https://amzn.to/39G7qlX

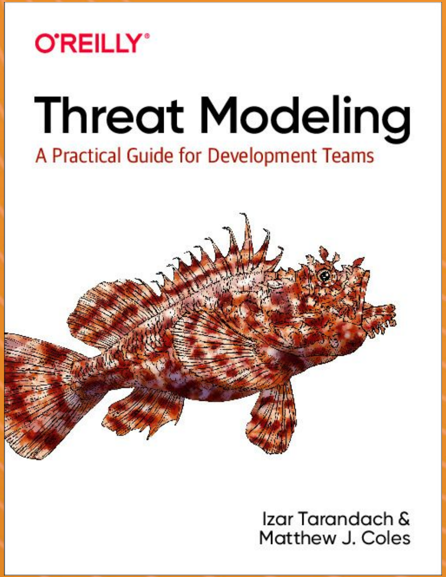
- pytm - https://github.com/izar/pytm

- Continuous Threat Modeling -
  https://github.com/izar/continuous-threat-modeling

- Adam Shostack's "Threat Modeling: Designing for Security",
  https://amzn.to/2NhRy1x

- Brook Schoenfields' "Securing Systems",
  https://amzn.to/3iq7Y3f

- SAFECode's "Tactical Threat Modeling",
  https://bit.ly/3bRB8au



O'REILLY®

Threat Modeling

A Practical Guide for Development Teams

Izar Tarandach &
Matthew J. Coles

Thank you!

Questions?

goto;

GOTO
Guide

★★★★★

Remember to
rate this session

THANK YOU!

Download on the
App Store

GET IT ON
Google Play

#GOTOams