Threat Modeling 

AmherstSec Feb 2021

Matthew Coles @coles_matthewj
Izar Tarandach @izar_t

About Us

Izar Tarandach

- Principal Security Architect/Engineer
- Doing the security thing since the 90's
- Currently focusing on modern SDLC's

Matthew Coles

- Product Security Leader, Architect, Engineer
- Enabling and influencing security for physical devices and the ecosystems that enable them

Collaborators and colleagues since 2010

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

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

Standard disclaimer applies: Our views not our respective employers'

Agenda

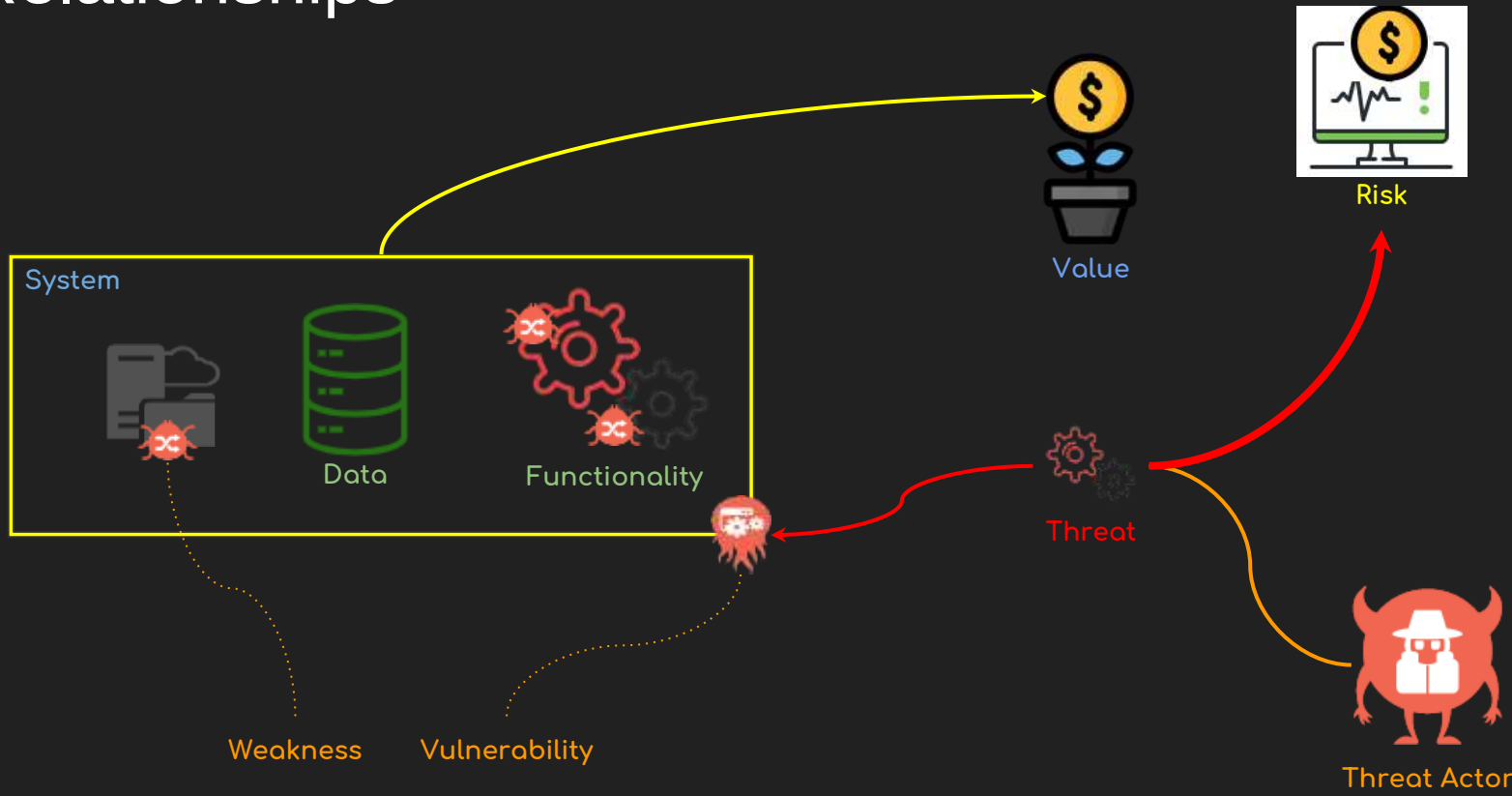
Fundamentals

Methods

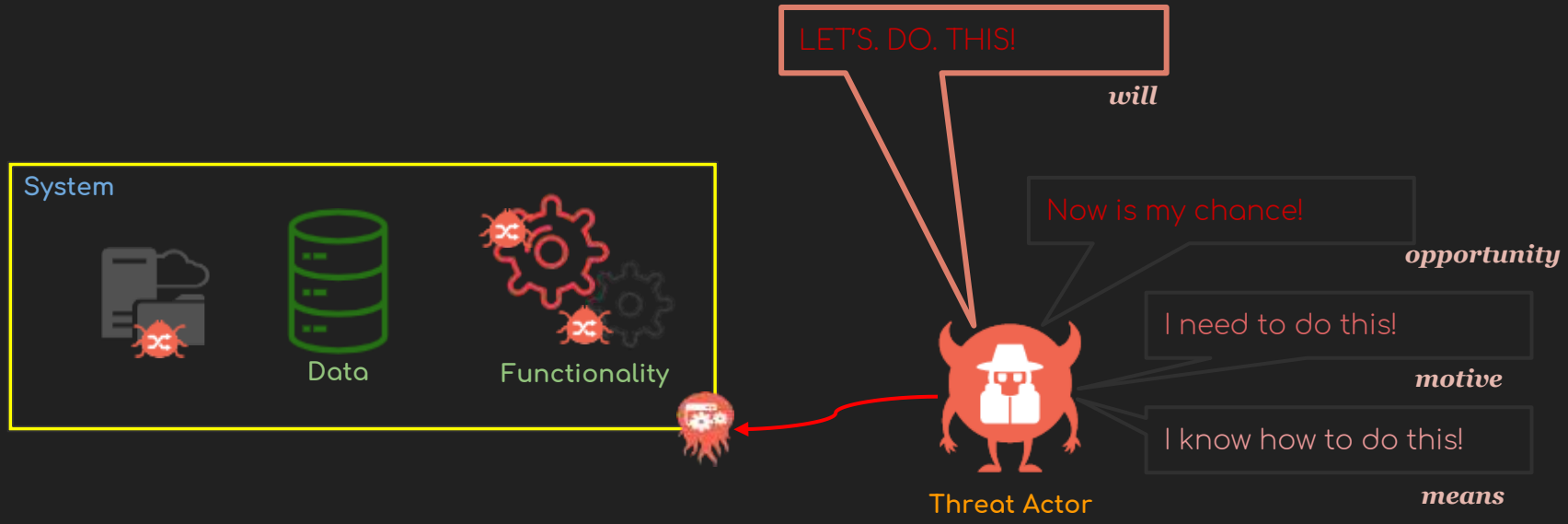
Demo

We enjoy interactive presentations, so we include plenty of time for questions.

Relationships



Understanding Risk



Security

Data

Functionality

confidentiality

integrity

availability

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

identification
authentication
authorization
audit

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

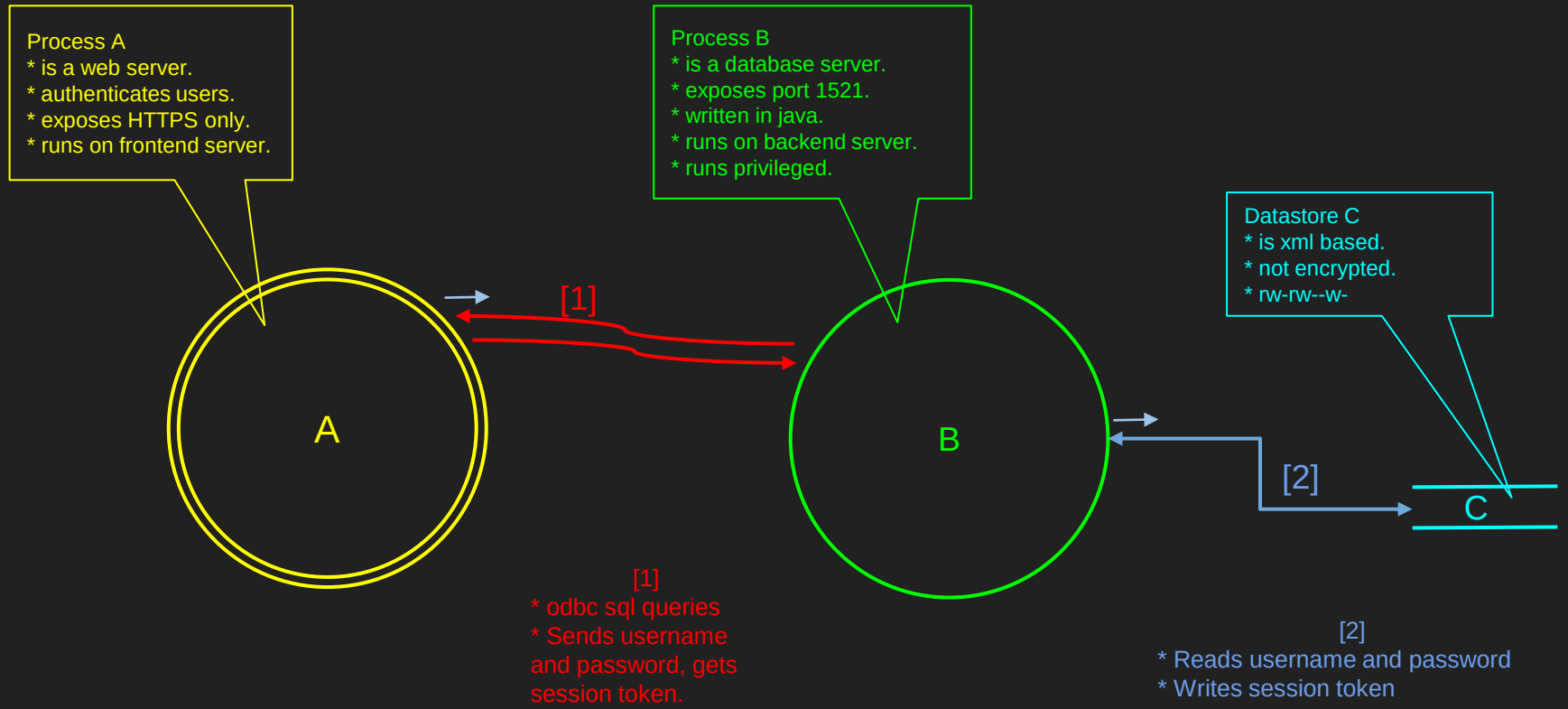
defense in depth
least privilege
fail secure
complete mediation

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

Modeling


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-
A
B
C
[1]
[2]
[1]
* odbc sql queries
* Sends username and password, gets session token.
[2]
* Reads username and password
* Writes session token

Analysis

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

“Threat Model Every Story”

TARA

Threat
Assessment &
Remediation
Analysis

Focus on Assets vs adversary Tactics, Techniques, and Procedures (TTPs)

Uses catalogs for TTPs and Countermeasures

Starts with a Model to Analyze

Spreadsheet Based

THREAT MODELING MANIFESTO

“Threat modeling is analyzing representations of a system to highlight concerns about security and privacy characteristics.”

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

5 values

4 principles

5 positive patterns

4 anti-patterns

Behind-the-Scenes

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

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

Values

A culture of finding and fixing design issues		checkbox compliance
People and collaboration		processes, methodologies, and tools
A journey of understanding	over	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.

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.

Perfect Representation
There is no single ideal view.

Making of pytm

Schema

Element

type
name

purpose
role

exposes...

privileges
uses...

contains...

connects to...
listens

Rules

```
Weakness w =

(target.type == Process &&
 target.privileges == "root" &&
 len(target.exposes) > 0
)

Threat t =

(target.exposes.port == 80 &&
 source_data.is_hci()
)
```

Engine

Loader

Parser

Sequencer

Analyzer

Renderer

Calculator

Reporter

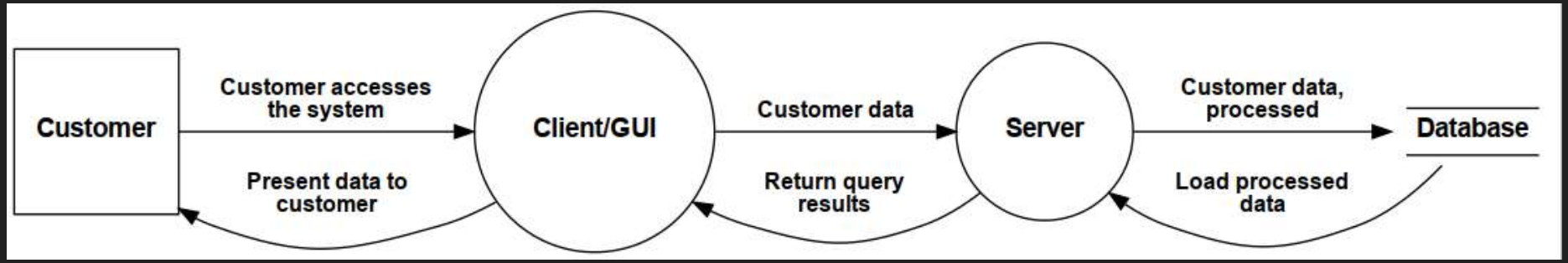
Demo - using pytm

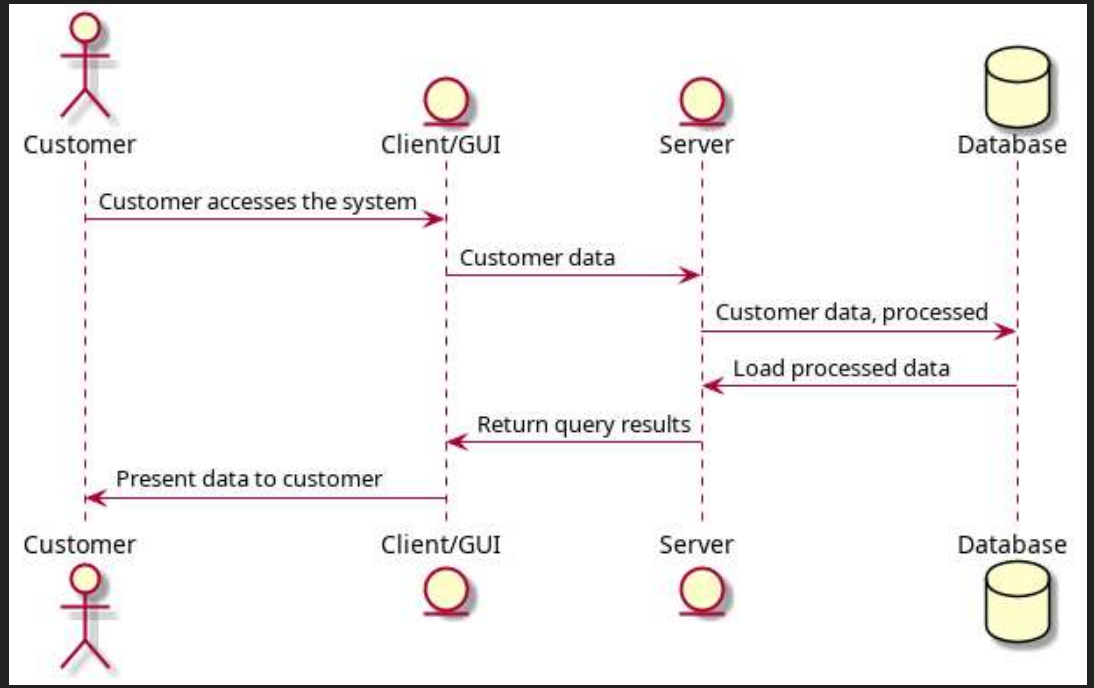
1. Define the components of the model and their relationships (dataflows)
2. Generate a dataflow diagram or a sequence diagram
3. Annotate the components with their attributes
4. Generate a report with the threats identified as a function of component and dataflow attributes

```
#!/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()
```

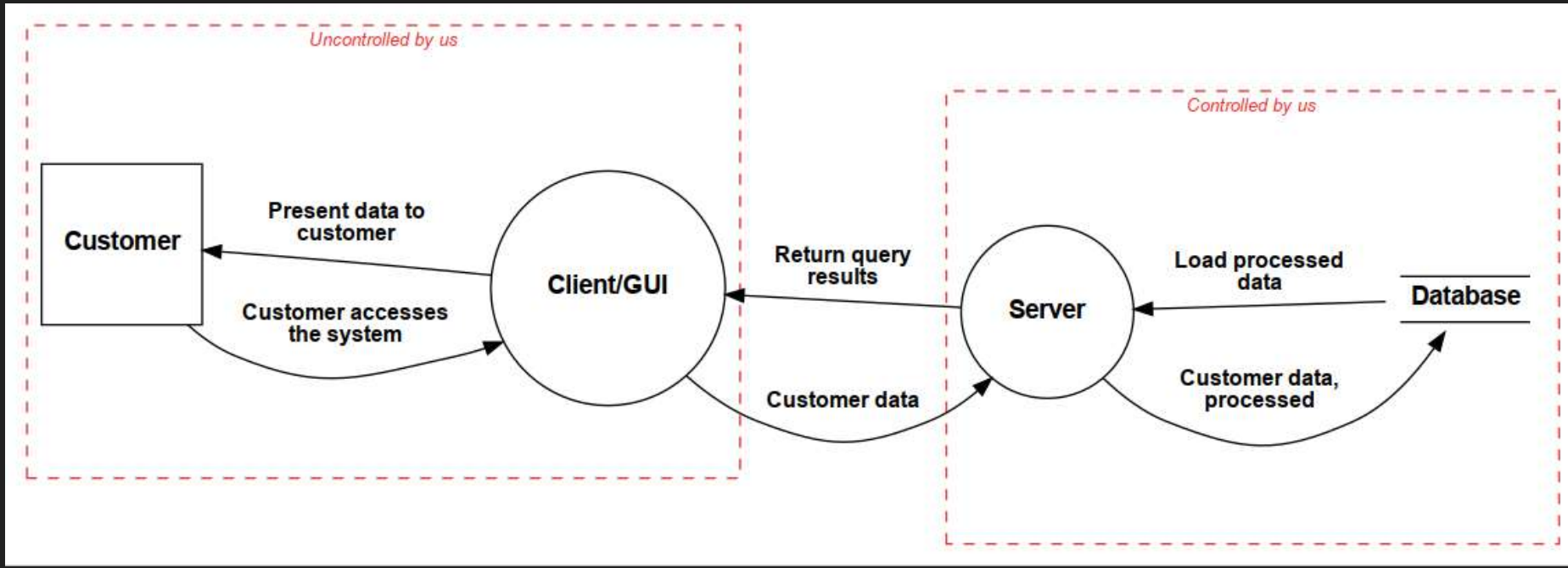
```
tm = TM("TM Demo v0.0.1")

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
Database
Customer data, processed

```
server.OS = "Ubuntu"
server.isHardened = True
server.sanitizesInput = False
server.encodesOutput = True
server.authorizesSource = False

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

enterData.protocol = "HTTP"
enterData.dstPort = 80
enterData.data = "New items to be stored, in JSON format"

saveData.protocol = "MySQL"
saveData.dstPort = 3306
saveData.data = "MySQL insert statements, all literals"

tm.process()
```

```
## Potential Threats

 
 

|{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>
   
   
   
</details>
}|
```

Potential Threats

|{findings:repeat:

▼ {{item.id}} -- {{item.description}}

Targeted Element

{{item.target}}

Severity

{{item.severity}}

Example Instances

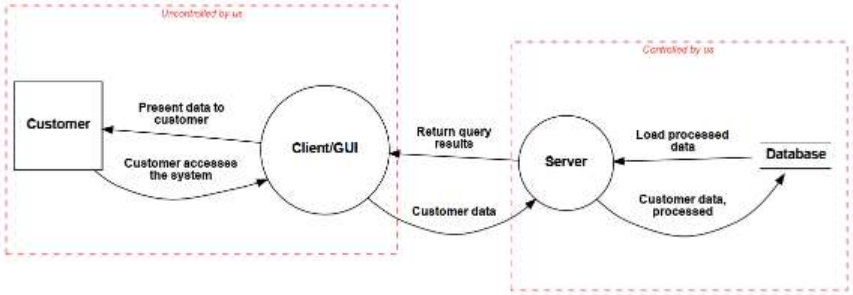
{{item.example}}

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

Questions?

THANK YOU!

Resources

- 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
- “Autodesk Continuous Threat Modeling”, https://github.com/Autodesk/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