Threat Modelling First Aid

Threat Modelling for Total Beginners

Petra Vukmirovic - Glasswall InfoSec Team

Dinis Cruz - Glasswall CISO

THREAT MODEL

PROCESS TO IDENTIFY VULNERABILITIES ON AN APPLICATION AND MITIGATE THEM

IF THE VULNERABILITIES, THREATS, RISKS AND CONTROLS ARE IDENTIFIED YOU ALREADY HAVE A THREAT MODEL!

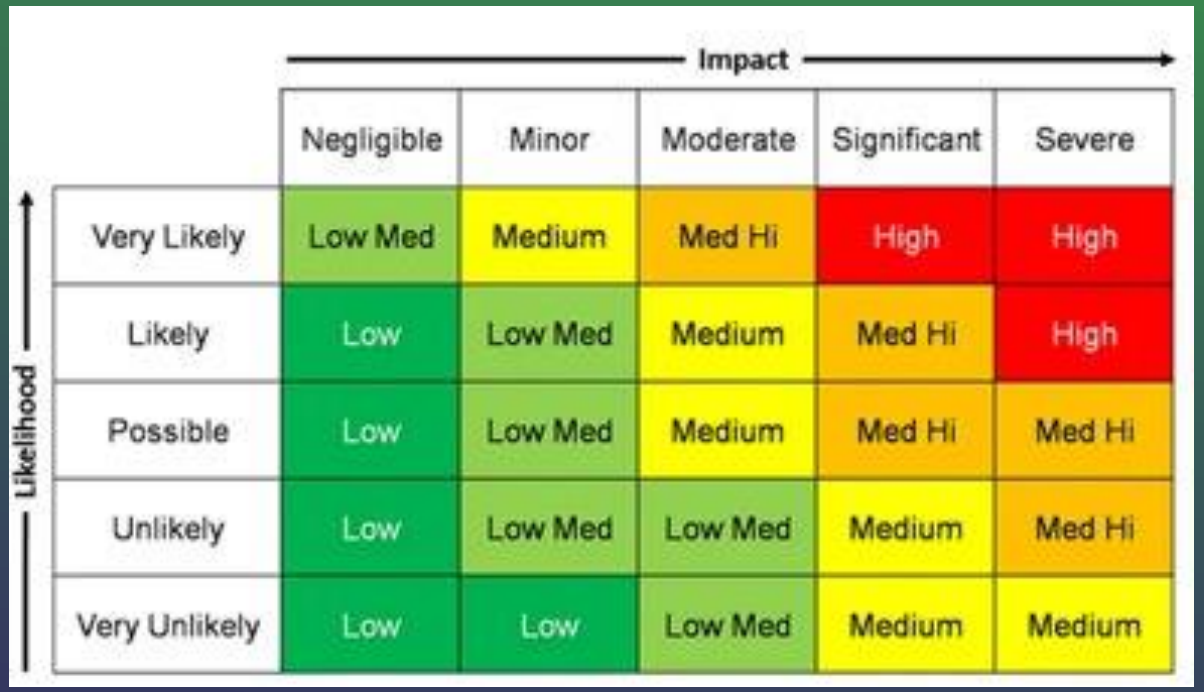
STRIDE

ATTACK TREES

PENTEST ON PAPER

THREAT MODEL

- Vulnerability
  - A flaw in architecture – a weakness that can be exploited
  - It can be a security bug in code, a flaw in the operating system, in processes, hardware, in a database, browser, or within people
- Threat
  - Any vector or agent that has the potential to take advantage of a vulnerability



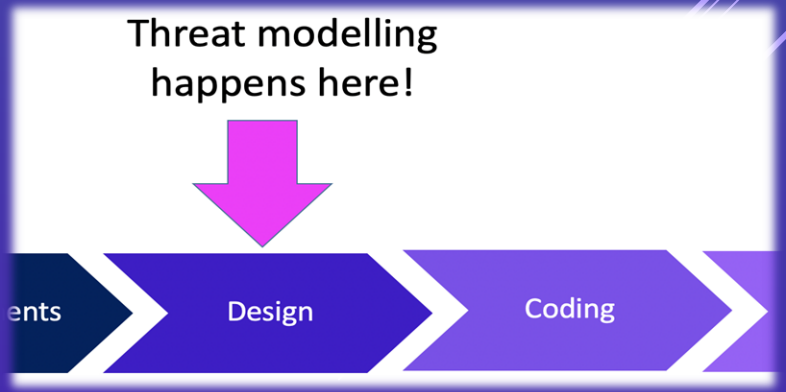
Likelihood \ Impact	Negligible	Minor	Moderate	Significant	Severe
Very Likely	Low Med	Medium	Med Hi	High	High
Likely	Low	Low Med	Medium	Med Hi	High
Possible	Low	Low Med	Medium	Med Hi	Med Hi
Unlikely	Low	Low Med	Low Med	Medium	Med Hi
Very Unlikely	Low	Low	Low Med	Medium	Medium

RISK

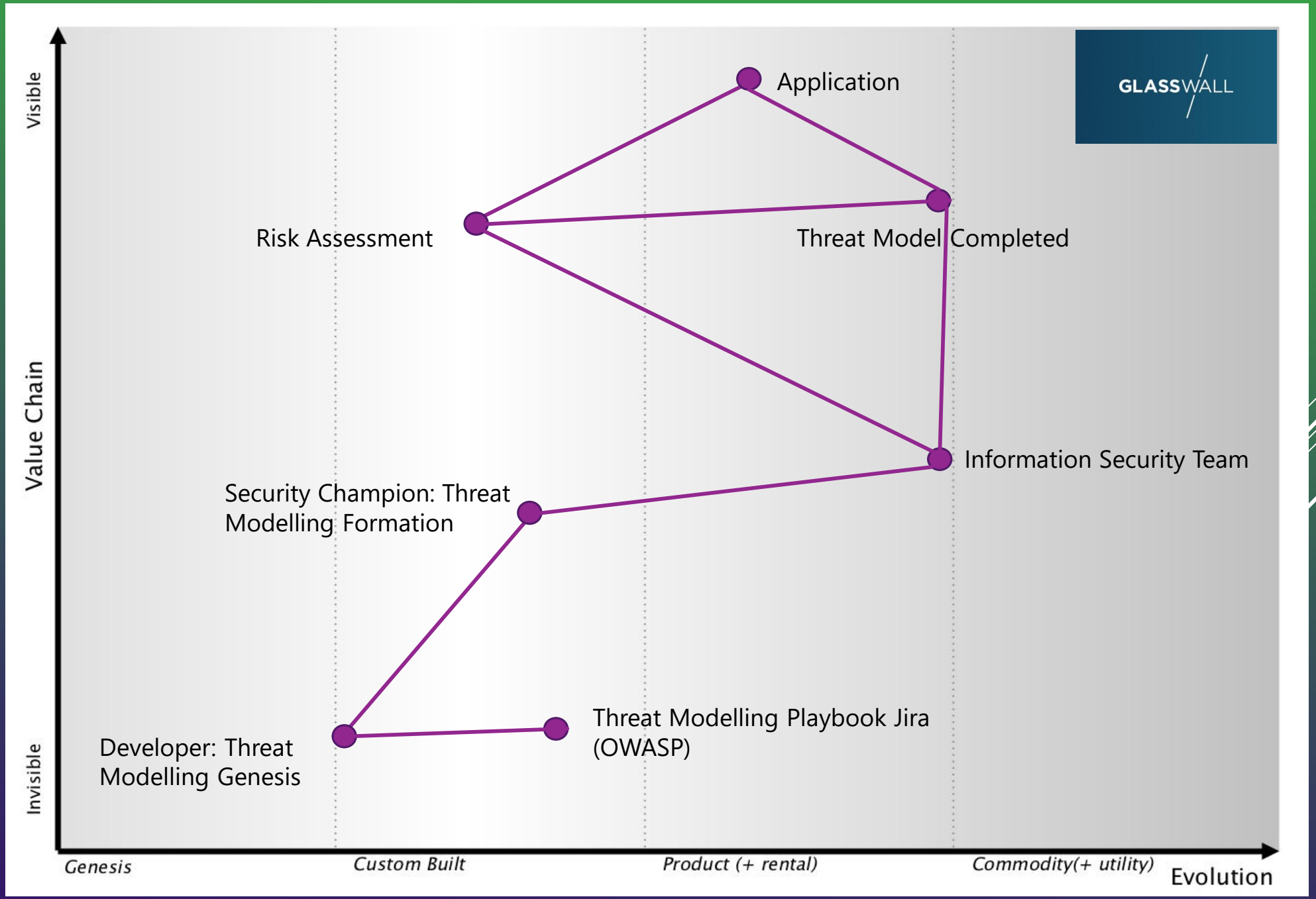
- Threat + Vulnerability = Risk
- Determine Risk Likelihood and Impact to calculate Risk Level

THREAT MODEL

- At the moment of launch every safe application should have a valid threat model already
- Developers should initiate threat model creation to delegate the risk to higher management which can then either accept the risk or ask for risk reduction controls to be implemented (tangible and intangible risks)
- In Glasswall Threat Modelling is initiated by the Development Team on Jira https://glasswall.atlassian.net/browse/PLAYBOOK-23
- InfoSec assistance can be sought on every step of the process



Threat Model

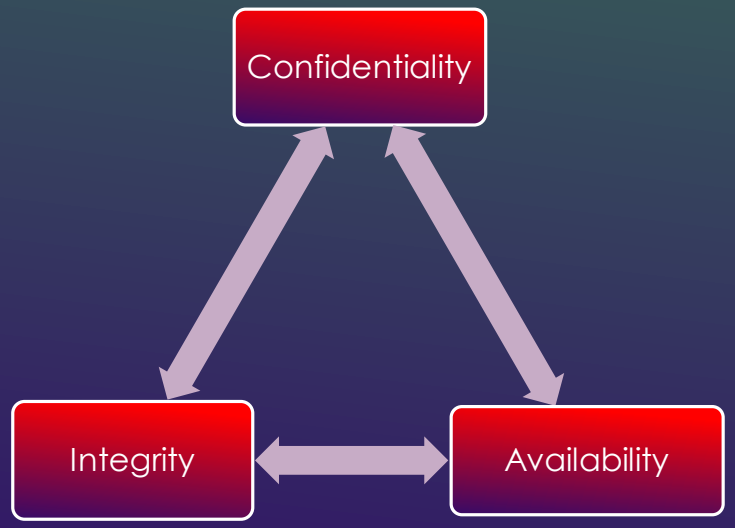


OPEN SECURITY SUMMIT

	Meaning	Description	Attacks
S	Spoofing	Impersonating another person/process	Cookie Replay, Session Hijacking, CSRF
T	Tampering	Unauthorised Alterations	XSS, SQLi
R	Repudiation	Denying Claims/Unproven actions	Audit Log Deletion, Insecure Backup
I	Information Disclosure	Exposure to unauthorised person/process	Eavesdropping, Path Traversal
D	Denial of Service	Service unavailability	Website defacement
E	Elevation of Privileges	Increasing person/process access level	Logic Flow Attacks, Buffer Overflow

THREAT MODEL IN PRACTICE (OWASP MODEL)

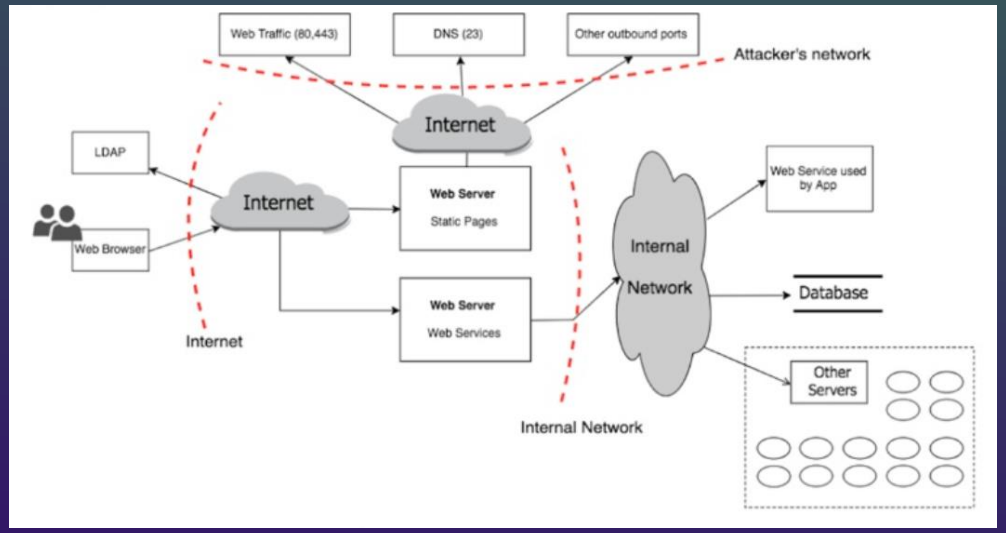
- Step 1: Profile the Application (Developer Role - Genesis):
  - Deployment Environment
  - Users
  - What are the Data Elements
  - What permissions will the actors have?
  - What technologies will be used? OS, Web/App Servers, Databases, Architectures, Programming Language
  - What security mechanisms apply? (CIA) – InfoSec team cooperation

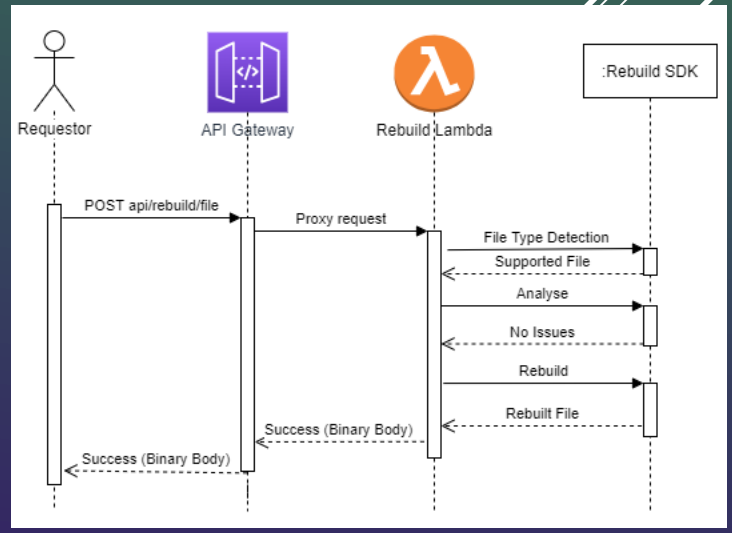


THREAT MODEL IN PRACTICE (OWASP MODEL)

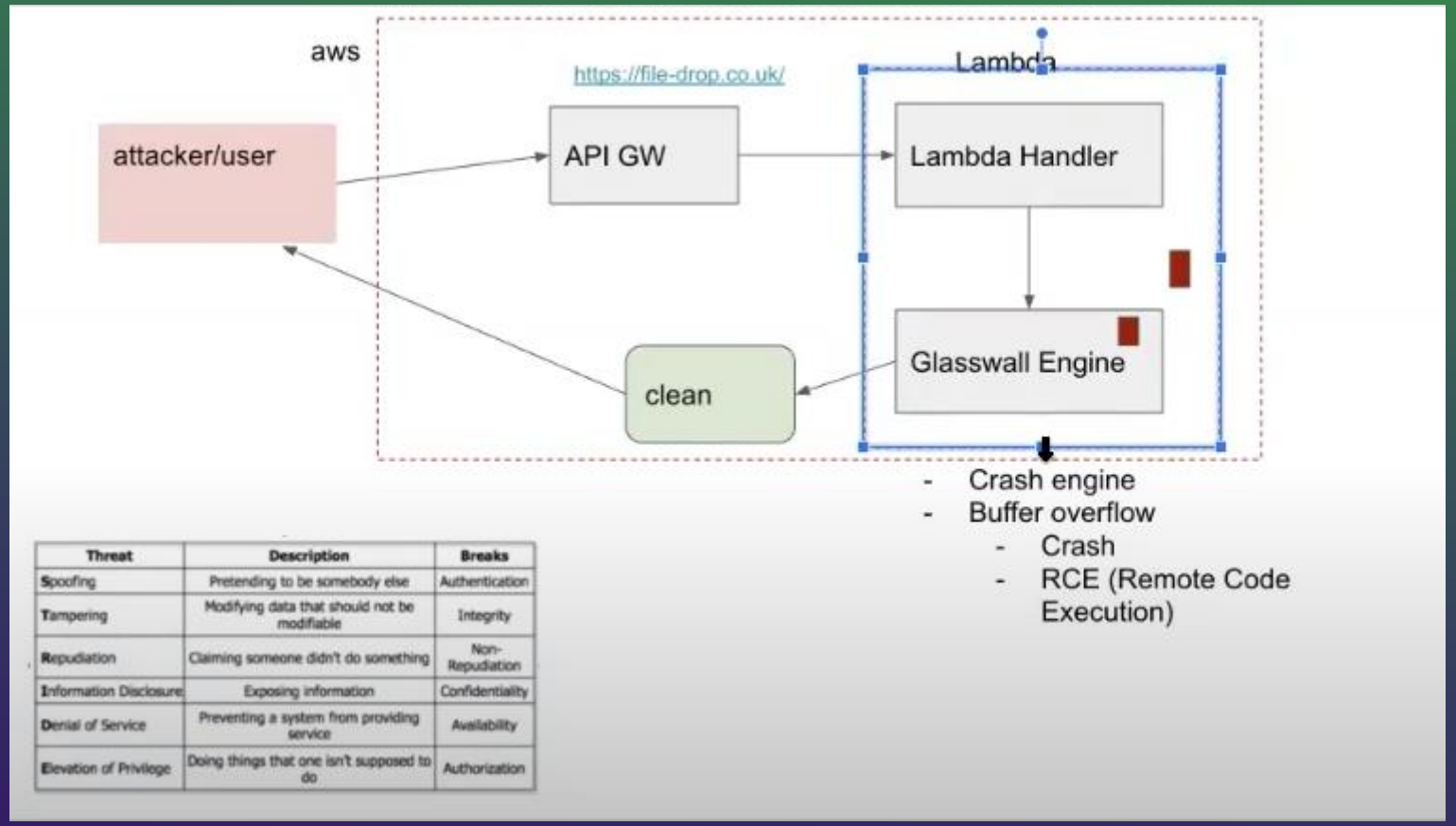
Step 2: Decompose Your Application (Developer Role - Genesis):

- Trust Boundaries (indicates where trust level changes)
- Firewall
- Webserver -> Database
- Your App -> External 3rd party Services
- Entry Points (Principal attack Targets)
- Ports, Pages, Components, APIs, Stored Procedures
- Exit Points
- Pages that display data, functions sending out values
- Data Flows
- Should we validate data at each node?





THREAT MODEL IN PRACTICE



Threat	Description	Breaks
Spoofing	Pretending to be somebody else	Authentication
Tampering	Modifying data that should not be modifiable	Integrity
Repudiation	Claiming someone didn't do something	Non-Repudiation
Information Disclosure	Exposing information	Confidentiality
Denial of Service	Preventing a system from providing service	Availability
Elevation of Privilege	Doing things that one isn't supposed to do	Authorization

IDENTIFY THREATS THAT APPLY ONLY TO THE APPLICATION CONTEXT & SCENARIOS GENERATED IN THE PREVIOUS STEP

USE ATTACK TREES

THINK LIKE AN ATTACKER (STRIDE/DREAD)

CREATE THE THREAT LIST: SQL INJECTION, XSS, REPLAY ATTACKS, MITM, EAVESDROPPING

STEP 3 IDENTIFYING THREATS (INFOSEC AND DEV)

THREAT MODEL IN PRACTICE (OWASP MODEL)

Step 4: Identifying Vulnerabilities (Infosec and Dev):

Identify vulnerabilities that apply only to the threats generated in the previous step

Vulnerabilities identified should be factored to shape the design of the application

Generate security test cases for testing in development stage (Typical vulnerabilities may be):

- Weak Encryption
- Clear Text Credentials
- Unhandled Exception
- Dynamic SQL
- Long Session Timeouts

THREAT MODEL IN PRACTICE

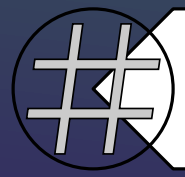
IN OTHER WORDS ASK YOURSELF SOME QUESTIONS:

Is the Application Vulnerable to OWASP Top 10 Web Application Threats?

- https://owasp.org/www-project-top-ten/

How safe is the API to intrusion? What protocols are used?

Are there any exposed API keys in the code or anywhere else?



Is the data encrypted at rest and transit?

What can an attacker do to get access to you Web Application and once access obtained, what further damage can be done to the application / customer data?

ONLINE THREAT MODELLING TOOLS

- https://online.visual-paradigm.com/drive/#diagramlist:proj=0&new=ThreatModelDiagram
- https://owasp.org/www-project-threat-dragon/
- https://github.com/filetrust/threat-model-cookbook
- https://www.owasp.org/index.php/OWASP_Threat_Model_Cookbook

LETS DO A THREAT MODEL NOW!

Scenarios:

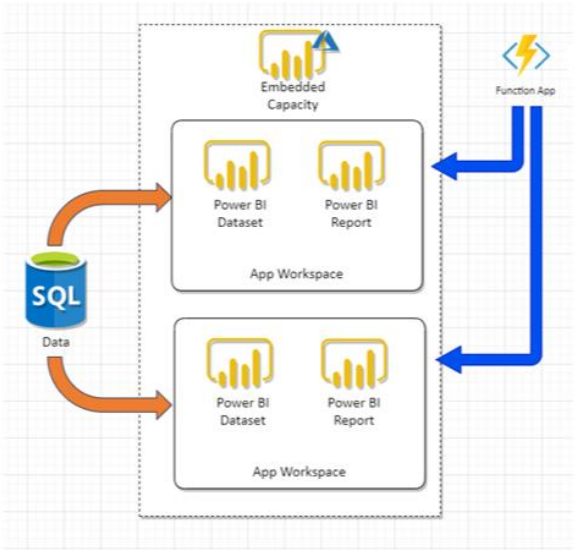
1. Threat Modelling Password Complexity With 2 FA
2. Using Lambda/Azure Functions to process malicious files
3. API Keys in Public GitHub Repos during an incident
4. Power Bi Scenario

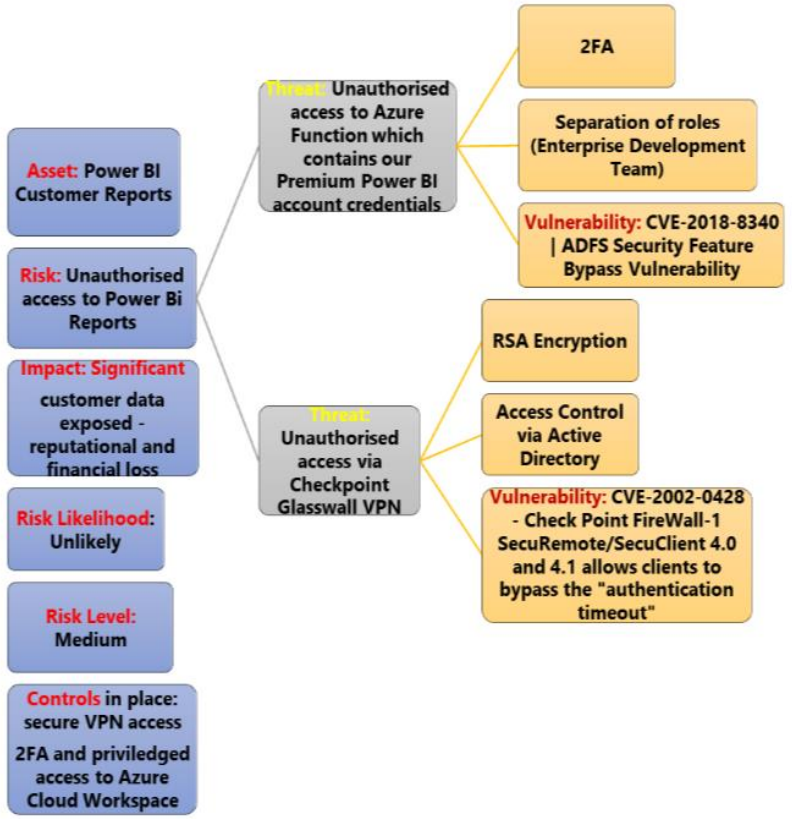
Do you have any questions for developers or business owners? Ask in breakout rooms!

POWER BI THREAT MODEL SCENARIO

Lets complete this threat model by asking the developers more questions ☺

Ask away!





OPEN SECURITY SUMMIT

WHAT IS NEXT?

- More About Threat Modelling
- Check out more about OWAS TOP – this is your secure coding bread and butter!
- Checkout the secure coding practice guideline & the SANS secure coding checklist
- Keep an eye on the most common vulnerabilities

REFERENCES

- D. Cruz, *Using Threat Models To Control Project Brief 2017*. Accessed at https://www.slideshare.net/DinisCruz/using-threat-models-to-control-project-brief

- OWASP, *Advanced Threat Modelling*. D. Cruz, *Using Threat Models To Control Project Brief 2017*. Accessed at https://www.slideshare.net/DinisCruz/using-threat-models-to-control-project-brief

- Microsoft, *Security Development Lifecycle 2010*. *Accessed at* https://www.microsoft.com/en-us/download/confirmation.aspx?id=12379

- OWASP, *Top 10 Web Application Security Risks 2017*. Accessed at https://owasp.org/www-project-top-ten/

- Microsoft, *Security Development Lifecycle 2010*. *Accessed at* https://www.microsoft.com/en-us/download/confirmation.aspx?id=12379

Thank You for Your Attention

Questions?