BSides Dublin

Threat Modeling: The Secret Sauce of an Effective Secure Software Development Life Cycle Programme

Shahidul Hoque

Who am I?

Senior Cyber Security Engineer at CoreHR

Working in security for last 7+ years

SDL, Threat Modeling, Vulnerability Management, Web Security etc.

R&D in security, data anonymization. Several papers

Love: Travelling, Biscuits, Reddit and Cricket!

Who is this talk for? And what're coming in the next slides?

Anyone – who like to know why threat modelling

Anyone – who are interested to leverage threat modelling in Secure Software Development Lifecyle (SDL/SSDL)

Anyone – who has confusion or hesitation on Threat Modeling!

Everyone – who loves Threat Modeling ☺

Threat modeling

Secure Software Development Lifecycle (SDL/SSDL)

Threat Modeling in Practice

Threat Modeling Driven Pentesting

Agile and Threat Modeling

Of Course, Q&A

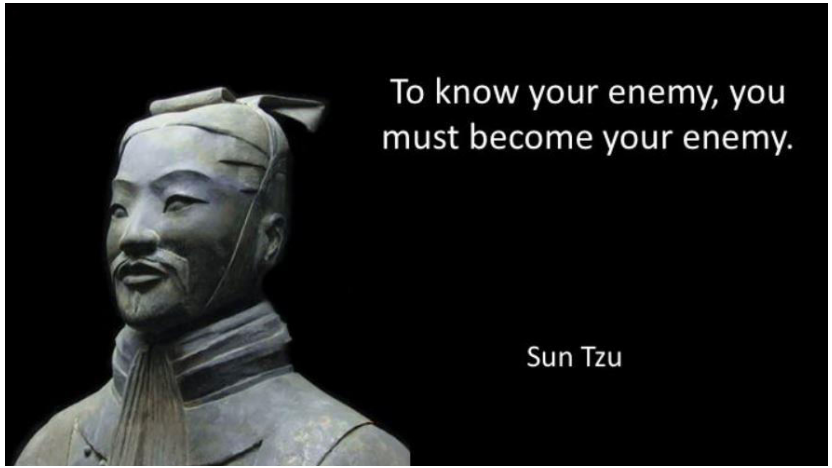
Wanna see a Threat Modeling in action at Convention?

Let's Talk about Threat Modeling; without definition!

Shared understanding of problems that could arise

Combined efforts in **reducing threats** such as potential attack use case generation by product managers and architect, defensive coding by developers, effective security test plan implementations by QA or security researchers

Reliable way in **measuring secure designs against new functionalities or features**, translating security efforts to **real measurable risk** and finding business-logic and system-level security issues



Threat modeling → Architectural Risk Analysis
Threat modeling → Applied Security Architecture
Threat modeling → Reducing Attack Surface
Threat modeling → Living document for SDL

What isn't Threat Modeling? With clarification!

Not an attacker model (Not a specific representation of how an attacker approaches a system but total system security

Not a test plan (Test plan can be derived from threat model but model itself offers a lot more than just test planning

Not a prescribed proof of system security (Can facilitate system security at best not full attestation)

Not a design review or code review (Threat models are the foundation of it, but Design review covers more implementation specific considerations beyond security and threat modelling

"The only truly secure system is one that is **powered** off, cast in a block of **concrete** and sealed in a **lead-lined** room with armed **guards** - and even then I have my doubts."

- Gene Spafford (aka Spaf)
Member of the Cyber Security Hall of Fame

DIY Toolbox to have with you

Tools: Microsoft TM (IDE), Tutamantic (discrete), IrusRisk (enterprise), Threat Modeler, Threat Dragon

Approaches: STRIDE, Kill Chain, Brainstorming, ATASM

Diagrams: Data Flow Diagram (DFD), Sequence Diagrams, State Diagrams

Threat Libraries: MITRE CAPEC (519 attack patterns), STRIDE (41 threats), Threat Modeler (many built in)

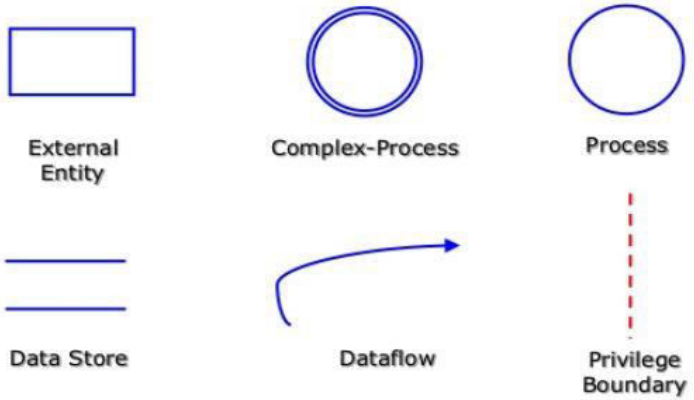
Needs to bring on your desk: Entry points, possible attackers and their perspective, external dependencies, assets, roles and trust levels

Identify: Sensitive data, privileged function, trust zone

Look out for: Proxies, facades etc. Services – web services, beans etc. UI vs implementation, aggressive caching schemes etc.

Frequent Terms and Symbols

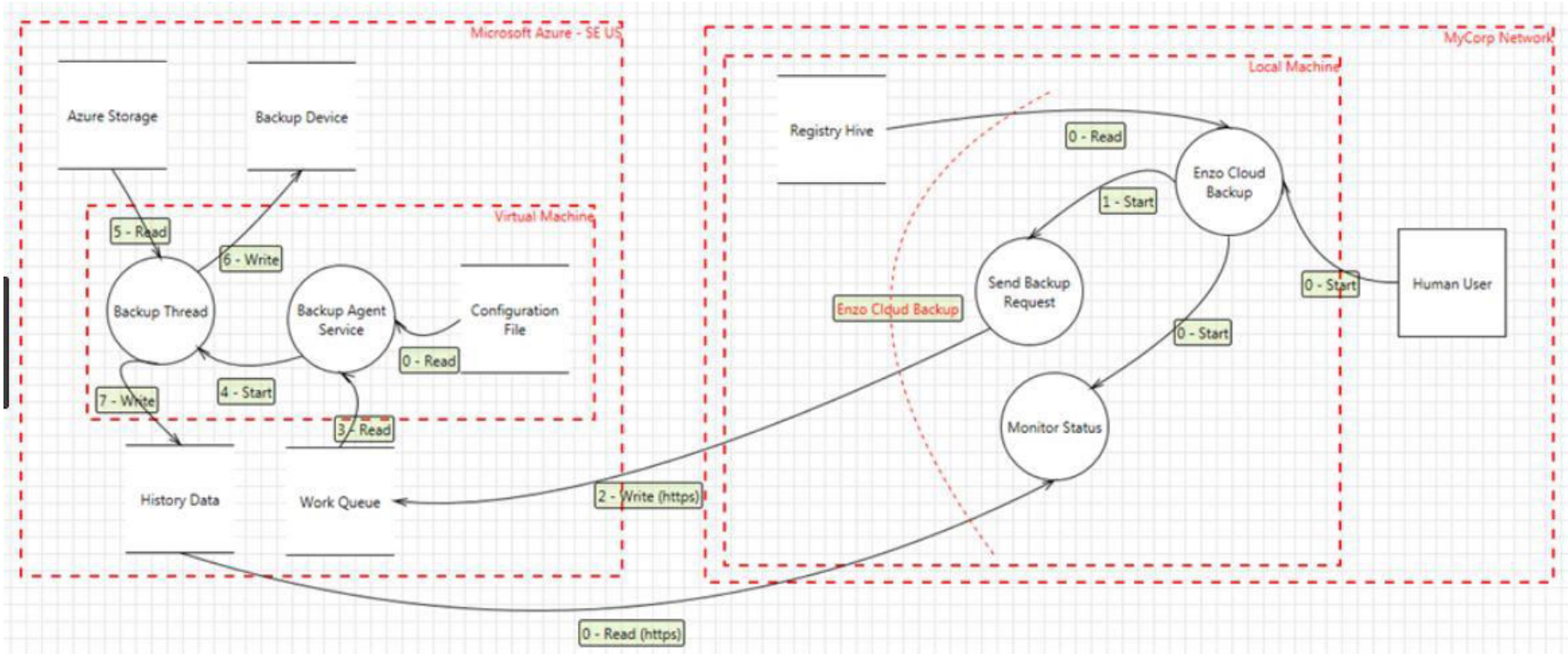
DFD Symbols



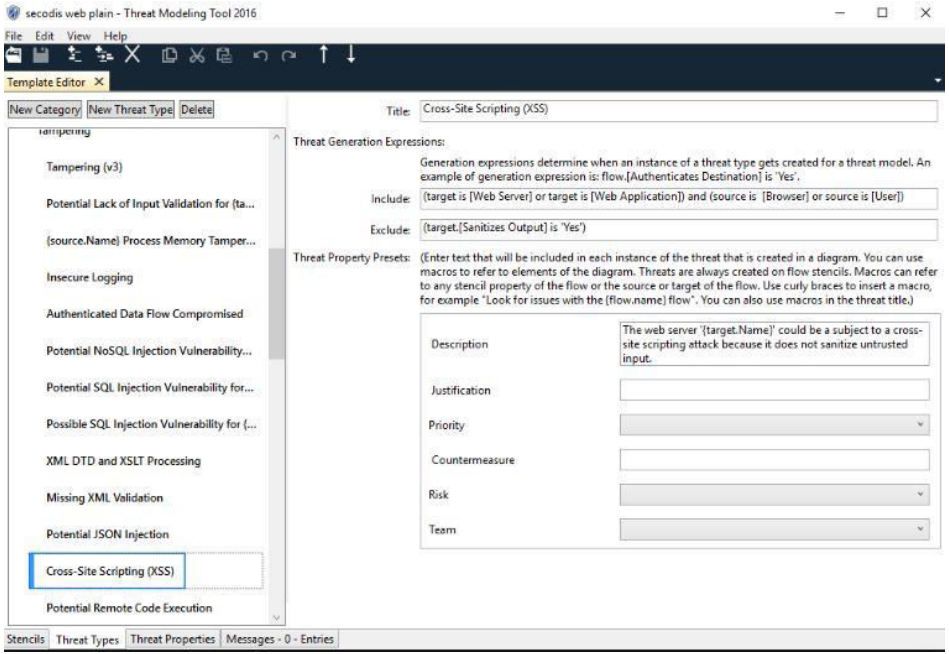
Key Definitions

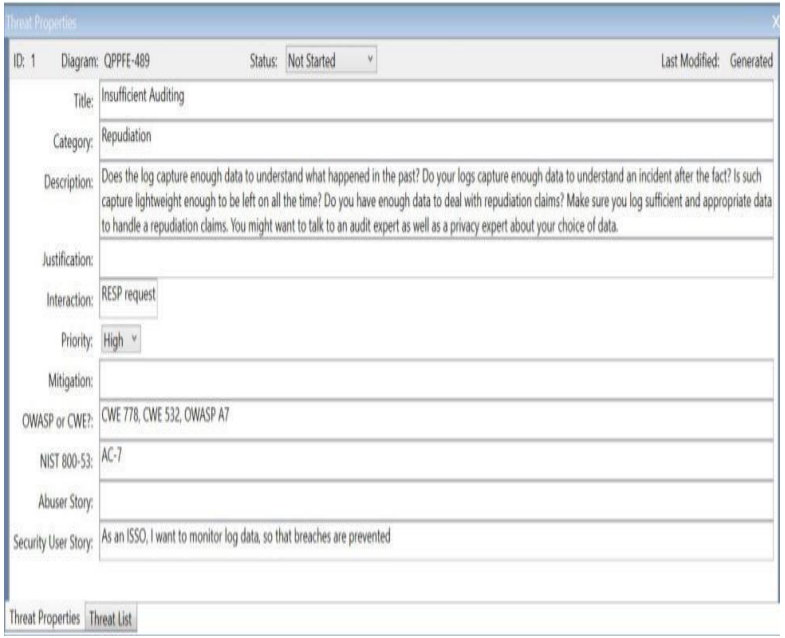
Term	Definition
Assets	The system must have something that the attacker is interested in
Roles	Roles are the "different trust-level" entities under which components interact or run within a system
Entry points	Entry points include any path through which an attacker can access the system
Trust Boundary	Trust boundaries indicate where trust levels change.

How a Threat Model will look like



On doing a Threat Model with Microsoft TM





How to do…

Identify assets that are potentially vulnerable and/or have insecure properties

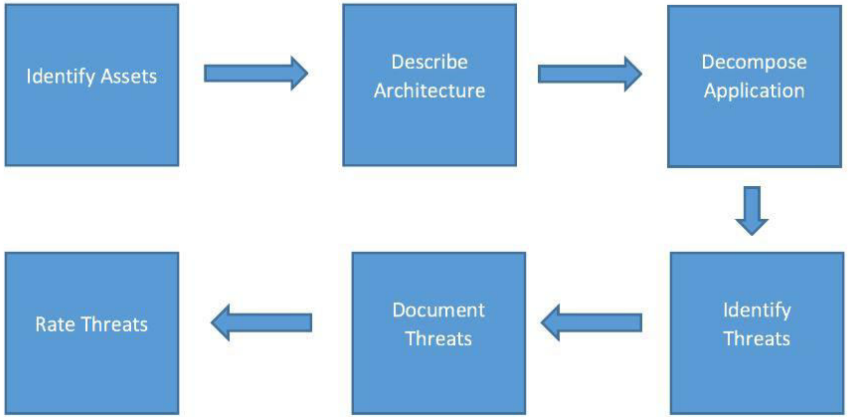
Draw down the architecture at first and describe as much as possible

Decompose Application so that in encompasses all the properties of it in granular level so that security aspects become more visible

Identify Threats against each components, data pipeline and stream of interactions

Document threats in details along with potential attack vectors

Rate threats so that the whole efforts become measurable in risk-



Design → Interactions → Threat → Risks

Two Decades of STRIDE, still striving to do Threat Modelin

Spoofing: Can an attacker gain access using a false identity? **[Authentication]**

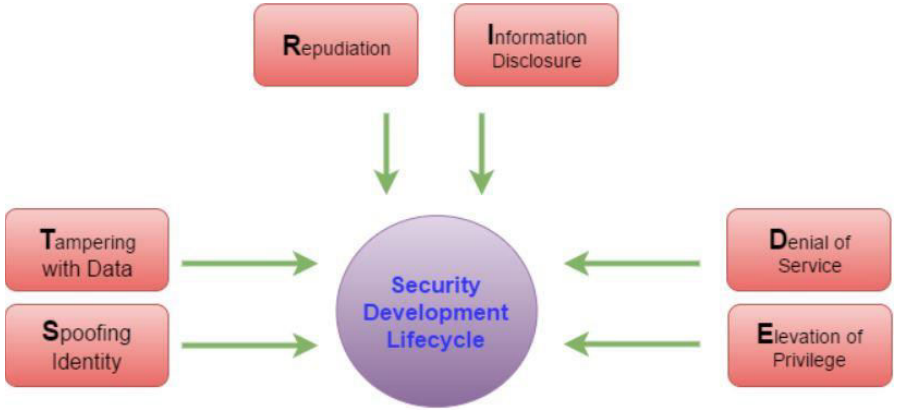
Tampering: Can an attacker modify data as it flows through the application? **[Integrity]**

Repudiation: If an attacker denies doing something, can we prove he did it? **[Non-repudiation]**

Information Disclosure: Can an attacker gain access to private or potentially injurious data? **[Confidentiality]**

Denial of Service: Can an attacker crash or reduce the availability of the system? **[Availability]**

Elevation of Privilege: Can an attacker assume the identity of a privileged user? **[Authorization]**



STRIDE in real terms

Spoofing Mitigations: Authentication – passwords, multifactor authN, digital signatures

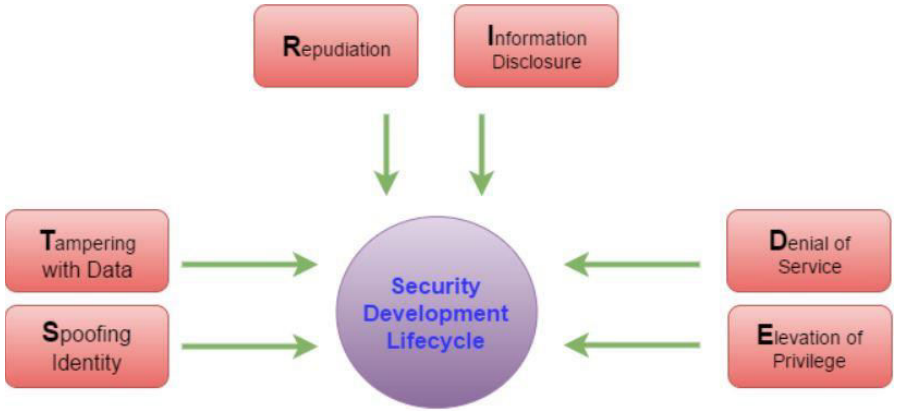
Tampering Mitigations: Integrity - Permissions/ACLs, Digital Signatures

Repudiation Mitigations: Non-repudiation - Secure logging and auditing

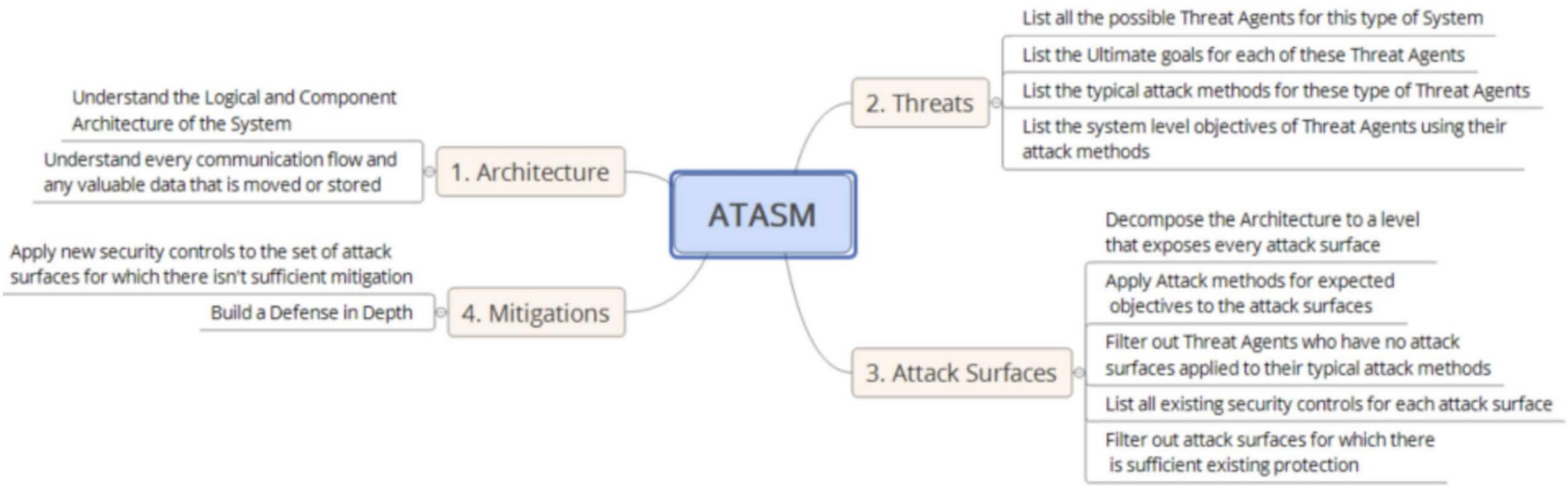
Information Disclosure: Confidentiality – Encryption, Permissions/ACLs

Denial of Service: Availability – Permissions/ACLs, Filtering with Analytics, Quotas

Elevation of Privilege: Authorization – Input Validation, Permissions



Why not try ATASM – Focus on Architecture



Securing Systems - Brook Schoenfield, 2015

Secure Software Development Lifecycle – SDL (Agile)

Security Definition of Done (DoD)

Security Architecture Review

Security Design Review

Threat Modeling

Security Testing and Validation

Static Analysis (SAST)

Dynamic Analysis – Web Apps (DAST) Fuzz Tes

Vulnerability Scan

Penetration Testing

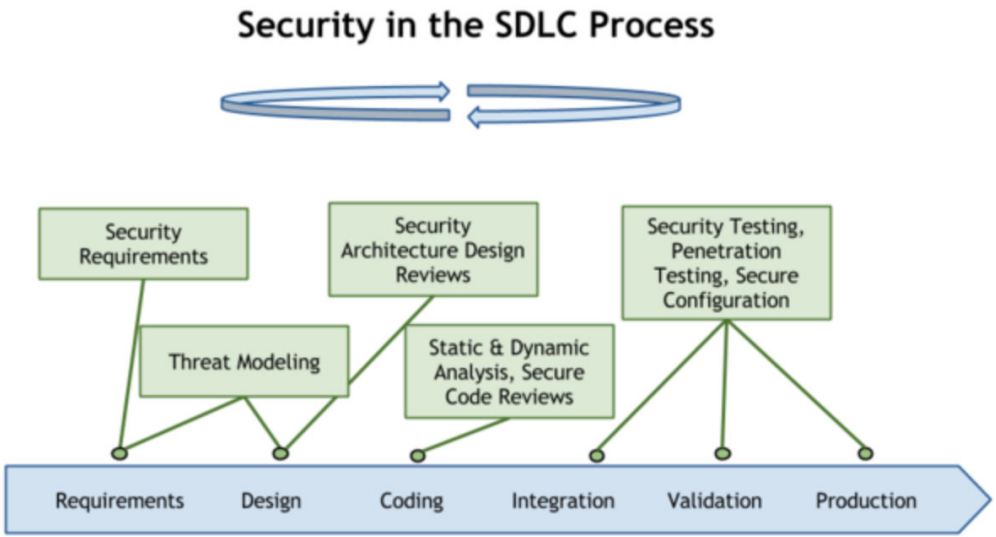
Manual Code Review

Secure Coding Standards (includes cryptograp

Open Source and 3rd Party Libraries

Vendor Management (includes legal compliance)

Privacy



Threat Modeling in SDL

Define Scope - SDL requirements (questionnaires, document)

Features, features, features and architectural changes

Research on each feature - need a team work or engagement

At the end of the day, it's all about **A Diagram! – Threat Modeling**

Details description of the diagram in Threat Modelling doc

Threat Modeling Each Story* – Security Labeling [subject area] plus checklist [security principles]

Why not clustering of stories based on architectural priorities or sensitive data points?

Agile story board into security backlogs

* AppSec Cali 2019

Risks from Threat Modeling

Describe each threats along with risk level

Risk = Threat * Vulnerability * Consequence

Threat and Vulnerability is based on Likelihood or Probability and Consequence is the potential impact of the threat/vulnerability. All of these outcomes

Risk in terms of Severity

Threat	Description	Vector	Prevalence	Detectability	Impact	Rating	Risk
TH – 01	• Credentials can be brute forced	2	2	3	3	7.00	High
TH – 02	• No security rules on password	2	2	2	3	6.00	Medium
TH – 03	• No SSL for Android App	2	3	2	2	4.67	Medium
TH – 04	• No SSL active for admin module	1	2	3	2	4.00	Medium
TH – 05	• No accountability of Drupal updates	3	2	2	1	2.33	Low
TH – 06	• API calls can be tampered with	1	1	1	2	2.00	Low
TH – 07	• Fake IDs can be used	1	1	1	2	2.00	Low

Low: 1-3, Medium: 4-6, High: 7-9

Threat Modeling Driven Penetration Testing

Associate **Bugs (BZ#), CVEs** with the Threat Model doc

Highlight the **severe or most critical points** within the threat model

Identify **high risk items** within the threat doc

Define test scope of most critical feature's outcomes

Submit to Pentester and/or re-evaluate from Pentester.

BSides Dublin

Thank you all

Any Question?

Find me: Shahidul.Hoque@corehr.com or shahidulhoque85@gmail.com or https://www.linkedin.com/in/shahidul-hoque-334a069/