



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Threat Modelling and Security Trends

Lecture 7

Trawsnewid Addysg; Trawsnewid Bywydau
Transforming Education; Transforming Lives



www.uwtsd.ac.uk

Learning Outcomes

- **Describe threat modelling concepts** by identifying assets, threats, and attack paths in modern systems.
- **Priorities security risks** using impact, probability, and exposure.
- **Apply threat modelling techniques** such as STRIDE, attack trees, and trust boundaries.
- **Select appropriate mitigations** based on risk level, system context, and organisational maturity.

What Is Cyber Threat Modelling?

- Threat modelling is a structured way to think about security risks before they cause damage.
- It helps us understand what we are protecting, what could go wrong, and how attackers might act.
- Think of it like planning home security: you consider doors, windows, and valuables before a break-in happens.

Security Risk Analysis and Assessment

- The goal is to understand risks so we can reduce them effectively.
- Start by identifying valuable assets such as data, systems, or services.
- Then assess how likely a risk is and how serious the damage would be if it happened.

Risk Impact Assessment

- Impact measures how bad the consequences would be if a risk occurred.
- We often use simple scales such as Low, Medium, High, or numbers from 1 to 5.
- Example: If confidential email is leaked to the press, the impact could be catastrophic.
- Ex:
 - Asset: Internal MD mailbox
 - Risk: Access to content by press
 - Impact: Catastrophic (5)

Risk Probability Assessment

- Probability estimates how likely it is that a risk will occur.
- Exact probabilities are hard, so relative values are often used.
- Example: A rare but serious event may have low probability but still need attention.
- Ex:
 - Asset: Internal MD mailbox
 - Risk: Access to content by press
 - Probability: Low (0.3)

Risk Exposure and Prioritization

- Risk exposure combines probability and impact.
- Exposure = Probability × Impact.
- High-exposure risks should be addressed first, while low-exposure risks may need minimal controls.
- Example:
 - Press may access MD mailbox:
Exposure = $P(\text{Low}=0.3) \times I(\text{Catastrophic}=5) = 1.5$
 - By the way, minimum exposure is 0.3 and maximum is 4.5 is our examples

Mitigation and Contingency Planning

- Mitigation reduces the likelihood or impact of a risk.
- Avoidance removes the risky asset or activity altogether.
- Contingency planning prepares actions for when something goes wrong.

Threat Modelling in Practice

- Threat modelling looks for vulnerabilities and possible attack paths.
- It originated in secure software development but now applies to systems and networks.
- The goal is to identify threats early and design effective defenses.

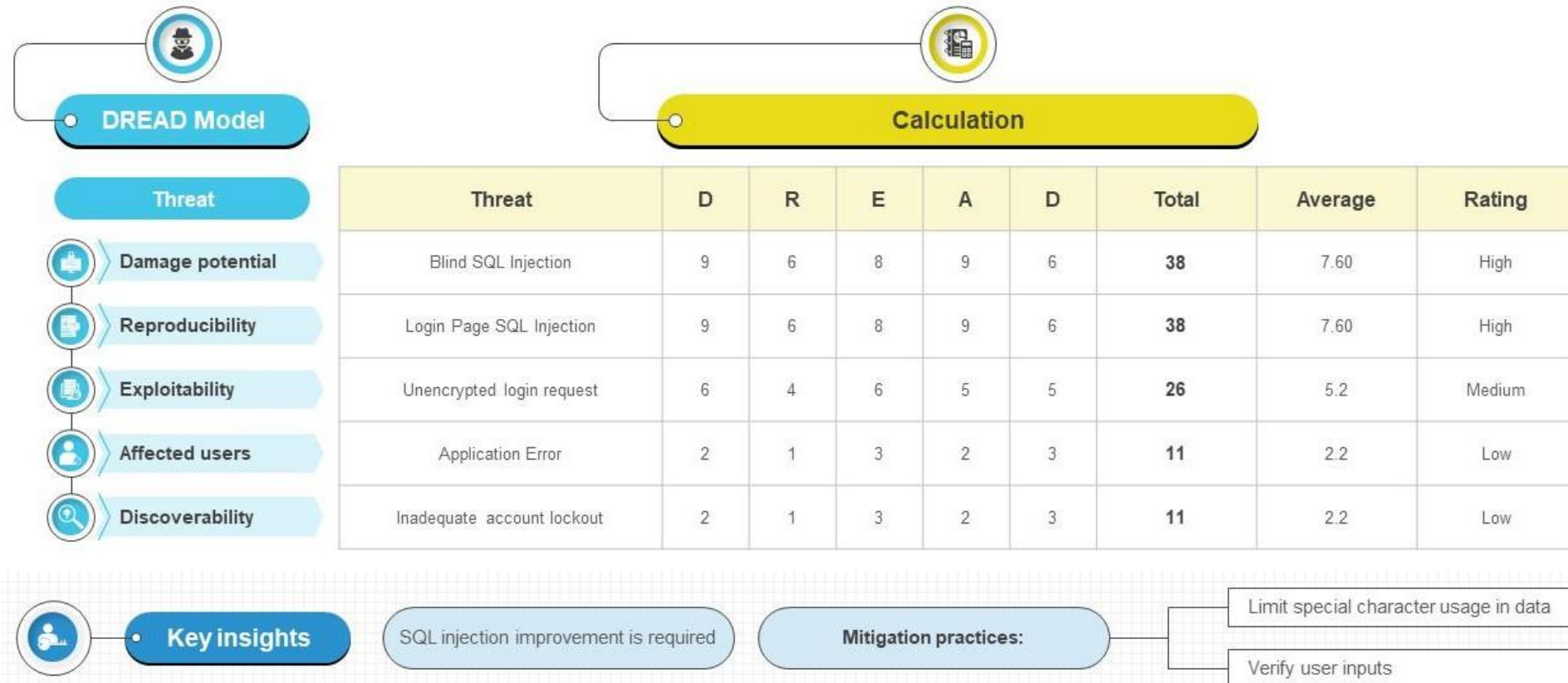
STRIDE and DREAD Overview

- STRIDE helps identify types of threats such as spoofing or data tampering.
- DREAD helps rate threats based on damage and likelihood.
- These frameworks provide a shared language for discussing security risks.

DREAD Methodology

<u>D</u> AMAGE	Impact of an Attack
<u>R</u> EPRODUCIBILITY	How Easily the Attack can be Reproduced?
<u>E</u> XPLOITABILITY	How Easy it is to Launch the Attack
<u>A</u> FFECTED USERS	How Many Users will be Impacted
<u>D</u> ISCOVERABILITY	How easily the vulnerability can be found

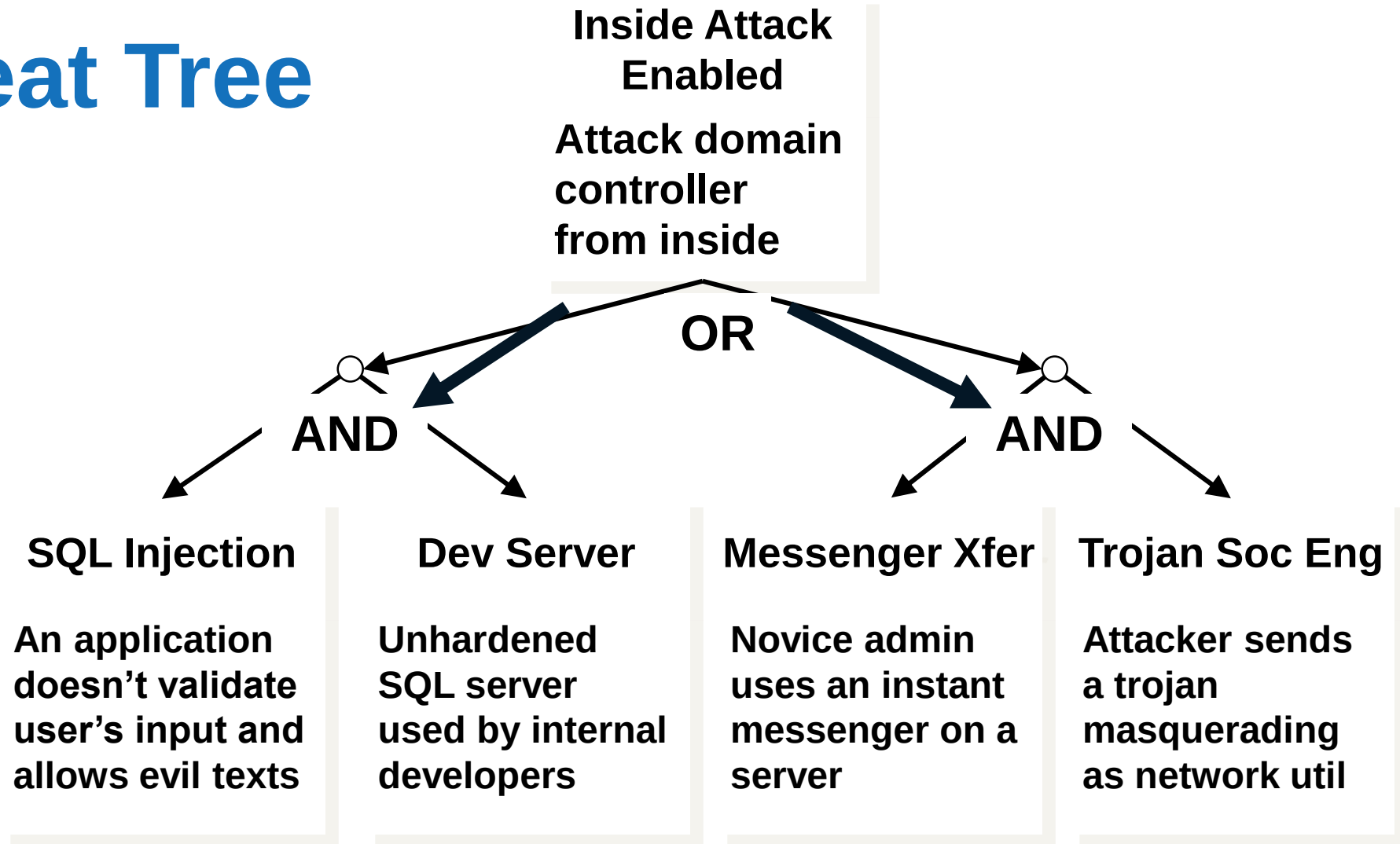
DREAD Threat Modeling in practice



STRIDE Threat Modeling in Practice

Type of Threat	Examples
Spoofing	• Forging Email Message • Replaying Authentication
Tampering	• Altering data during transmission • Changing data in database
Repudiation	• Delete critical data and deny it • Purchase product and deny it
Information disclosure	• Expose information in error messages • Expose code on web site
Denial of Service	• Flood web service with invalid request • Flood network with SYN
Elevation of Privilege	• Obtain Administrator privileges • Use assembly in GAC to create acct

Threat Tree



Modern Threat Modelling Methodologies

- Threat modelling is evolving to support agile and DevSecOps environments.
- Models are updated continuously rather than only at design time.
- **Example:** Updating threat models whenever new features are added to an application.

Common Challenges in Threat Modelling

- Threat modelling can feel complex or time-consuming at first.
- Teams may struggle with incomplete information or changing systems.
- A practical approach is to start small and refine models over time.

Threat Modelling Tools and Techniques

- Tools such as Microsoft Threat Modeling Tool help visualize threats.
- Attack trees and data-flow diagrams make complex systems easier to understand.
- These tools support collaboration between developers and security teams.

Case Study: Web Application Threat Modelling

- Consider an online login system as the asset.
- Threats may include password theft or SQL injection.
- Mitigations could include multi-factor authentication and input validation.

Emerging Threat Modelling Methodologies

- Threat modelling is increasingly embedded into Agile and DevSecOps practices.
- Rather than a one-time activity, models evolve as systems change.
- **Example:** Reviewing threat models at each sprint review when new features are released.

Common Challenges in Threat Modelling

- Limited system visibility can hide important threats early on.
- Time constraints often push teams to deprioritize structured analysis.
- A practical strategy is to begin with critical assets and refine over time.

Threat Modelling Tools and Techniques

- Data-flow diagrams clarify how information moves across trust boundaries.
- Attack trees decompose complex attacks into smaller, understandable steps.
- Tools support consistency, but expert judgment remains essential.

Case Study: Cloud Application Threat Modelling

- Asset: Sensitive customer data stored in a cloud-hosted database.
- Threat: Unauthorized access due to overly permissive access controls.
- **Mitigation insight:** Apply least privilege and continuous configuration monitoring.

Threat Modelling for Supply Chain Security

- Modern systems rely heavily on third-party software and services.
- Threat modelling must consider risks introduced by vendors and dependencies.
- **Example:** A compromised software update can introduce malicious code into trusted systems.

Abuse Cases vs. Misuse Cases

- Abuse cases describe how attackers intentionally exploit a system.
- Misuse cases focus on unintentional but harmful user behavior.
- Using both helps teams think beyond normal, expected system usage.

Trust Boundaries and Assumptions

- A trust boundary marks where data or control moves between different trust levels.
- Incorrect assumptions about trust are a common source of vulnerabilities.
- **Example:** Assuming internal network traffic is always safe.

Threat Modelling Metrics and Maturity

- Mature teams track how often threat models are updated and acted upon.
- Metrics help measure effectiveness, not just completion.
- **Insight:** A model that is reviewed regularly is more valuable than a perfect one used once.



Current Security Issues

Some Industry Issues

Without undue generalisation:

- Edge security at the data layer
- Malwares
- Compliance auditing
- Identity management
- Patch/update management
- Application defence
- Intrusion detection
- Integration

Mobile Security at Data Layer

- Laptops and PDAs are rarely protected against physical data extraction
- Encryption with removable keys is very effective, though deployment requires planning and is sometimes cumbersome
 - Smartcards plus EFS or an alternative system, such as PGP etc. can be applied
 - Data recovery needs (legal and practical) complicate the matter greatly

Spyware (Malware) Protection

70% of machines have malicious software, on average 28 separate spyware programs

- Zombies
- Network bandwidth and CPU degradation
- Commercial secrets leaked
- Privacy destroyed
- 3rd party liability arises

Best practice:

- SpyBot Search and Destroy (www.spybot.info)
- Microsoft AntiSpyware (in beta)
- AdAware
- Limit use of administrative privileges for end-users

Identity Management

- Heterogeneity of authentication and security measures is a common fact
 - Don't fight it, integrate it
- Synchronisation between directories, no matter how different, is becoming a reality with solutions built on systems such as MIIS (Identity Integration Server)
 - Alternatively, converge onto a client-solution, such as smartcards or OTP/tokens

Patch and Update Management

- Microsoft Update is fully functioning, and integrates, at present:
 - Windows OS updates
 - Office
 - SQL Server
 - Exchange
- More Microsoft products being added over the next months
- Enterprise solutions, however, will still benefit from a fully managed software distribution system, such as SMS (Systems Management Server)

Application Defence

- As networks and hosts become well protected, application-level attacks are on the increase
- Other than for very new in-house applications, development security has rarely been a concern
- This is a major area of worry from both perspectives of an insider and outside attacks
- Approaches:
 - Prove it's safe (threat modelling)
 - Isolate-and-monitor
 - Replace

Treating Unproven Applications

Until proven to be secure, treat all applications as “evil”

- Restrict access only to users on need-to-use basis
- Restrict remote use
- Isolate to dedicated application servers
 - Restrict servers through IPSec policies to only allow communication that applications explicitly require
- Monitor usage pattern to establish a baseline and raise alarm when patterns vary
 - Enable stringent auditing
- Request a formal threat analysis if above restrictions are too severe

Intrusion Detection

- Intrusion Detection Systems (IDS) are still fairly basic, though sophistication grew at network-level detection
- Honeypots, i.e. monitored vulnerable servers exposed as “bait” are still very effective, though may pose legal problems



Trends

MFA and Biometrics

- Multi-Factors Authentication (Phone, App ...)
- Useful as a secondary protection of a private encryption key on a smartcard in a *controlled environment*
- Advantage:
 - Simple and works in some environments, e.g. immigration control or secondary authentication of staff
- Weakness:
 - Not useful for at-home, remote etc. applications as no way to ensure it is your real fingerprint, iris, retina etc. being scanned
 - Biometric data can be stolen and can be used to fake identity – no way to change it later
 - Too many positive and negative false matches

Application-level Protection

- Developers are increasingly seen as responsible for security
 - This extends even to database developers, previously unlikely to engage in cryptography or ACL management
- It is very important that all in-house and vertical solution-provider application developers undergo security training
 - Refresher courses or workshops are a good idea
 - Community participation helps

Next Week

- **Get a clear**, real-world overview of common web application threats like **SQL injection**, XSS, and CSRF, and why they matter in practice
- **Learn how attackers exploit** weak input validation and broken authentication, with concrete examples you can easily follow
- **Explore practical techniques** to secure web applications across the application, server, network, and user layers
- **Understand essential security foundations** such as session management, HTTPS, and safe authentication practices you can apply right away



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Any Questions?

Trawsnewid Addysg; Trawsnewid Bywydau
Transforming Education; Transforming Lives



www.uwtsd.ac.uk