School of Computing
FACULTY OF ENGINEERING & PHYSICAL SCIENCES

UNIVERSITY OF LEEDS

COMP3911 Secure Computing

2: Threat Modelling

Nick Efford

https://comp3911.info

Slides are customised for XJCO3911 by Turgay Celik (Turgay.Celik@swjtu.edu.cn)

Objectives

UNIVERSITY OF LEEDS

- For you to appreciate the importance of **threat modelling** in secure software development
- For you to understand the value of **data flow diagrams** and **STRIDE** to explore threats
- For you to see how **attack trees** can be used for risk assessment and threat mitigation

Threat Modelling Goals

UNIVERSITY OF LEEDS

- Understand system's **threat profile**
- Facilitate secure design and implementation
  - Identify and justify need for security features
  - Prescribe implementation practices
- Guide code reviews and **penetration tests**
- Discover **vulnerabilities**

Three Different Perspectives

Can focus on

- Assets
- Attackers
- Software

Software-focused approach
is considered to be the best;
see Shostack's book for
detailed explanation

Assets

UNIVERSITY OF LEEDS

- “The valuable things you have”
- Can be tangible, obvious targets for attackers
  - User credentials
  - Credit card / bank account details
  - Confidential business data
  - etc…
- There are also **intangible things** you may need to protect (system availability, company reputation, etc)

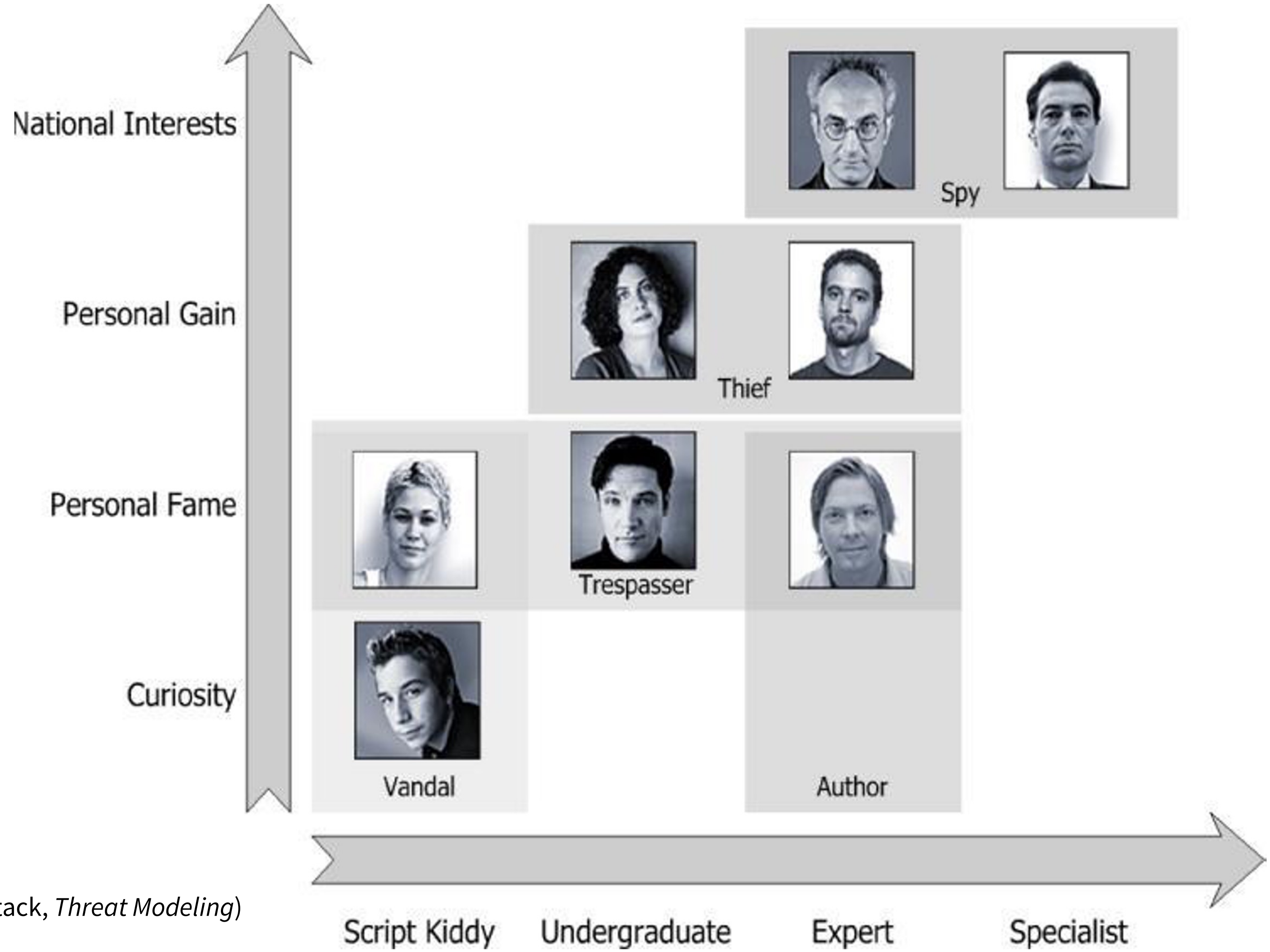
Attacker Lists

UNIVERSITY OF LEEDS

- Can help with brainstorming
- Talking about human threat agents helps to make the threat more real – useful when dealing with management?
- May not provide enough structure / detail to help people figure out what an attacker might do
- Easy for modeller to subconsciously project their own biases when creating or using an attacker list

Aucsmith's 'Threat Personas'

UNIVERSITY OF LEEDS



(from Shostack, *Threat Modeling*)

Thinking About Threats

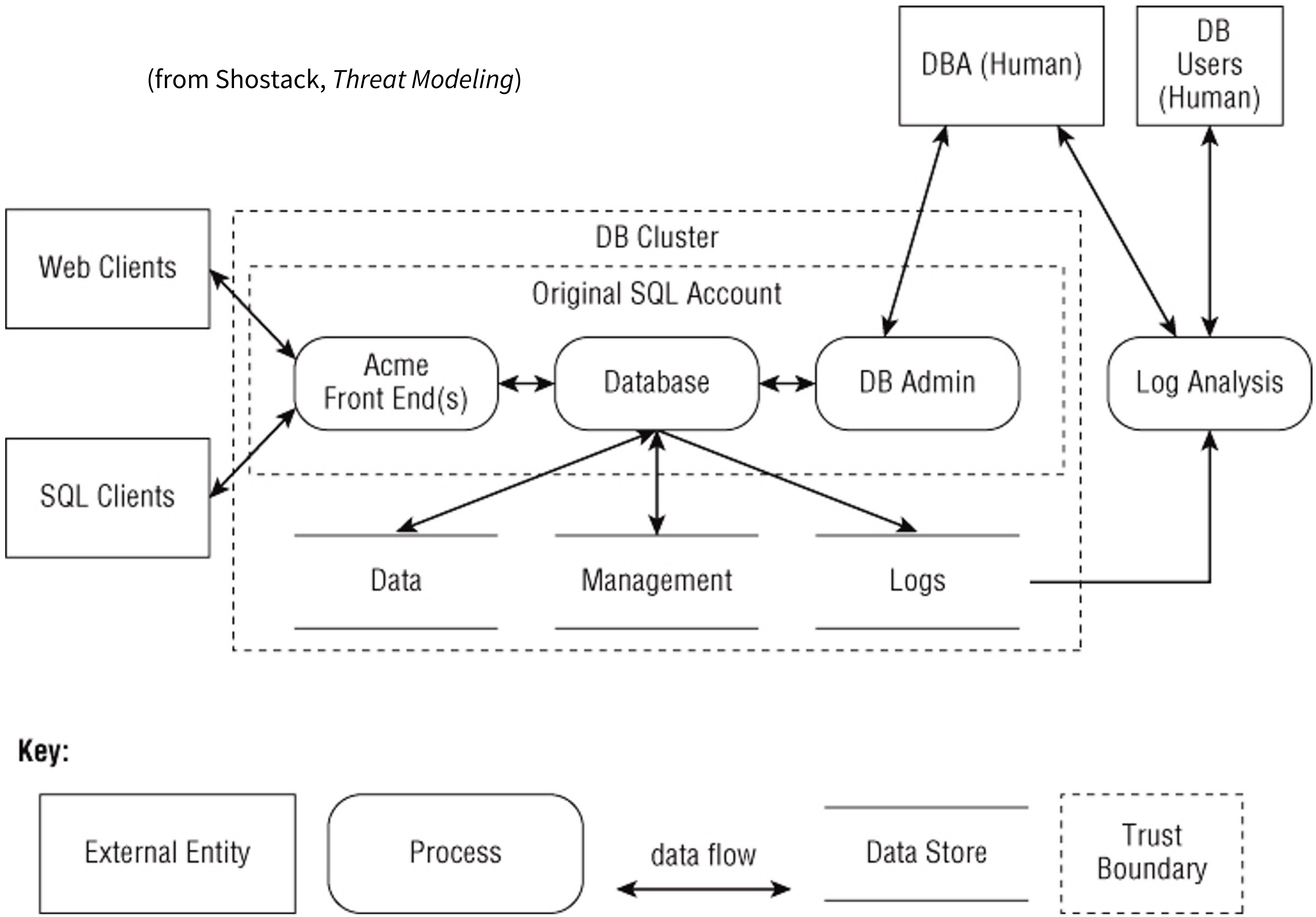
UNIVERSITY OF LEEDS

- Unstructured ('brainstorming')
- Moderately structured
  - Participants explore what could go wrong in a specific scenario, use case or user story
  - Protection Poker – analogous to the 'Planning Poker' game used by agile development teams
- Highly structured
  - Data flow diagrams
  - STRIDE
  - Attack trees

Data Flow Diagrams

UNIVERSITY OF LEEDS

(from Shostack, *Threat Modeling*)



Trust Boundaries

UNIVERSITY OF LEEDS

- These will occur at any point where entities with different levels of privilege interact
- TBs can typically be drawn around different user accounts, physical computers, VMs, network segments, etc
- To find them:
  - Identify the different **principals**
  - Start from either end of the 'privilege spectrum' (e.g., Internet user or system administrator)
  - Add a new TB each time a principal 'talks to' another

Entry & Exit Points

UNIVERSITY OF LEEDS

- Places where control or data crosses a trust boundary
- Easy to overlook infrastructure entry points!
  - Config files, Windows registry, etc
- Entry points can be layered

https://www.foo.com/login.jsp?cmd=NewUser

Port 443 of www.foo.com

Login page

NewUser operation

Why Is All This Useful?

UNIVERSITY OF LEEDS

- TBs delineate the **attack surface** between principals
- Threats tend to cluster around the entry/exit points on TBs
- … and can follow data flows
- So diagramming mainly helps us to learn, in a systematic way, **where we should look for threats**

Labelling Threats: STRIDE

UNIVERSITY OF LEEDS

- **S**poofing identity
- **T**ampering with data
- **R**epudiation
- **I**nformation disclosure
- **D**enial of service
- **E**levation of privilege

See pages 114–121 of the SDL book for more on STRIDE

Benefits of STRIDE

UNIVERSITY OF LEEDS

- Acts as a useful ‘checklist’ when considering what might threaten an element of a DFD
- Makes it easier to understand threat effects
- Helps to assign a priority to threats
    - EoP often the most serious
    - Repudiation may be critical in financial sector

STRIDE-Per-Element

UNIVERSITY OF LEEDS

- We can constrain STRIDE because certain threats are more prevalent than others for particular DFD elements
- Minimal goal is typically to find **one threat per tick**, for each element of a DFD

DFD Element Type	**S**	**T**	**R**	**I**	**D**	**E**
External entity	✓		✓			
Data flow		✓		✓	✓	
Data store		✓	?	✓	✓	
Process	✓	✓	✓	✓	✓	✓

Simpler Models

UNIVERSITY OF LEEDS

IIMF

Interception, Interruption, Modification, Fabrication

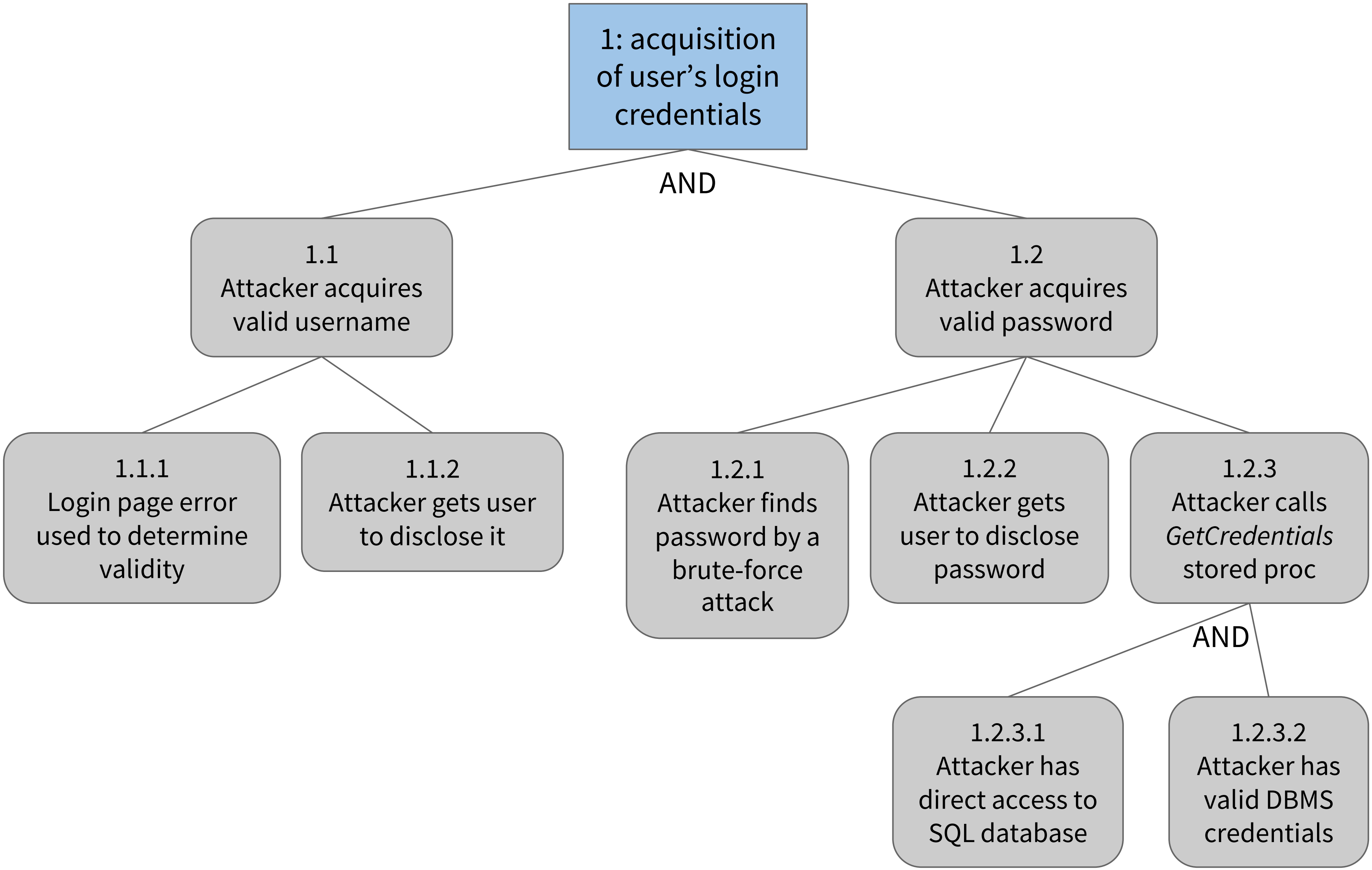
CIA

Confidentiality, Integrity, Availability

Attack Trees

UNIVERSITY OF LEEDS

- Root node, representing a threat
- Child nodes, each representing a **condition** that must be true for attack to succeed
- Arcs between parent and child nodes
    - Path from a leaf node to root is an **attack path**
    - Dashed line can be used to indicate low likelihood
- Boolean relationships between siblings
    - OR is implicit
    - AND must be shown explicitly


1: acquisition of user's login credentials
AND
1.1 Attacker acquires valid username
1.2 Attacker acquires valid password
1.1.1 Login page error used to determine validity
1.1.2 Attacker gets user to disclose it
1.2.1 Attacker finds password by a brute-force attack
1.2.2 Attacker gets user to disclose password
1.2.3 Attacker calls *GetCredentials* stored proc
AND
1.2.3.1 Attacker has direct access to SQL database
1.2.3.2 Attacker has valid DBMS credentials

Textual Representation

UNIVERSITY OF LEEDS

```
1 Threat
    1.1 (AND) Unmitigated condition
    1.2 (AND) Unmitigated condition
        1.2.1 Mitigated condition
                - Mitigation information
        1.2.2 Mitigated condition
    1.3 Unmitigated condition
        1.3.1 Mitigated condition
        1.3.2 Unmitigated condition
```

Easier to create, but harder to use

Can be made machine-readable – e.g., using XML or JSON

Risk Assessment: DREAD

UNIVERSITY OF LEEDS

For each threat, estimate

Damage potential

Reproducibility (how often an attempt at exploiting a vulnerability actually works)

Exploitability (effort required to exploit vulnerability)

Affected users (proportion of install base that would be affected if an exploit became widely available)

Discoverability (likelihood that a vulnerability will be found by external security researchers or black hats)

(Use a 1–3 or 1–5 scale for each, compute *R* as a simple average)

Issues With DREAD

UNIVERSITY OF LEEDS

- Values are highly subjective
- Not all dimensions are useful – e.g., why assume that Discoverability is anything other than 100%?
- Dimensions are weighted equally

CVSS

UNIVERSITY OF LEEDS

- Open standard: v2 and v3 in current use
- Scoring is more thorough and objective that DREAD
  - Uses a 0.0 – 10.0 scale
  - Provides a base score for severity, optionally modified to reflect risks in a specific environment
- Guided online calculators exist
  - https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator
  - https://www.first.org/cvss/calculator/3.0

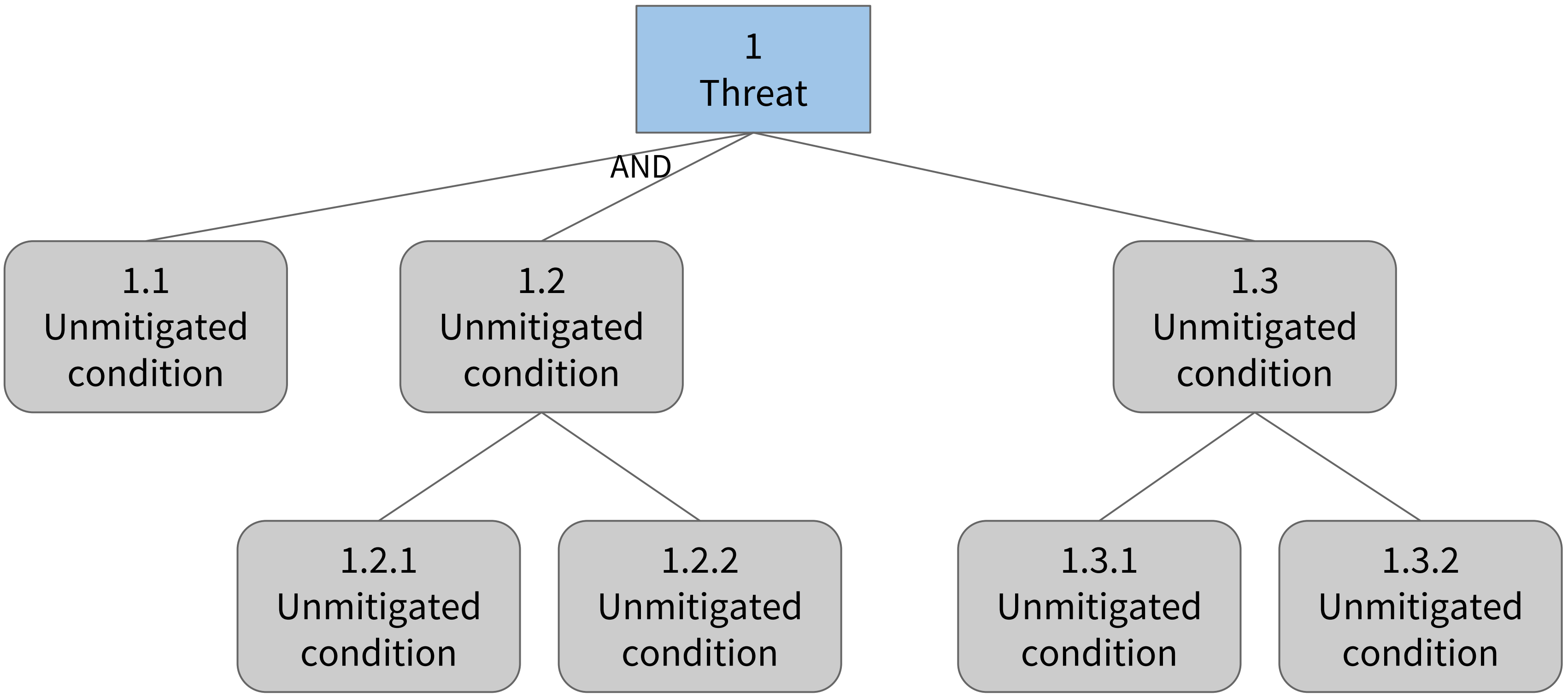
Mitigation

UNIVERSITY OF LEEDS

- Ideally, every threat is mitigated either by design or implementation of the system
    - Corollary: every security feature should address at least one threat from the threat model!
- Rank unmitigated threats by risk, and focus on mitigating the high-risk threats first

Mitigation

UNIVERSITY OF LEEDS



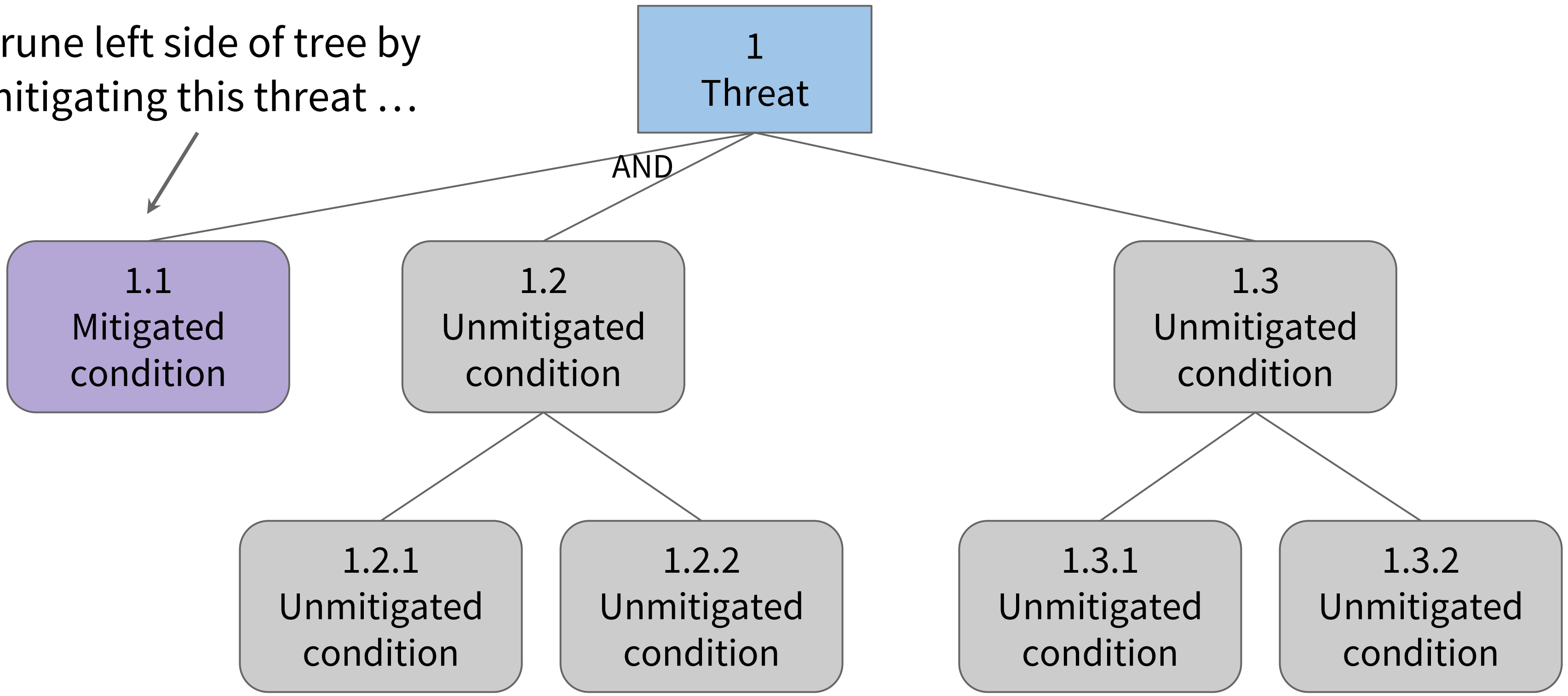
How many **attack paths**?

(unique paths from leaf condition to root node)

Mitigation

UNIVERSITY OF LEEDS

Prune left side of tree by mitigating this threat …



… halving the number of attack paths!

Summary

UNIVERSITY OF LEEDS

We have

- Seen how threats can be found by analysing software using data flow diagrams and STRIDE
- Explored the use of attack trees to indicate the possible attack paths for a threat
- Noted that unmitigated attack paths correspond to vulnerabilities
- Considered two approaches to risk assessment and seen how these drive the mitigation process

Follow-Up / Further Reading

UNIVERSITY OF LEEDS

- Howard & Lipner, *The Security Development Lifecycle*
- Shostack, *Threat Modeling: Designing For Security*
- Protection Poker: instructions and materials
- Microsoft Threat Modeling Tool
  - 2018 version: https://aka.ms/tmt
- An alternative approach: OCTAVE Allegro
- Common Vulnerability Scoring System