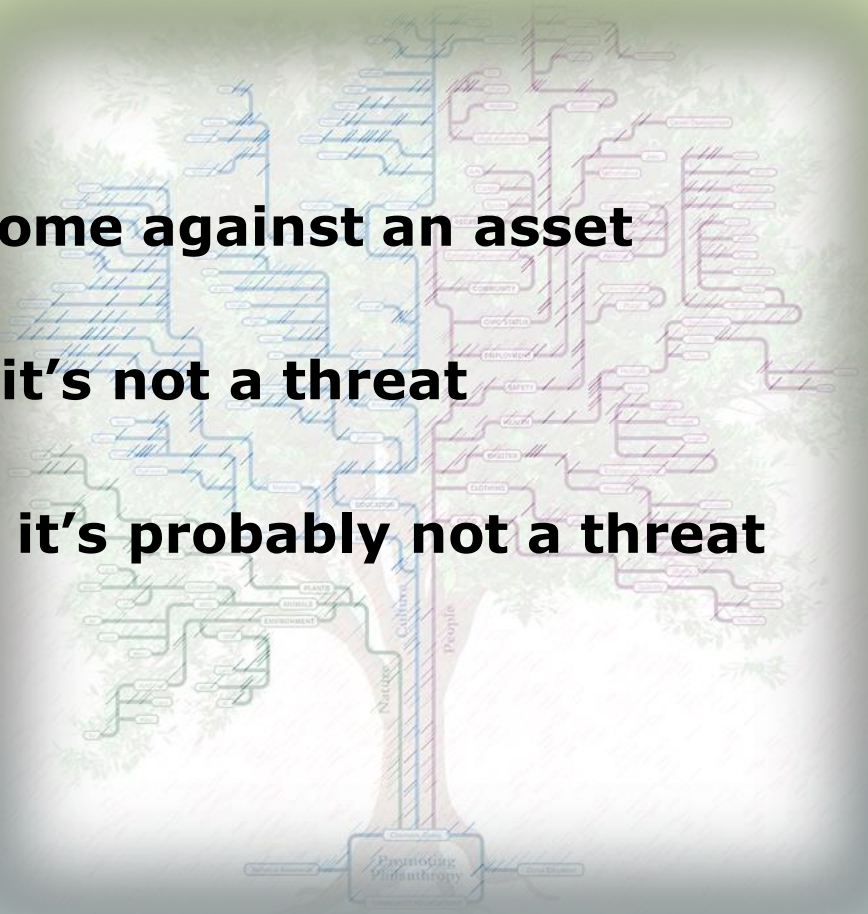
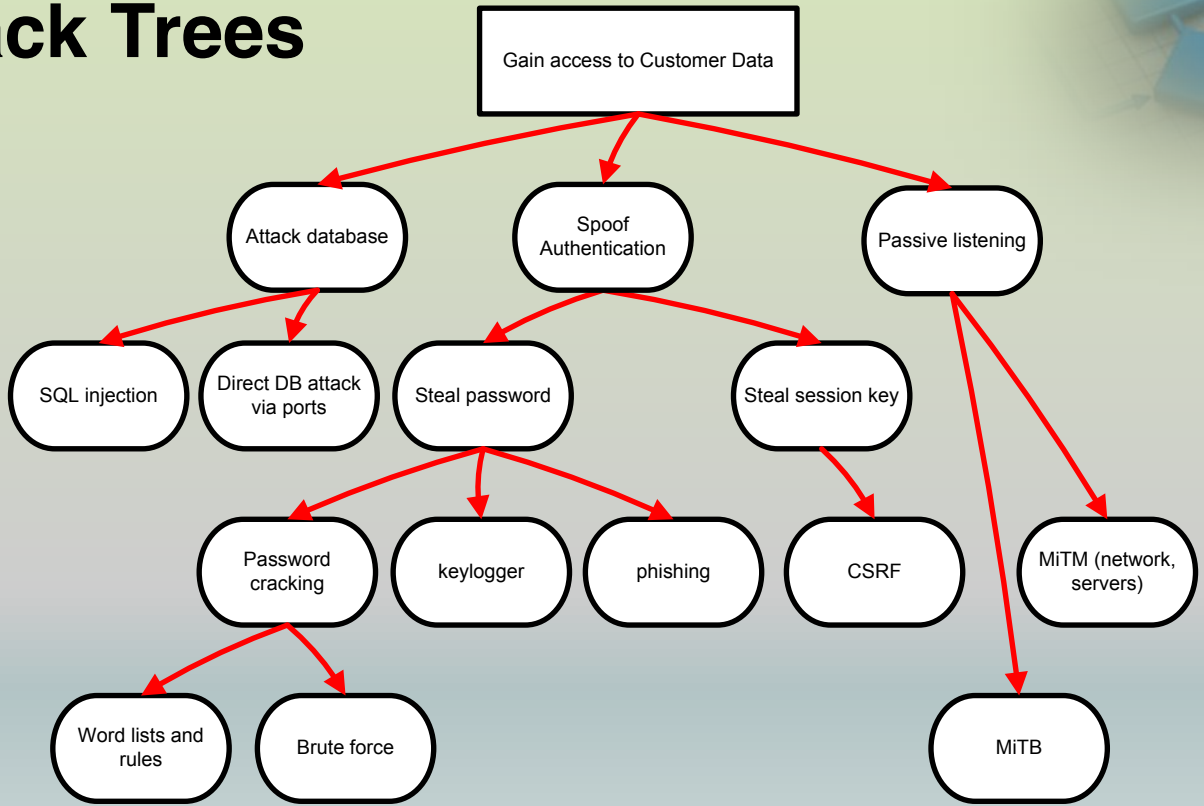
Terminology - Threat

- An undesired action or **outcome against an asset**
- If it doesn't affect an asset, **it's not a threat**
- If it doesn't cost resources… **it's probably not a threat**
- *Example threats:*
  - *Loss of credit card numbers*
  - *Loss of network connectivity*
  - *Inability to log in to the system*

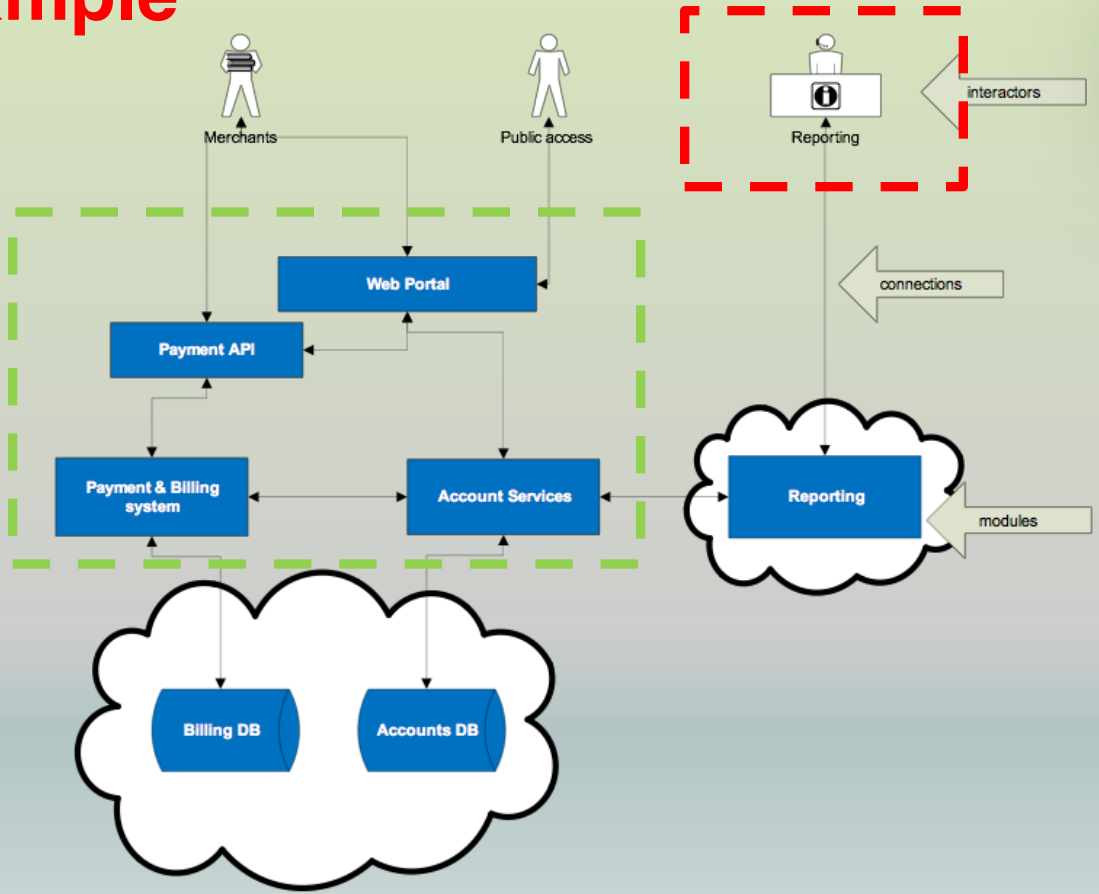


The Tabular System & Attack Trees



Data Flow Diagramming

Example



System Modelling & Threat Analysis
STRIDE Threats

Component (STRIDE)	Spoofing	Tampering	Repudiation	Inf. Discl.	Denial/Svc.	Elev. Priv.
connections		■		■		
Merchants interactor	■					
Public access interactor	■					
Reporting interactor	■					
Web portal module		■	■	■		■
Payment API module		■	■	■		■
Payment & Billing system		■	■	■		■
Billing DB (cloud)		■		■		■
Account Services module		■	■	■		■
Accounts DB (cloud)		■		■		■
Reporting (cloud)		■	■	■		■

System Modelling & Threat Analysis
Security Frame Threats

Component (Sec. Frame)	Validation	authN	authZ	Config.	Sens. Dat.	Session	Crypto.	Auditing
connections						■		
Merchants interactor		■						
Public access interactor		■						
Reporting interactor		■						
Web portal module	■	■	■		■	■		
Payment API module	■	■	■		■	■		
Payment & Billing system			■			■		
Billing DB (cloud)			■	■				■
Account Services module			■		■		■	
Accounts DB (cloud)			■	■				■
Reporting (cloud)				■				■

Mitigation Analysis & Verification
Lab Exercise – Apply Mitigations!

- 20 minutes - Each group associates mitigations to the Threats
  - Use the DFD that was created in the previous exercise
  - Apply a mitigation to each STRIDE or Security Frame element that is annotated to a DFD element
  - eg. An annotated dataflow (TAMPERING or Data Validation) gets a *Validation* mitigation

- 5 minutes – Review
  - Presenter will show a solution on the next slide. Use this time for discussion about the solution

Mitigation Analysis & Verification
Lab – Example Mitigations

Mitigations		
authN	=	strong authentication (SAML, OpenID connect)
authZ	=	strong authorization (XACML, OpenID connect)
auditing	=	centralized auditing and logging module
Encryption	=	asset-level or transport-level (transport is easier)
SQL params	=	parameterize queries (prepared statements, stored procs)
TLS	=	specifically transport-level encryption (do Not use SSL)
Validation	=	Centralized validation module (Constrain, Reject, Sanitize)