Code Review Training Course. Part 1: SDLC and Secure Design

For VIB only

namhabach@gmail.com

#whoami > /dev/null

Nam Ha Bach
A.K.A kendyhikaru
Web Security Pentester/Researcher
namhabach@gmail.com
+84973706272

Theory of Everything (inc Hacking)

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

root>hack the pentagon_
```

Reminder

- Pentest/Audit/VA
- Black Box/White Box/Code Review
- OWASP/PETES/ISSAFF
- ….

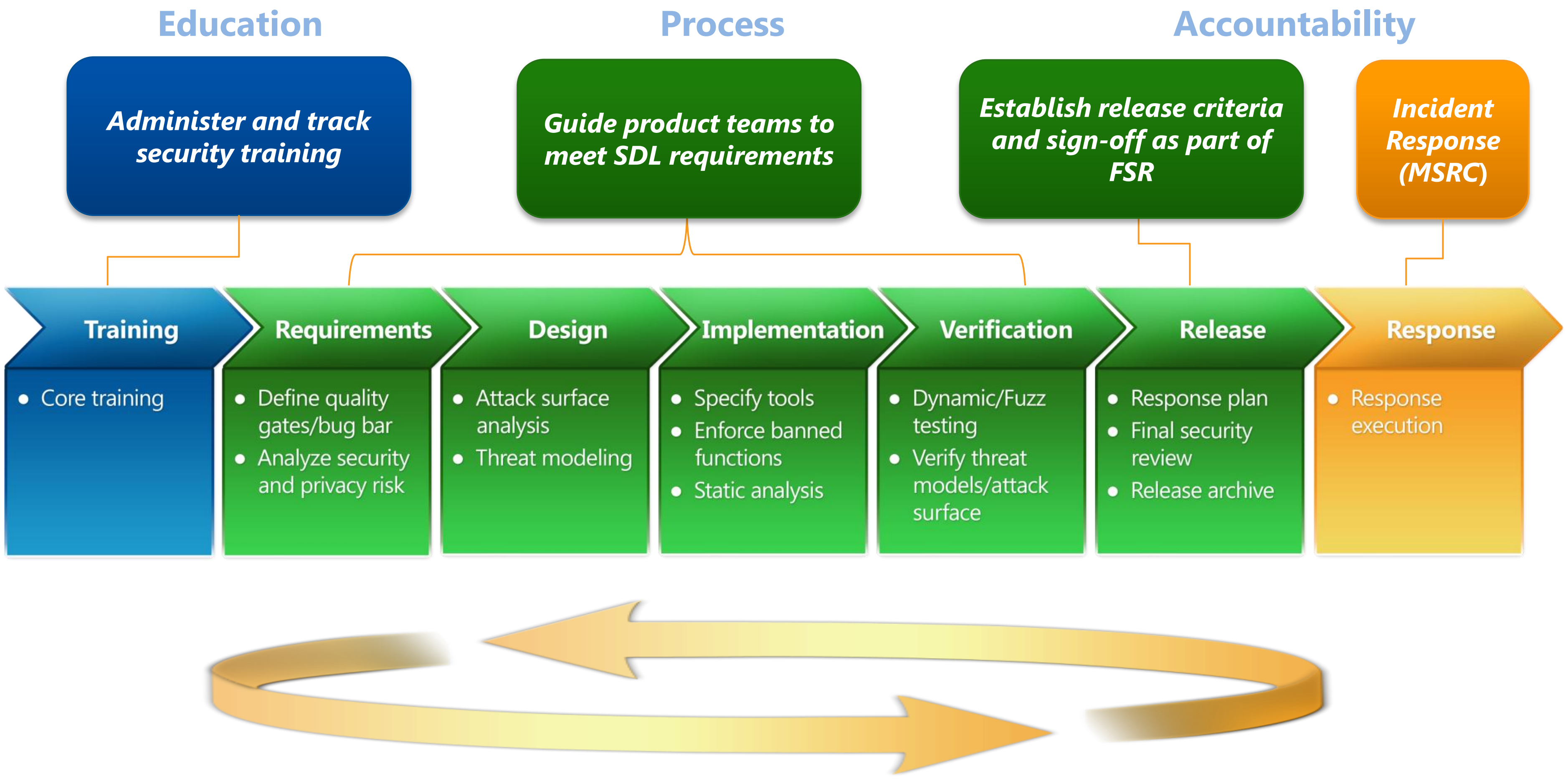
Motivation

- Code Review Only ???
- NO NO NO !!!
- We need:
  - Secure Design ?
  - Application Audit ?
  - Coding & Secure Coding?

About Course

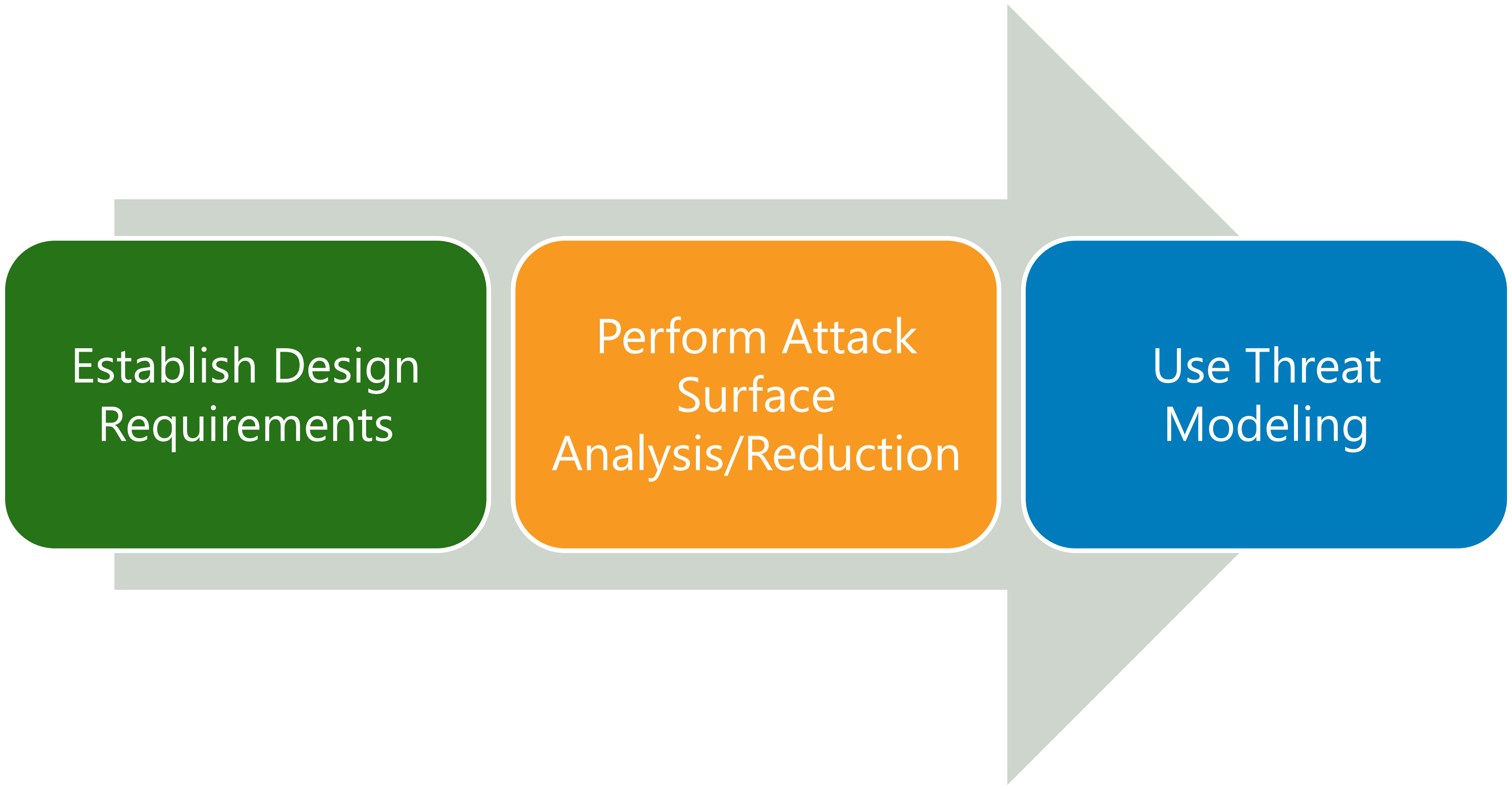
1. SDLC and Secure Design
2. Common Web Attack & Secure Coding
3. Code Review with Programming Language
4. Code Review with Web Framework, Cryptography
5. Web Services, Mobile Application
6. Seminar: Taint Tracking, CTF writeup

Secure Development Life Cycle

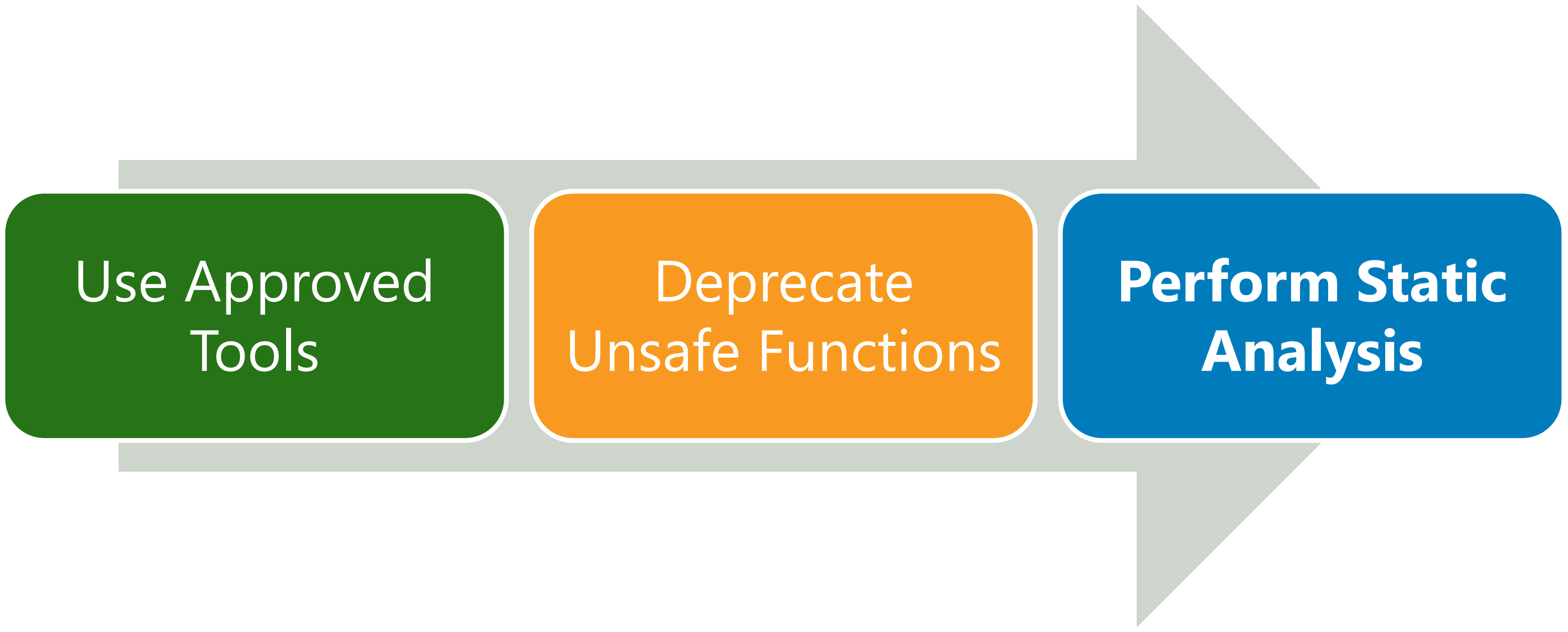

Education
Process
Accountability
Administer and track security training
Guide product teams to meet SDL requirements
Establish release criteria and sign-off as part of FSR
Incident Response (MSRC)
Training
• Core training
Requirements
• Define quality gates/bug bar
• Analyze security and privacy risk
Design
• Attack surface analysis
• Threat modeling
Implementation
• Specify tools
• Enforce banned functions
• Static analysis
Verification
• Dynamic/Fuzz testing
• Verify threat models/attack surface
Release
• Response plan
• Final security review
• Release archive
Response
• Response execution

Ongoing Process Improvements

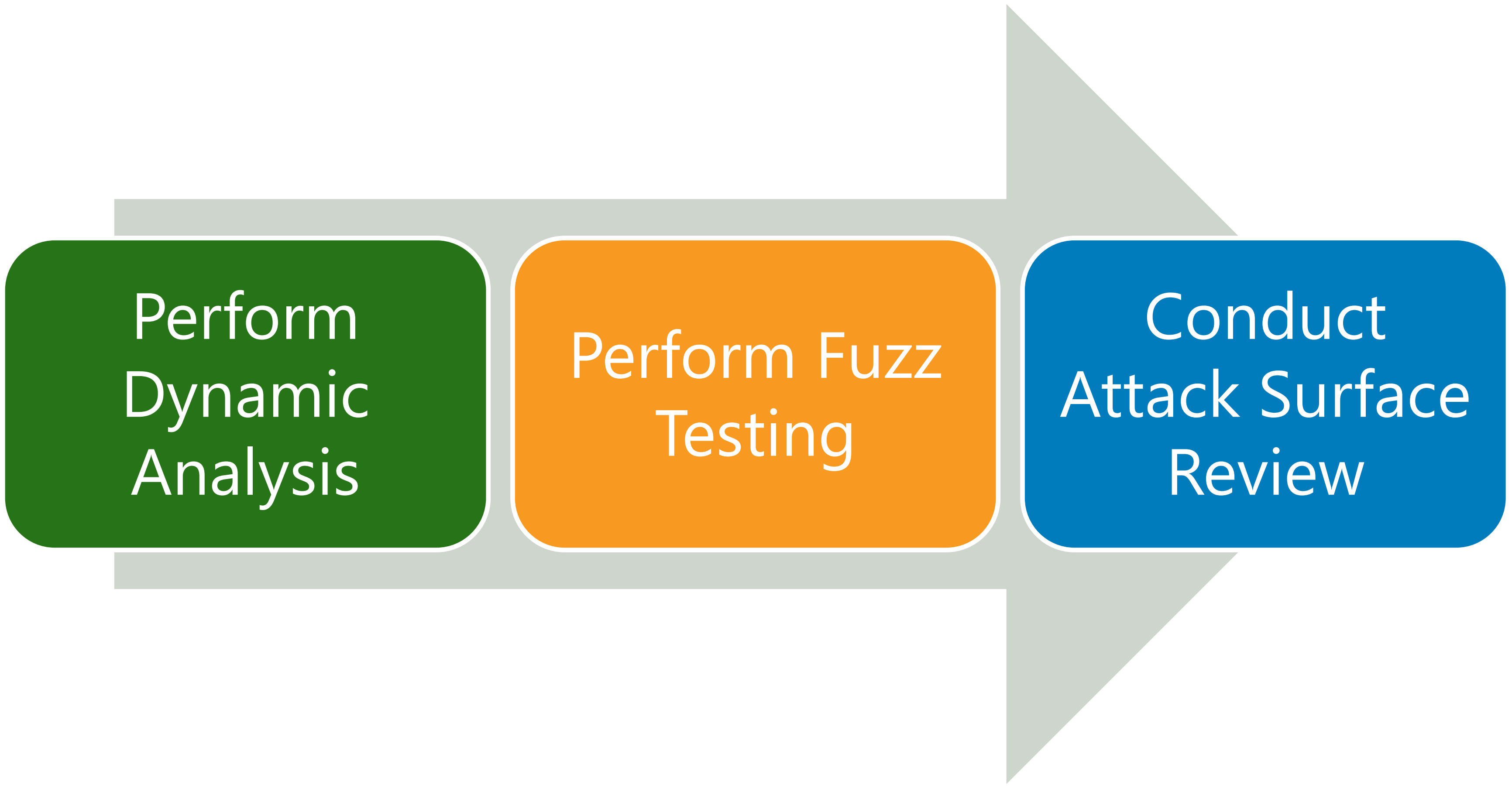
Design


Establish Design Requirements
Perform Attack Surface Analysis/Reduction
Use Threat Modeling

Implementation



Verification



Secure Design

- Attack Surface Reduction (ASR)
- Threat Modeling

The Attack Surface Reduction Process

- Look at all of your entry points
  - Network I/O
  - File I/O
- Rank them
  - Authenticated versus anonymous
  - Administrator only versus user
  - Network versus local
  - UDP versus TCP

Watch Out for Fanout!

File formats

- For example, JPG, MSH, or GIF

Subprotocols

- SSL2, SSL3, TLS, PCT

Verbs

- HTTP
  - Classic
    GET, POST, HEAD, DELETE
  - WebDAV
    PROPPATCH, PROPFIND, MOVE, LOCK
- SMTP
  - HELO, EHLO, MAIL, RCPT
- Queries
  - Extended sprocs and sprocs

It’s Not *Just* About Turning Stuff Off!

Higher Attack Surface	**Lower Attack Surface**
Executing by default	Off by default
Open socket	Closed socket
UDP	TCP
Anonymous access	Authenticated access
Constantly on	Intermittently on
Admin access	User access
Internet access	Local subnet access
SYSTEM	Not SYSTEM!
Uniform defaults	User-chosen settings
Large code	Small code
Weak ACLs	Strong ACLs

ASR Examples

Windows

- Authenticated RPC
- Firewall on by default

SQL Server 2005

- xp_cmdshell off by default
- CLR and COM off by default
- Network service

Internet Information Services version 6 (IIS6)

- Off by default
- Network service by default
- Static files by default

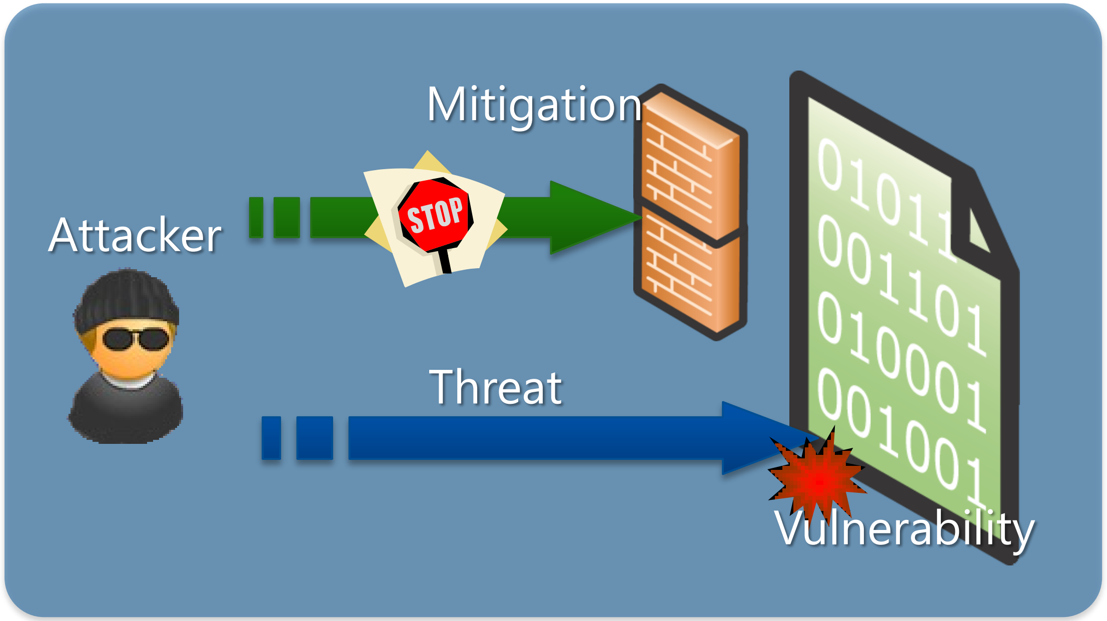
Visual Studio® 2005

- Web server localhost only
- SQL Server Express localhost only

Attack Surface Reduction
is as important as trying
to get the code right

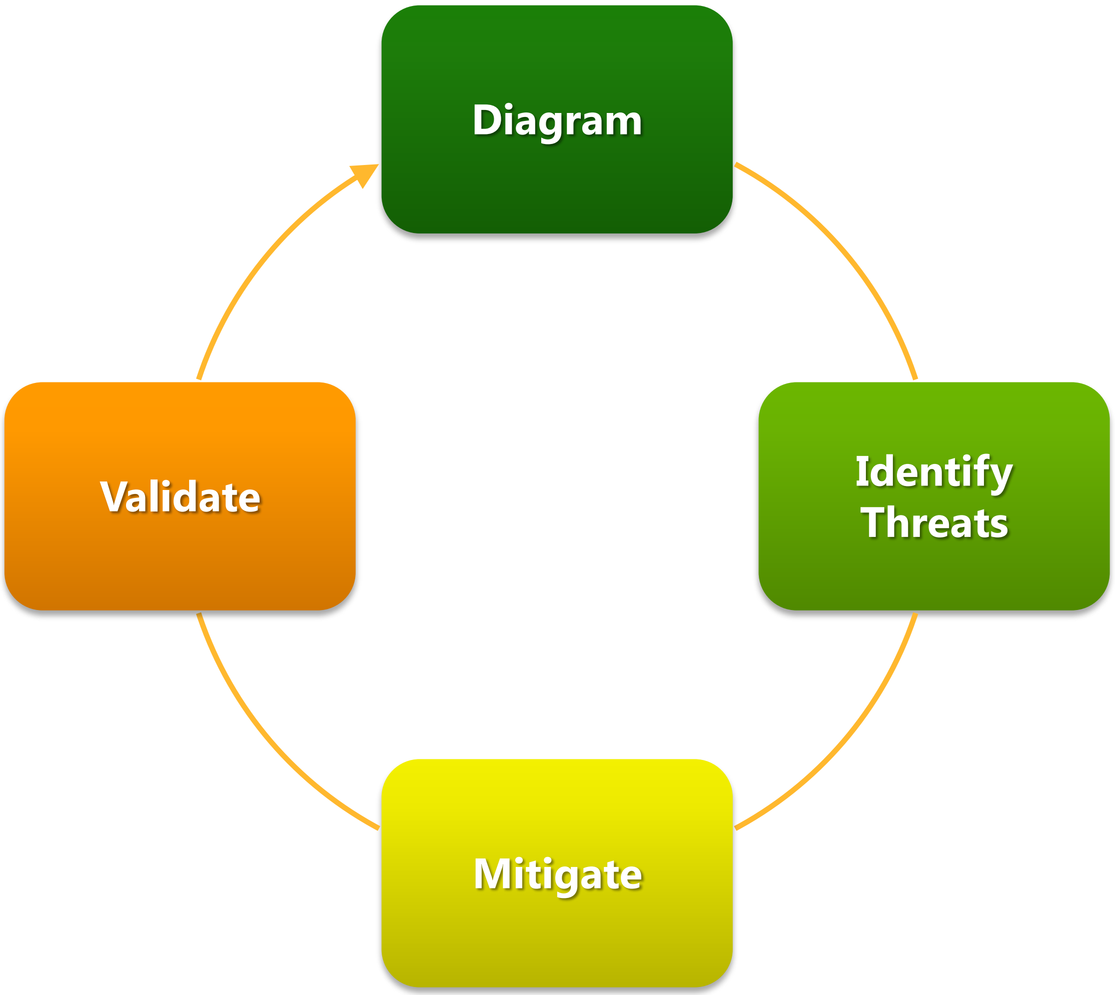
Threat Analysis

- Secure software starts with understanding the threats
- Threats are not vulnerabilities
- Threats live forever; they are the attacker's goal



How to Threat Model

The Process in a Nutshell



Diagramming

- Use DFDs (Data Flow Diagrams)
  - Include processes, data stores, data flows
  - Include *trust boundaries*
  - Diagrams per scenario may be helpful
- Update diagrams as product changes
- Enumerate assumptions, dependencies
- Number everything (if manual)

Diagram Elements: Examples

External Entity

- People
- Other systems
- Microsoft.com

Process

- DLLs
- EXEs
- COM object
- Components
- Services
- Web Services
- Assemblies

Data Flow

- Function call
- Network traffic
- Remote Procedure Call (RPC)

Data Store

- Database
- File
- Registry
- Shared Memory
- Queue / Stack

Trust Boundary

- Process boundary
- File system

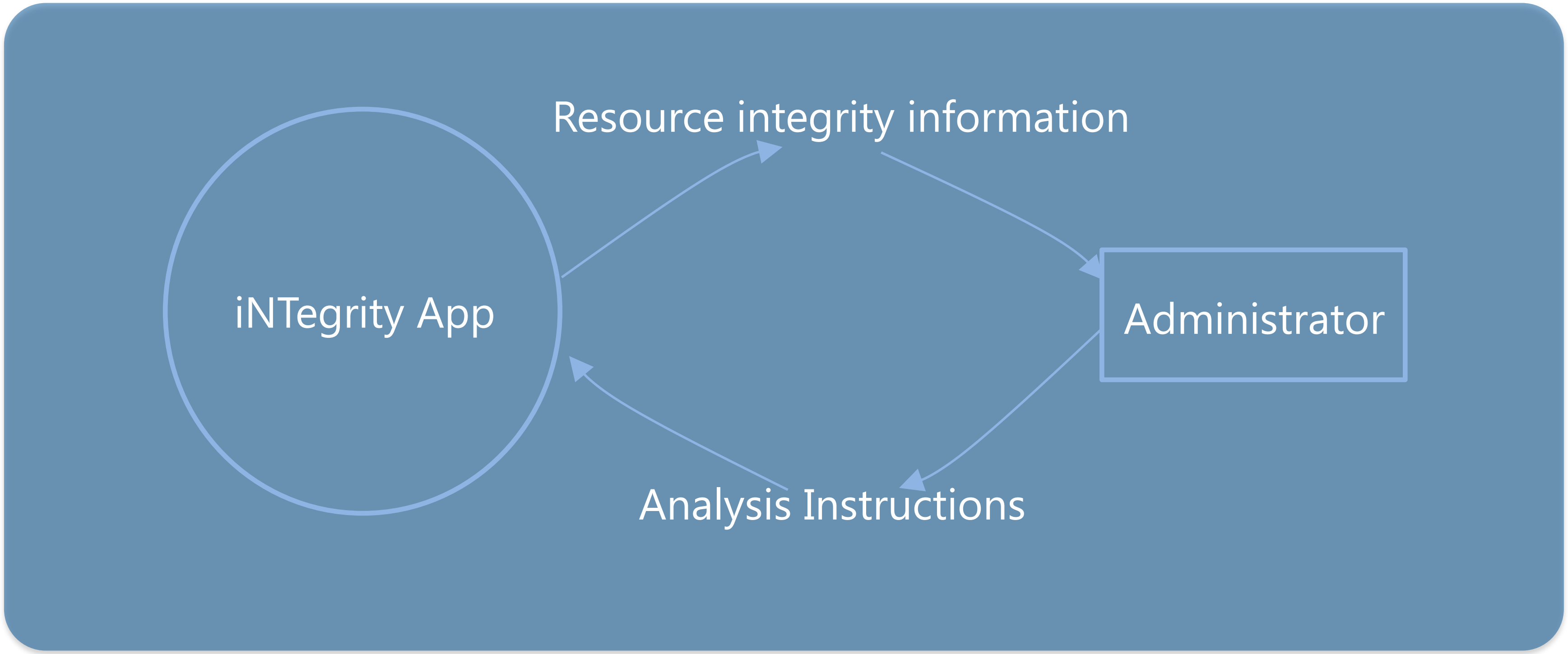
Diagrams: Trust Boundaries

- Add trust boundaries that intersect data flows
- Points/surfaces where an attacker can interject
  - Machine boundaries, privilege boundaries, integrity boundaries are examples of trust boundaries
  - Threads in a native process are often inside a trust boundary, because they share the same privs, rights, identifiers and access
- Processes talking across a network always have a trust boundary
  - They make may create a secure channel, but they're still distinct entities
  - Encrypting network traffic is an 'instinctive' mitigation
    - But doesn't address tampering or spoofing

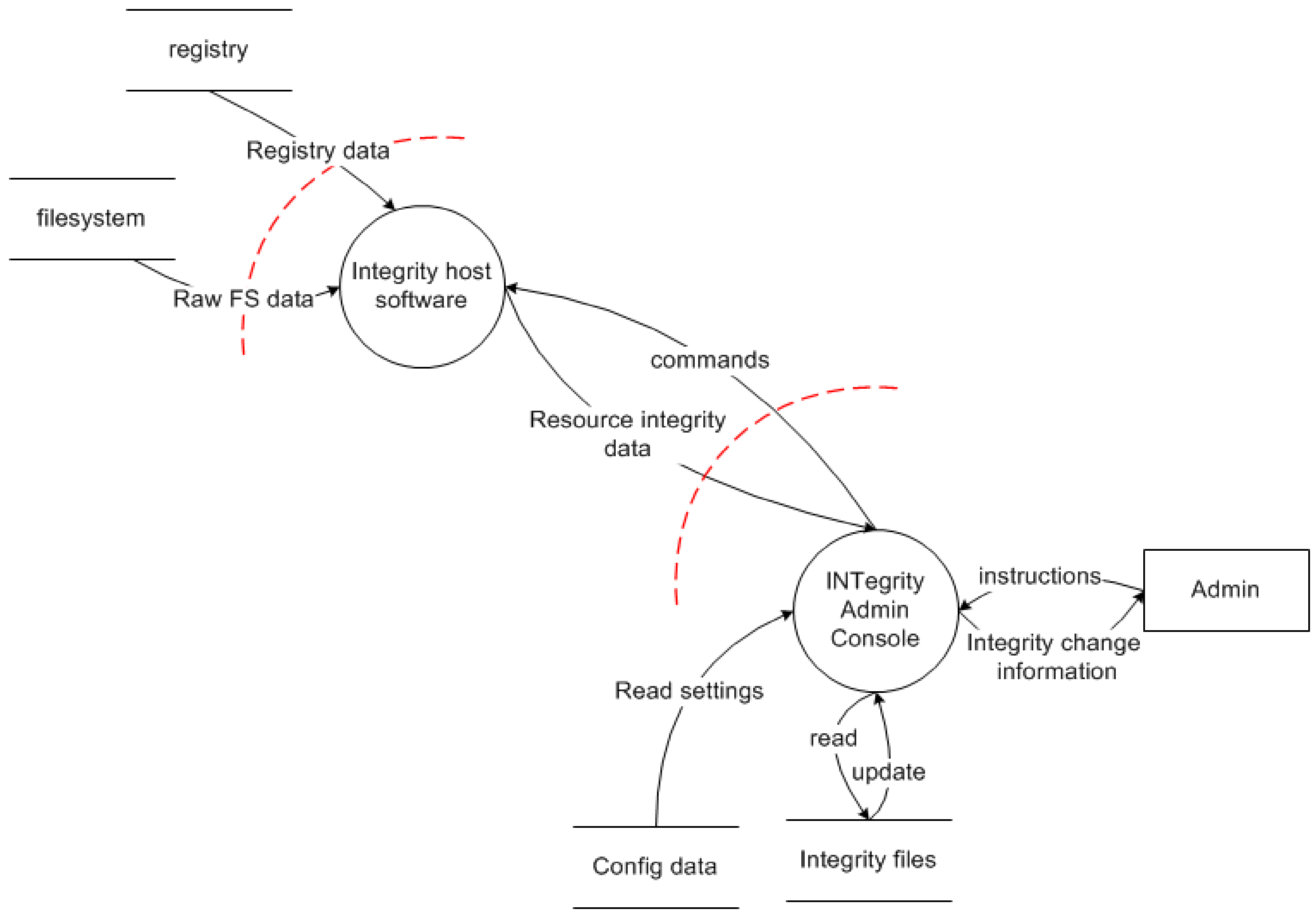
Diagram layers

- Context Diagram
  - Very high-level; entire component / product / system
- Level 1 Diagram
  - High level; single feature / scenario
- Level 2 Diagram
  - Low level; detailed sub-components of features
- Level 3 Diagram
  - More detailed
  - Rare to need more layers, except in huge projects or when you’re drawing more trust boundaries

Context Diagram

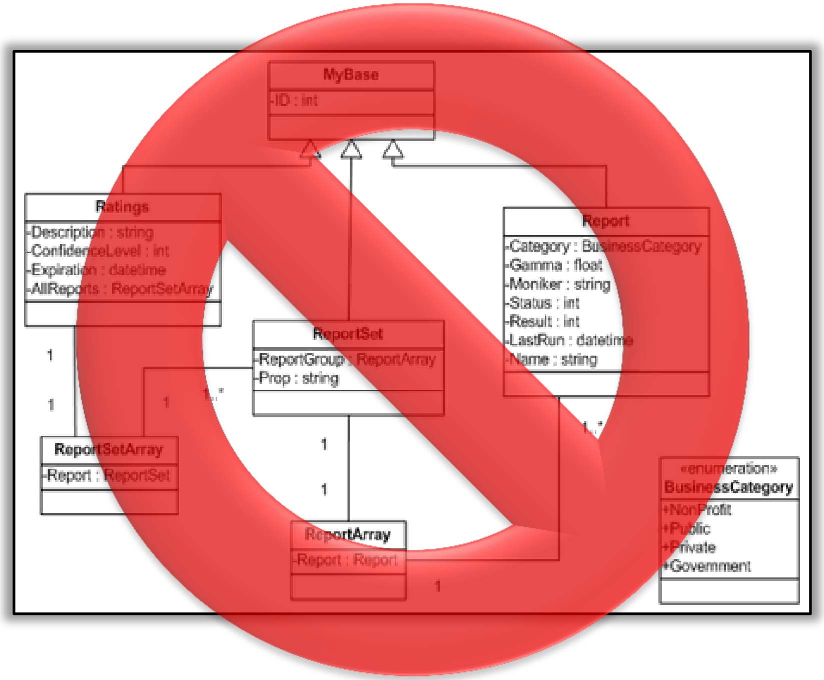


Level 1 Diagram



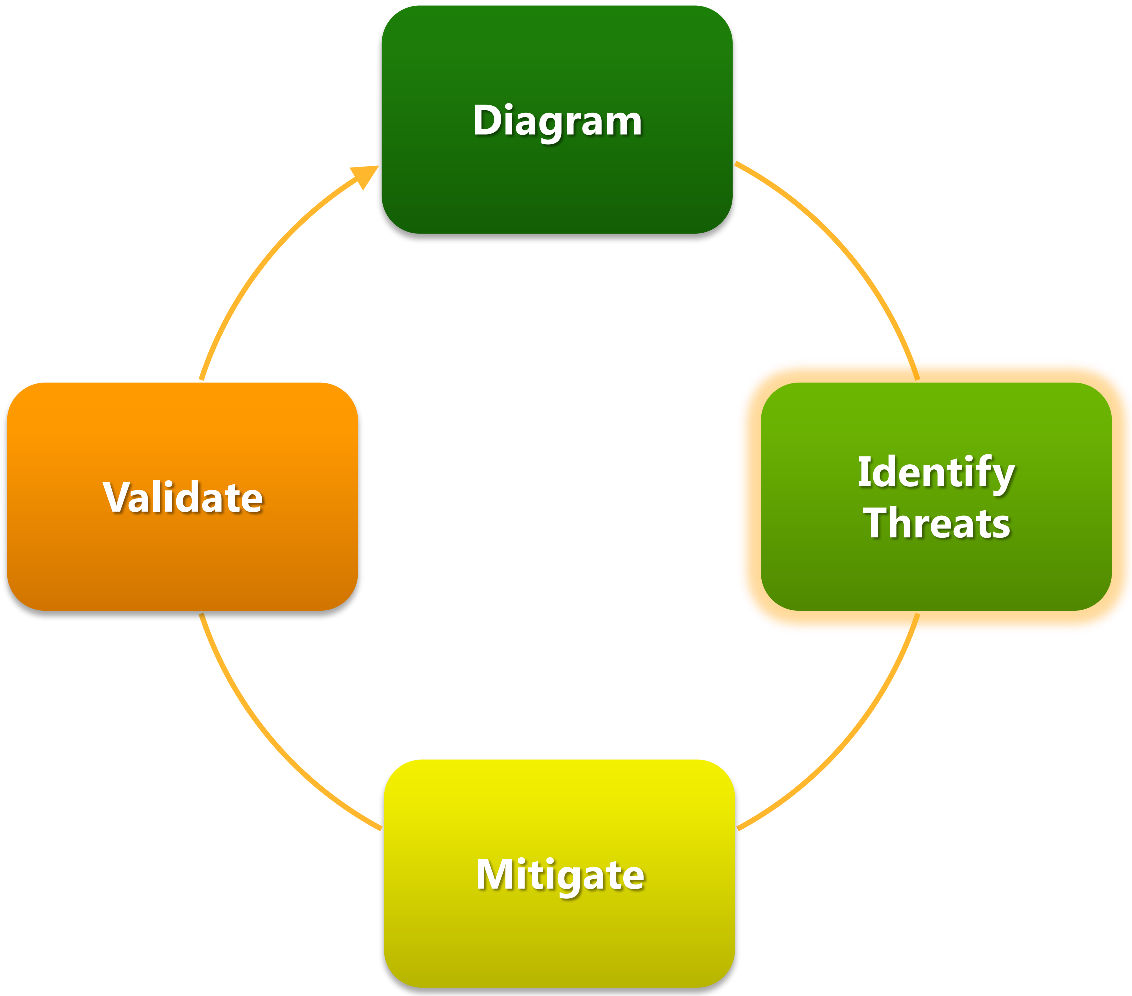
Diagrams Should Not Resemble

- Flow charts
- Class diagrams
- Call graphs



SQL Server Security Flowchart

The Process: Identifying Threats



Identify Threats

- Experts can brainstorm
- How to do this without being an expert?
  - Use STRIDE to step through the diagram elements
  - Get specific about threat manifestation

Threat	Property we want
Spoofing	Authentication
Tampering	Integrity
Repudiation	Nonrepudiation
Information Disclosure	Confidentiality
Denial of Service	Availability
Elevation of Privilege	Authorization

Understanding the STRIDE Threats

Threat	Property	Definition	Example
Spoofing	Authentication	Impersonating something or someone else.	Pretending to be any of billg, microsoft.com or ntdll.dll
Tampering	Integrity	Modifying data or code	Modifying a DLL on disk or DVD, or a packet as it traverses the LAN.
Repudiation	Non-repudiation	Claiming to have not performed an action.	"I didn't send that email," "I didn't modify that file," "I *certainly* didn't visit that web site, dear!"
Information Disclosure	Confidentiality	Exposing information to someone not authorized to see it	Allowing someone to read the Windows source code; publishing a list of customers to a web site.
Denial of Service	Availability	Deny or degrade service to users	Crashing Windows or a web site, sending a packet and absorbing seconds of CPU time, or routing packets into a black hole.
Elevation of Privilege	Authorization	Gain capabilities without proper authorization	Allowing a remote internet user to run commands is the classic example, but going from a limited user to admin is also EoP.

Find Threats: Use STRIDE per Element

- Start with items connected to dangerous data flows (those crossing boundaries)
- Use the chart to help you think of attacks
- Keep a running list

	S	T	R	I	D	E
External Entity	✓		✓			
Process	✓	✓	✓	✓	✓	✓
Data Store		✓	✓	✓	✓	
Data Flow		✓		✓	✓	

Different Threats Affect Each Element Type

ELEMENT	S	T	R	I	D	E
External Entity	✓		✓			
Process	✓	✓	✓	✓	✓	✓
Data Store		✓	?	✓	✓	
Data Flow		✓		✓	✓	

Apply STRIDE Threats to Each Element

- For each item on the diagram:
  - Apply relevant parts of STRIDE
  - Process: STRIDE
  - Data store, data flow: TID
    - Data stores that are logs: TID+R
  - External entity: SR
  - Data flow inside a process:
    - Don’t worry about T, I, or D
- This is why you number things

Use the Trust boundaries

- Trusted/ high code reading from untrusted/low
  - Validate everything for specific and defined uses
- High code writing to low
  - Make sure your errors don’t give away too much

DFD Elements Are Threat Targets: A “Work List”

Data Flow	S	T	R	I	D	E
1→5		✓		✓	✓	
5→6		✓		✓	✓	
6→7		✓		✓	✓	
7→8		✓		✓	✓	
Data Store						
7		✓		✓	✓	
9		✓		✓	✓	
11		✓		✓	✓	
Interactor						
1	✓		✓			
2	✓		✓			
8	✓		✓			
Process						
3	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓	✓
6	✓	✓	✓	✓	✓	✓
10	✓	✓	✓	✓	✓	✓

Each ✓ is a potential threat to the system

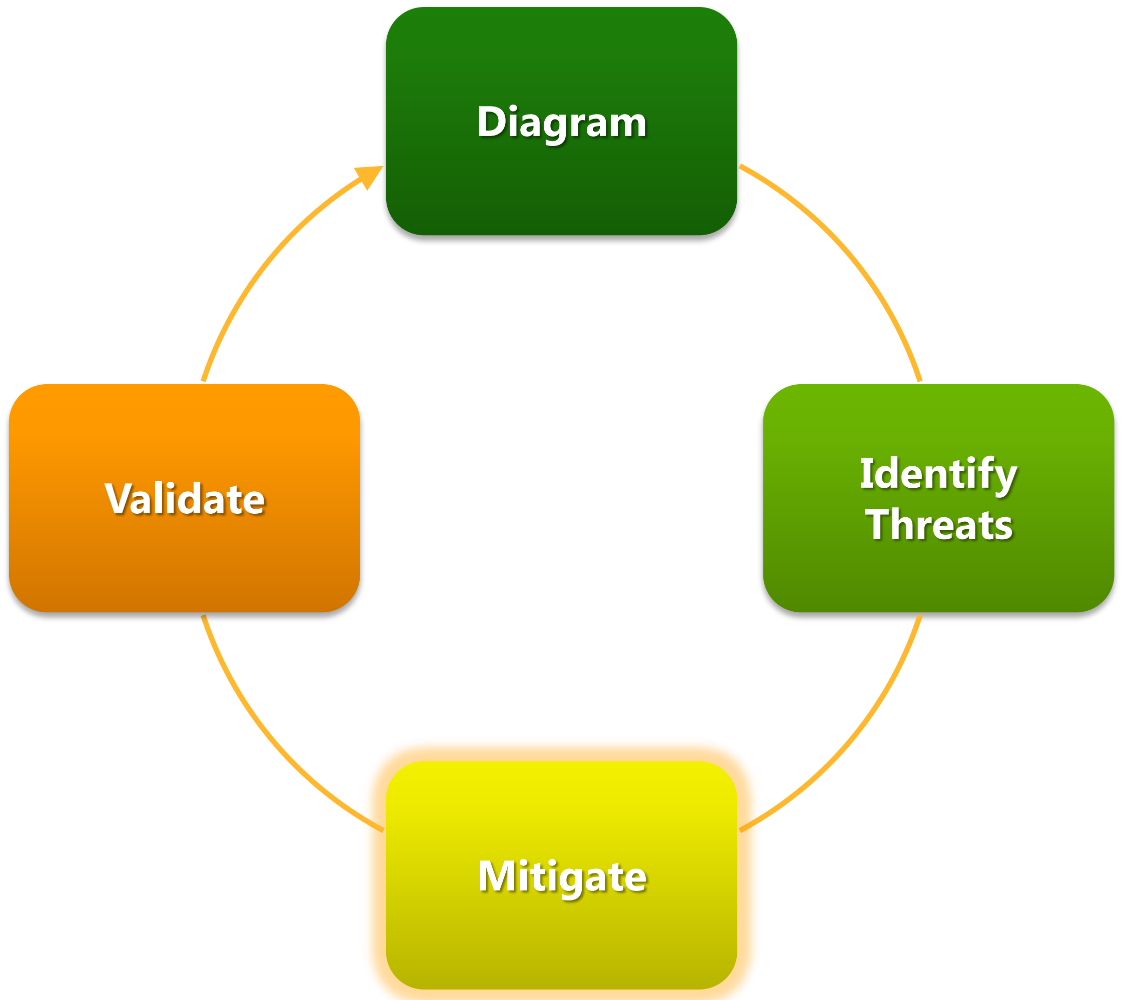
Each threat is governed by the conditions which make the threat possible

A Special Note About Information Disclosure Threats

All information disclosure threats are potential privacy issues

Is the data sensitive or PII?

The Process: Mitigation



Mitigation Is the Point of Threat Modeling

- Mitigation
  - To address or alleviate a problem
- Protect customers
- Design secure software
- Why bother if you:
  - Create a great model
  - Identify lots of threats
  - Stop
- So, find problems and fix them

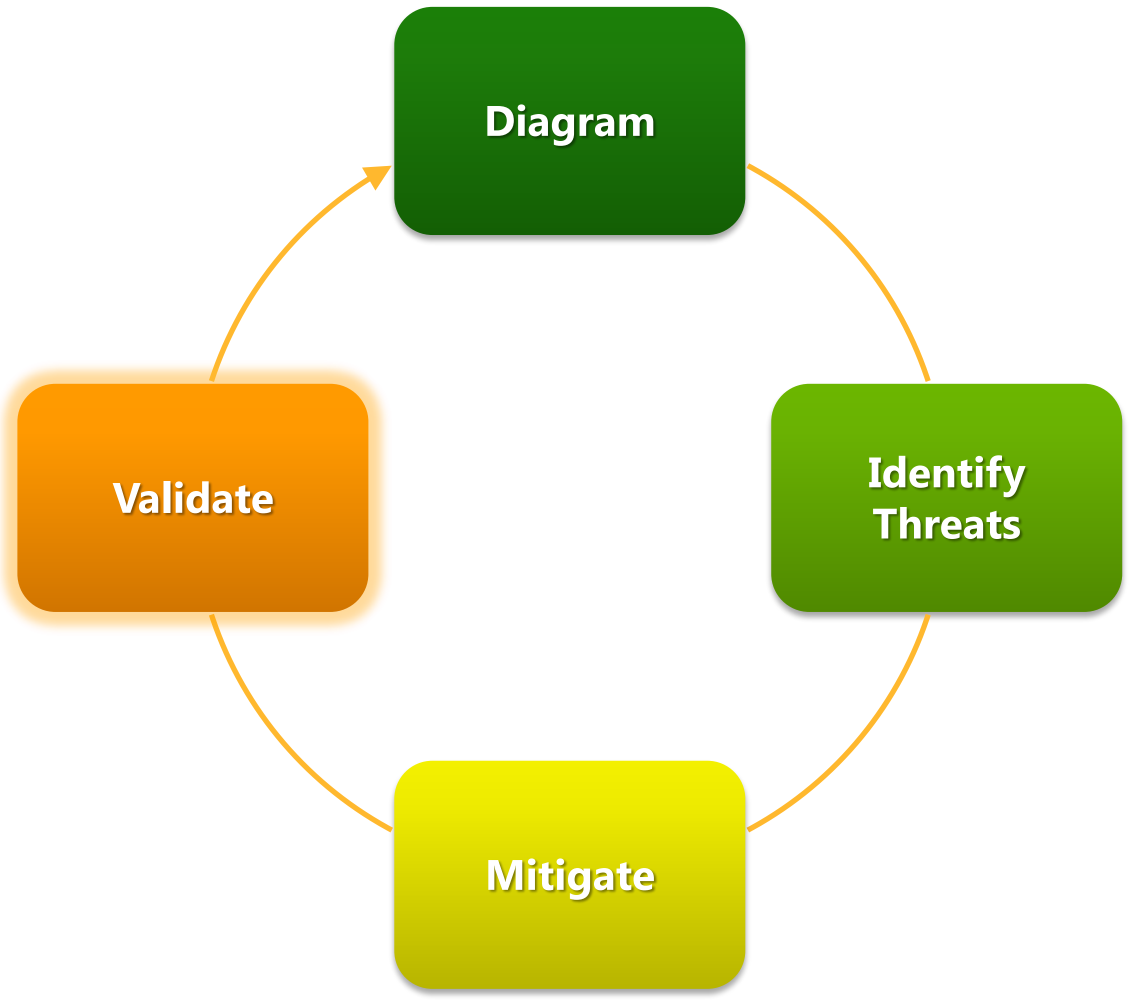
Mitigate

- Address each threat
- Four ways to address threats
  1. Redesign to eliminate
  2. Apply standard mitigations
     - What have similar software packages done and how has that worked out for them?
  3. Invent new mitigations (riskier)
  4. Accept vulnerability in design
     - SDL rules about what you can accept
- Address each threat

Standard Mitigations

Spoofing	Authentication	To authenticate principals: • Cookie authentication • Kerberos authentication • PKI systems such as SSL/TLS and certificates To authenticate code or data: • Digital signatures
Tampering	Integrity	• Windows Vista Mandatory Integrity Controls • ACLs • Digital signatures
Repudiation	Non Repudiation	• Secure logging and auditing • Digital Signatures
Information Disclosure	Confidentiality	• Encryption • ACLS
Denial of Service	Availability	• ACLs • Filtering • Quotas
Elevation of Privilege	Authorization	• ACLs • Group or role membership • Privilege ownership • Input validation

The Process: Validation



Validating Threat Models

- Validate the whole threat model
  - Does diagram match final code?
  - Are threats enumerated?
  - Minimum: STRIDE per element that touches a trust boundary
  - Has Test / QA reviewed the model?
    - Tester approach often finds issues with threat model or details
  - Is each threat mitigated?
  - Are mitigations done right?
- Did you check these before Final Security Review?
  - Shipping will be more predictable

Validate Quality of Threats and Mitigations

- Threats: Do they:
  - Describe the attack
  - Describe the context
  - Describe the impact
- Mitigations
  - Associate with a threat
  - Describe the mitigations
  - File a bug

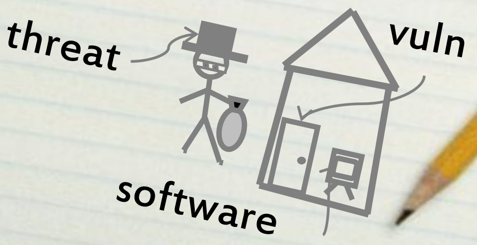
✗ Fuzzing is a test tactic, not a mitigation

Validate Information Captured

- Dependencies
  - What other code are you using?
  - What security functions are in that other code?
  - Are you sure?
- Assumptions
  - Things you note as you build the threat model
    - ✗ "HTTP.sys will protect us against SQL Injection"
    - ✗ "LPC will protect us from malformed messages"
    - ✓ GenRandom will give us crypto-strong randomness

Threat Model Checklist

- ☑ No design is complete without a threat model!
- ☑ Follow anonymous data paths
- ☑ Every threat needs a security test plan
- ☑ Check all information disclosure threats – are they privacy issues?



Questions?

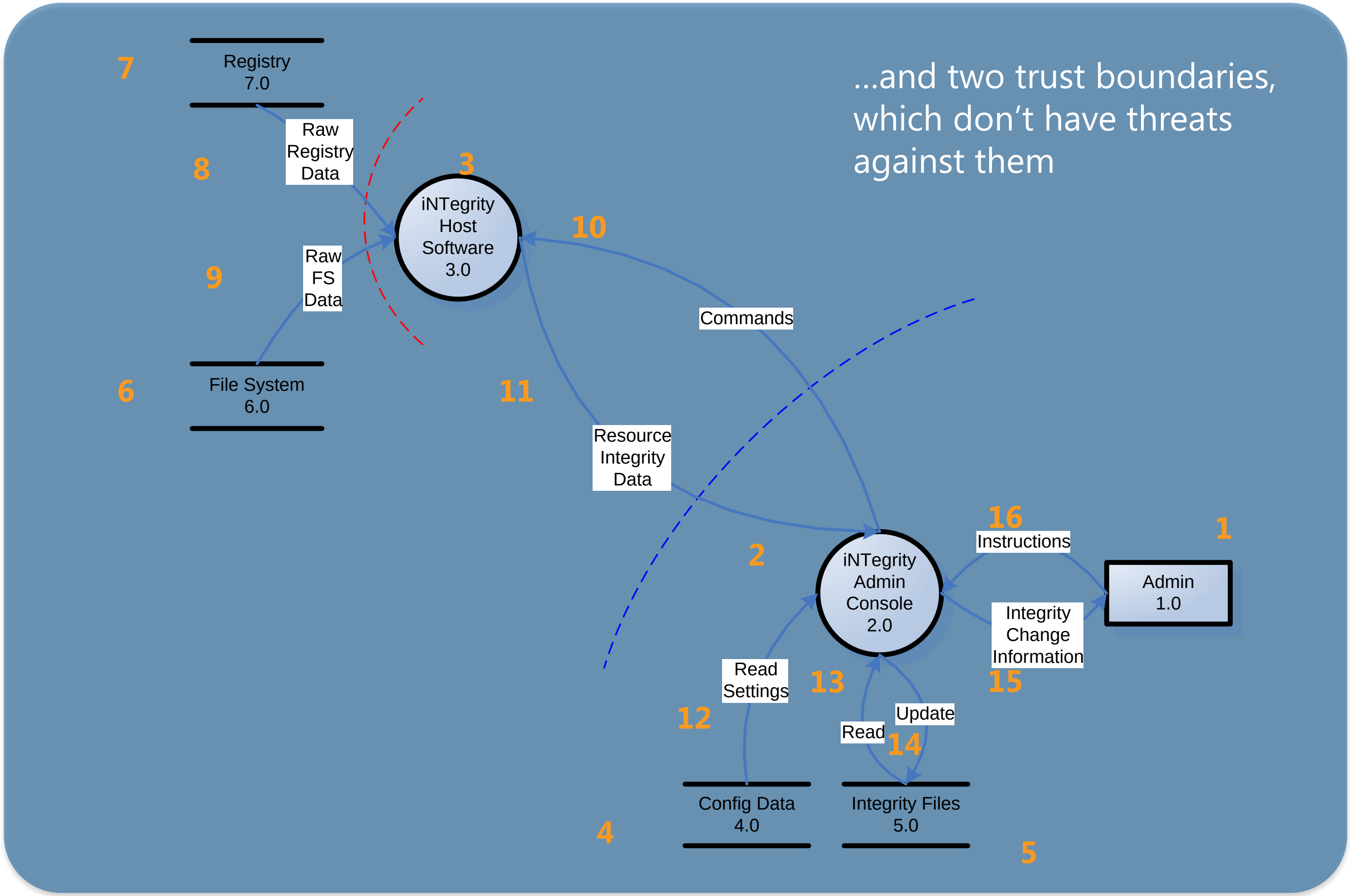
Exercise

Exercise

- Handout
- Work in teams to:
  - Identify all diagram elements
  - Identify threat types to each element
  - Identify at least three threats
  - Identify first order mitigations

Extra credit: Improve the diagram

Identify All Elements (16 Elements)



…and two trust boundaries, which don't have threats against them

Identify Threat Types to Each Element

Identify STRIDE threats by element type

Threats Elements

ELEMENT	S	T	R	I	D	E	
External Entity	✓		✓				Administrator (1)
Process	✓	✓	✓	✓	✓	✓	Admin console (2) , Host SW (3)
Data Store		✓	✓	✓	✓		Config data (4), Integrity data (5), Filesystem data (6), registry (7)
Data Flow		✓		✓	✓		8. raw reg data 9. raw filesystem data 10. commands 16

#Enter to next part_>

End of Part 1