SELF-DEFENDING APPLICATIONS: CYBER DEFENSE AND INTELLIGENCE

J. Drew Miller / Richard MacVarish
U.S. Public Sector, Security and Privacy
Professional Services

Homeland Security

Finding exploits is a test plan.

Using exploits requires an attack plan.

Patching exploits is a reaction.

Preventing exploits requires… **strategy.**

EDS an HP company

“A military man can scarcely pride himself on having 'smitten a sleeping enemy'; it is more a matter of shame, simply, for the one smitten. I would rather you made your appraisal after seeing what the enemy does, since it is certain that, angered and outraged, he will soon launch a determined counterattack.”

– *Isoroku Yamamoto*
to Ogata Taketora on January 9, 1942

Self-Defending Applications

Self-Defending Applications

- What makes an application self-defending?
  - Passive/Active response to the knowledge that an attack is currently underway
- How do you create a self-defending application?
- What do you do with a self-defending application?

Why We Are Here Today

- Everyone is talking about Lifecycle Integration
- Recent supportive automation tools (Static Code Analysis) raising the bar but still not the silver bullet
- HOW to fix the problems by design?
  - So code review finds another exploit before it hits production… why is it there in the first place?
- What we will show you how to do
  - Stop the status quo cycle (not just developers faults)
  - Design it correctly, develop it correctly
  - Critical Infrastructure
    - Metrics generation to determine risk for Critical Infrastructure for applications

The Finish Line

- Defensive shield for application infrastructures
  - NO MORE ADD-ON HARDWARE
- Know when attacks occurs in real-time
  - Including internal personnel (attacks) on the application
- Global Defensive Event Aggregation and Correlation
- Prevent a 21st Century Pearl Harbor
  - Estonia and Russian cyber attacks

The Path

- Custom applications can be asserted to be secure mathematically/logically from vulnerabilities WHILE in operation
  - During development
  - During testing
  - During operation
- Become Native Non-Kinetic Attack Intelligence Gathering Devices
  - Integrate Contextual Properties
- Attacks 'natively' defeated AND monitored
  - SQL Injections, XSS/Forgery, Unauthorized Data Access, etc.

The Gap Analysis

- How many applications know when they are under attack from an authenticated user that bypasses client validation and spoofs a data element?
  - (web browser, or TCP client, etc. any protocol)
- If no, why not?

Innovation's Evolution (Not a Checklist)

- Attack Vectors – Defined by technology, central defense
- Risk – Monitored or mitigated
- Compliance – Asserted during run-time
- Assessment – Application reports events from assessment
- Mitigation – Native mitigation defined by technology
- Security Monitoring – Instant root-cause knowledge
- Health Monitoring – Custom events support RAM/CPU/DISK
- Remediation – Updating technology from one source tree
- Incident Response – Becomes Attack Response
- Availability – Attack Response extends beyond application
- Confidentiality – Enforced by technology
- Integrity – Application self-diagnostics

“Whenever you are asked if you can do a job, tell 'em, 'Certainly, I can!'
Then get busy and find out how to do it.”

– Theodore Roosevelt

Foundation

Enterprise Considerations at Technical Level

- Design/Development work flow
  - Data/System classification
    - U.S. and international regulatory requirements
      - »PCI, FFIEC, SOX, etc.
    - Secret, Top Secret, etc.
  - Design w/ security concepts
    - Validation
    - Authentication
    - Authorization
    - Non-repudiation
    - Compartment classifications
- Operations work flow
  - Change control
  - Audit
  - ATO

EDS an HP company

Step 1: Classification BEFORE Requirements

- System/Data classification and governance
  - What data?
    - PII/PHI
    - Sensitivity (Classified, Secret, etc.)
      - » Military asset locations
      - » Communications from field agents
      - » Infrastructure
        - » Operations, etc.
- The application must know the classification during runtime
  - Context derived logical protection
    - Unless SSN is defined as PII, application cannot determine what to do with the data

EDS an HP company

Step 2: Requirements

- Convergence
  - Government sensitivity requirements
    - U.S., Canada, EU, Germany, UK, etc.
      - » Where is the application portal located?
      - » Where is the application data stored?
      - » Where is the application data processed?
  - Export Compliance/ITAR requirements
    - Cyber defense, non-kinetic warfare, information warfare
  - Corporate policies
  - Functional specification
  - Privacy law

EDS an HP company

Step 3: Knowing You Must Define Roles

- Classification defines requirements
  - Based on requirements, who will be allowed access to specific data?
- Requirements and Best Practice defines Separation of Duties when implemented
  - Operations and SOC
    - Availability, sustain
  - Audit and oversight
    - GSA, PCI Council, etc.
  - Configuration
    - Changes controlled through Change Control Process/Board

Step 4: Roles and Locations Monitoring

- User kept from accessing data that would violate classification or export laws based on data location and user location
  - Must define
    - where the data is stored
    - how the data can be accessed legally
- ITAR example
  - User logs in to system from non-U.S. location?
  - Geo IP (not a perfect solution)
    - Where are users coming from *network-related*
  - Asserting their physical location
- Other controls
  - Passwords for users to enter under duress

“Knowledge of what is does not open the door directly to what should be.”

– Albert Einstein

Design

Step 5: Design

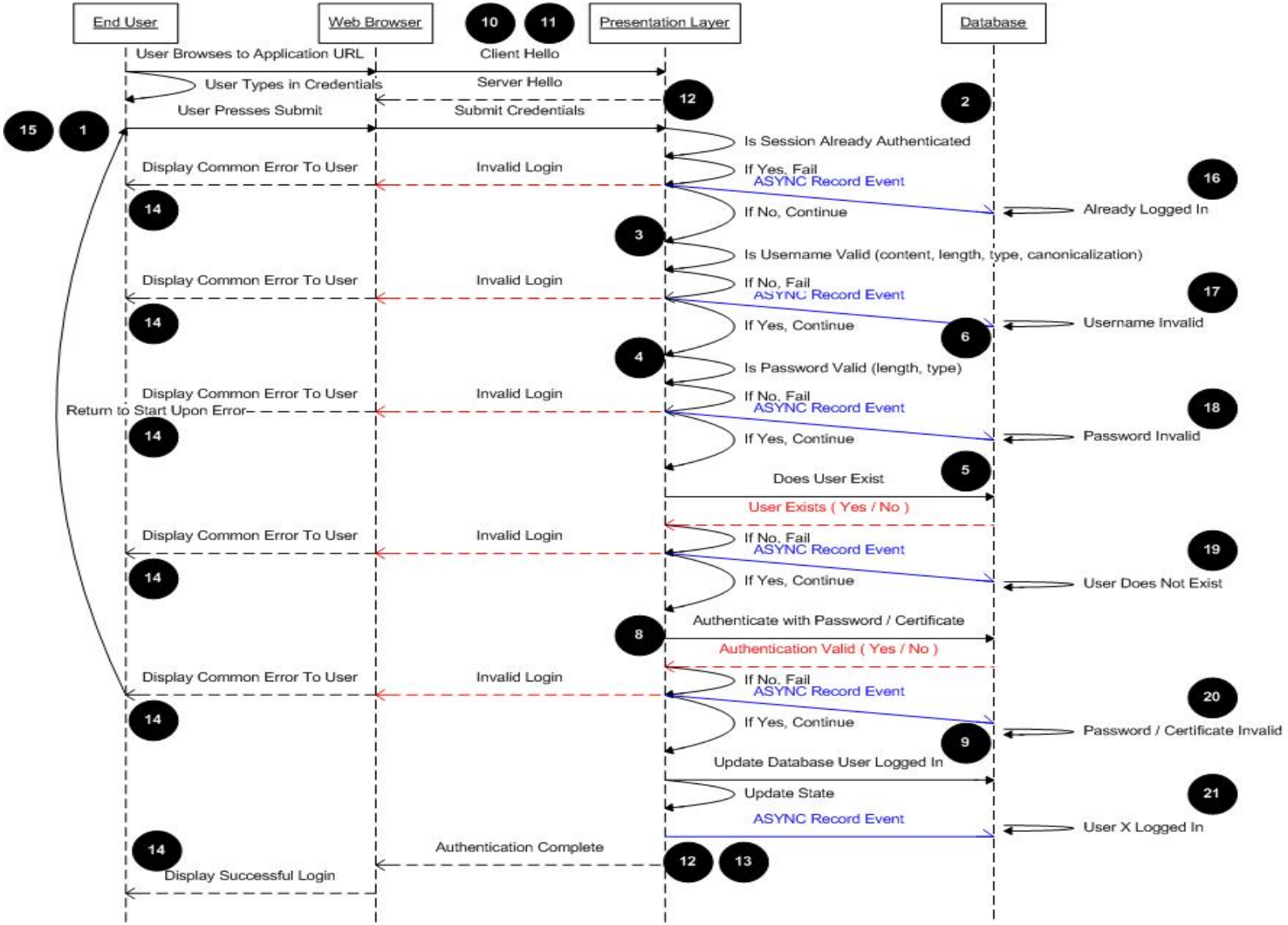
- Centralized
  - Validation, event reporting, error handling, database work flow, encryption, authorization
- No code duplication for base subsystems
- Developers should not be required to be security experts
  - Expecting developers to write secure code has not worked for the industry… why?
    - Sometimes because everyone is an expert?

EDS an HP company

Step 5: Design from Patterns

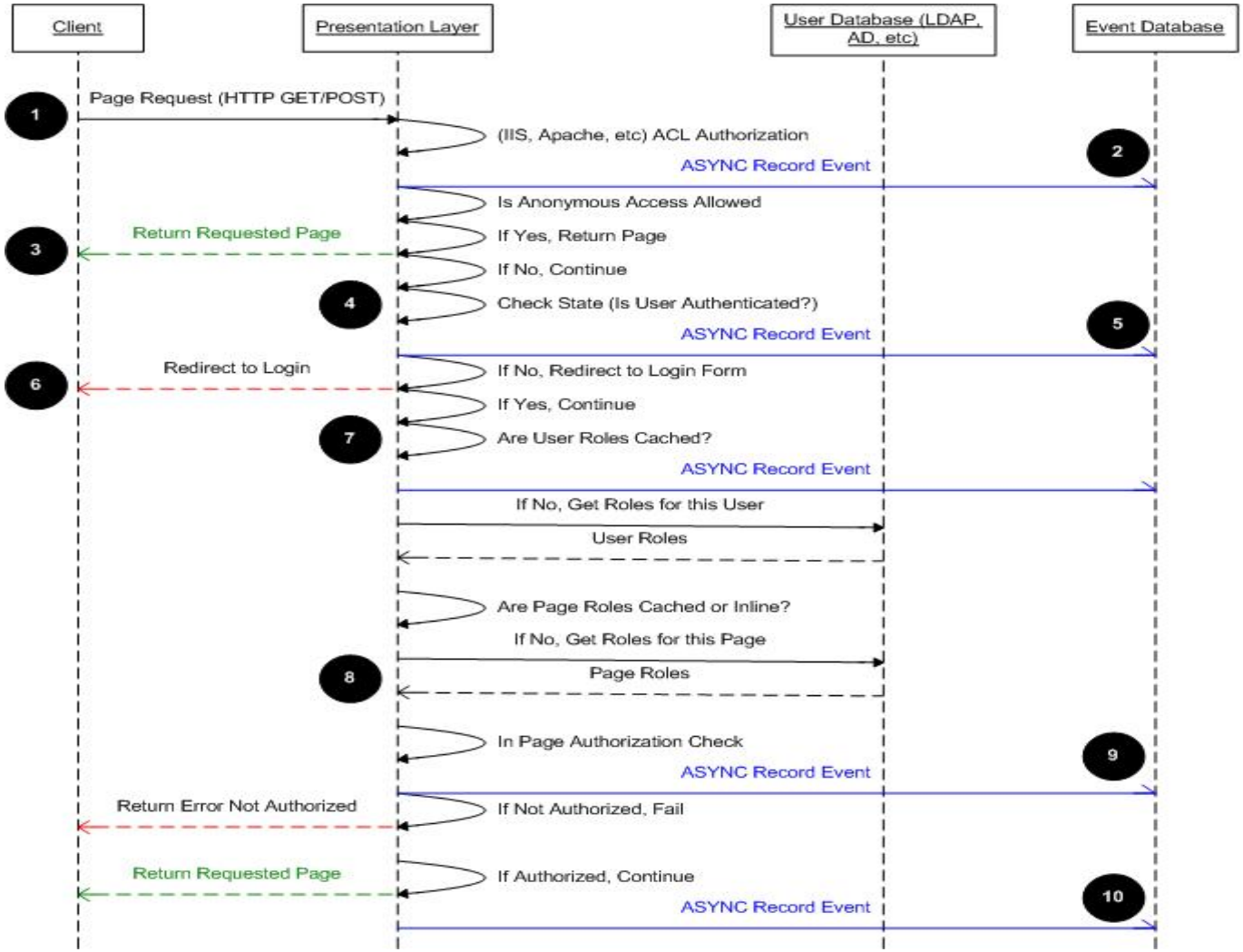
- Designs MUST work for any development language, platform, environment

Authentication

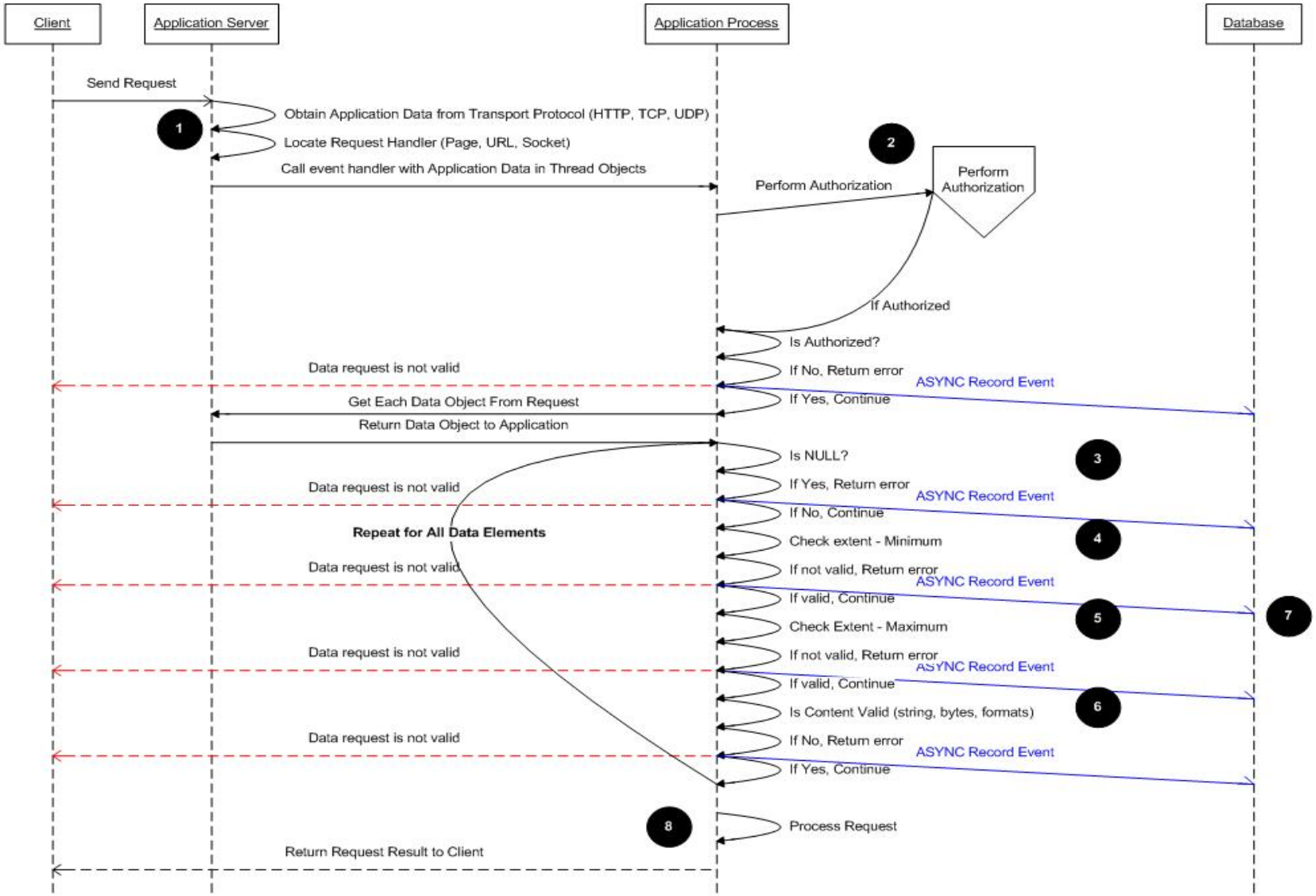


EDS an HP company

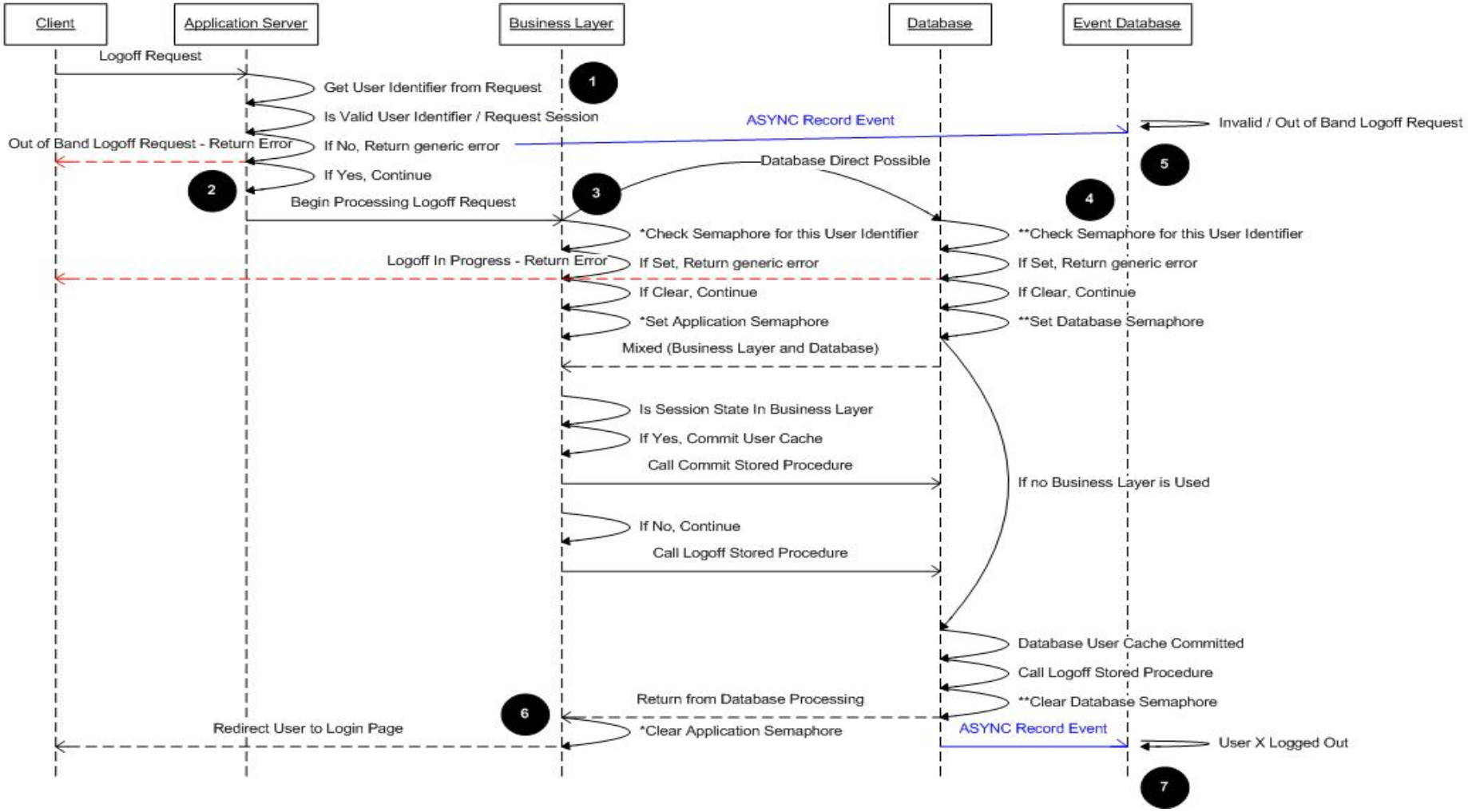
Authorization



Validation


Client
Application Server
Application Process
Database
Send Request
1
Obtain Application Data from Transport Protocol (HTTP, TCP, UDP)
Locate Request Handler (Page, URL, Socket)
Call event handler with Application Data in Thread Objects
2
Perform Authorization
Perform Authorization
If Authorized
Is Authorized?
If No, Return error
ASYNC Record Event
If Yes, Continue
Data request is not valid
Get Each Data Object From Request
Return Data Object to Application
Is NULL?
3
If Yes, Return error
ASYNC Record Event
If No, Continue
Data request is not valid
Repeat for All Data Elements
Check extent - Minimum
4
If not valid, Return error
ASYNC Record Event
If valid, Continue
Data request is not valid
Check Extent - Maximum
5
7
If not valid, Return error
ASYNC Record Event
If valid, Continue
Data request is not valid
Is Content Valid (string, bytes, formats)
6
If No, Return error
ASYNC Record Event
If Yes, Continue
Data request is not valid
8
Process Request
Return Request Result to Client

Termination (Logoff)



Step 6: Contextual Access Control

- SSN
  - Regulatory constraints, database field definition
  - Defined as PII
- Role X
  - Defined to access PII
  - Therefore can see SSN
- If Role X can see PII then Role X can see SSN
  - Stop doing this
    - If(User.IsInRole(X) Then
  - Start doing this
    - GetClassificationOfData(<data element enumeration> e.g. SSN)
    - GetClassificationAccess(Role);
    - If(DataClassification != RoleClassificationAccess)
      - » An entity is attempting to access data they should not be able to even request
- Comparisons done against context
  - NOT determined because one has access through an application to a Table, Form, etc.

EDS an HP company

“All war is deception.”

– Sun Tzu

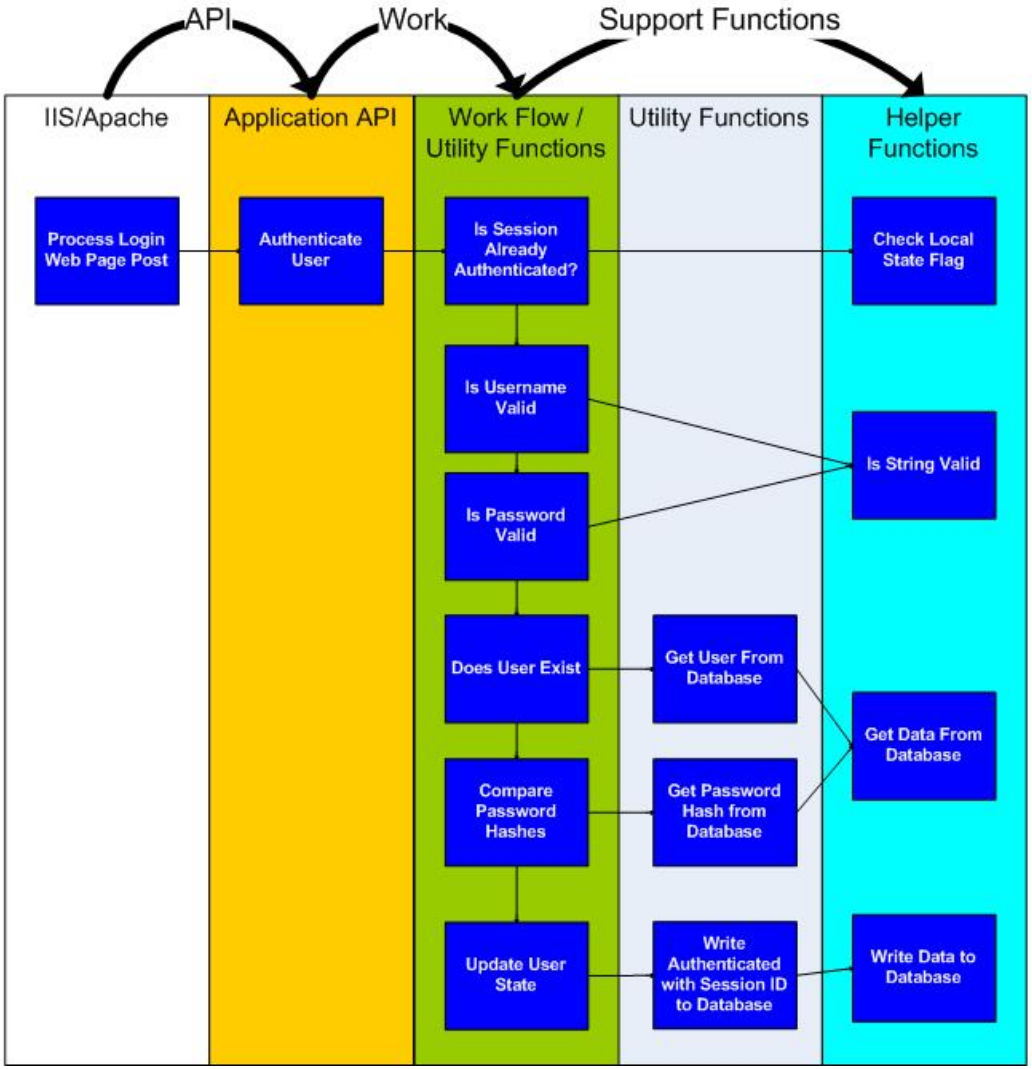
Developing the Foundation

Reducing Costs

- Modular code components
- Centralize what can be centralized
- Validate everything
- Build APIs not Page Handlers

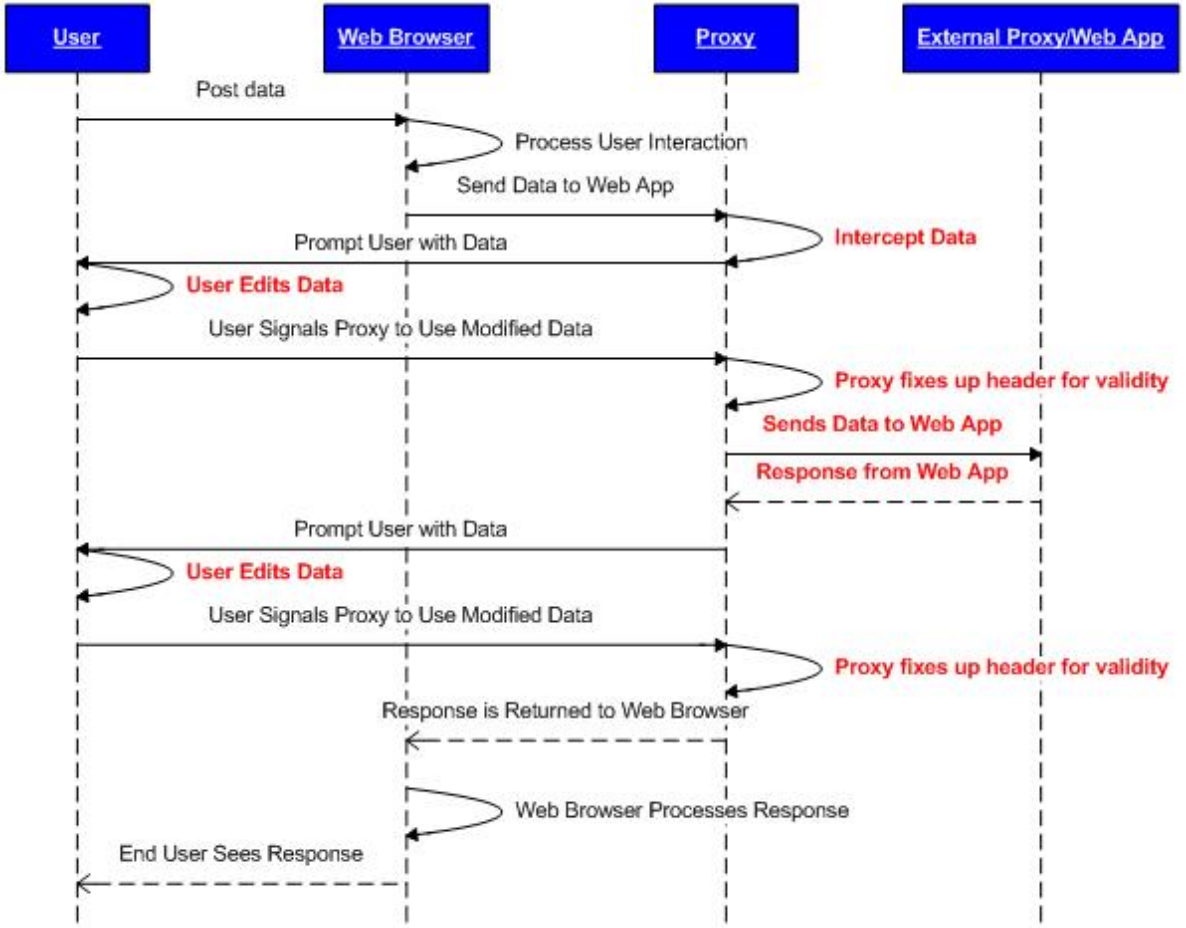
Need for Modularity

- Can the method be used once or many times?
  - Data driven processing



Need for Server Validation

- Centralized validation
  - Validation should include concept of authorization
  - Validation must be performed server-side at each entry point



Introduction to the Undefined Test Case

- How does Application Intrusion Detection work?
  - We test application for two scenarios
    - Success cases
    - Failure cases
  - Any data that causes an undefined output or state is a functional defect
  - Any functional defect *could* in turn be a security defect
- What happens if we find a third undefined test case?
- Return to Design and find where it should be handled and how!

If Only A and B are Possible, What About C?

- Logic of data processing
  - Can data be not valid by design?
    - If yes, invalid data must be handled by code flow
    - If no, invalid data is outside of client/server handling and is therefore an attack
- Doesn't the client validate data too?
  - Yes
- So is the user bypassing the client validation?
  - Yes
- If it is a compatibility error and client code is not synchronized to the server validation system, then it's a functional bug in either server or client validation software

EDS an HP company

Modular Validation Attack Detection

How do we use this contextual information?

```
// is string too big
Debug.Assert( Input.Length <= nMax );
if( Input.Length > nMax )
{
   // BUFFER OVERFLOW ATTACK JUST OCCURRED

   // return failure
   return false;
}
// is string too small
Debug.Assert(Input.Length >= nMin);
if( Input.Length < nMin )
{
   // BUFFER TOO SMALL ATTACK JUST OCCURRED

   // return failure
   return false;
}
// determine what type of content validation to perform
switch( ValidationSet )
{
   // if we are testing for alphanumeric content
   case EnumValidationSet.Alphanumeric:
      // if you compare against != true, or == false,
      // then you can avoid NESTING successful processing
      // only nest FAILURES
      if( IsAlphanumericString( Input ) != true )
      {
         // CANONICALIZATION ATTACK

         // return failure
         return false;
      }
      break;

   <snip>
```

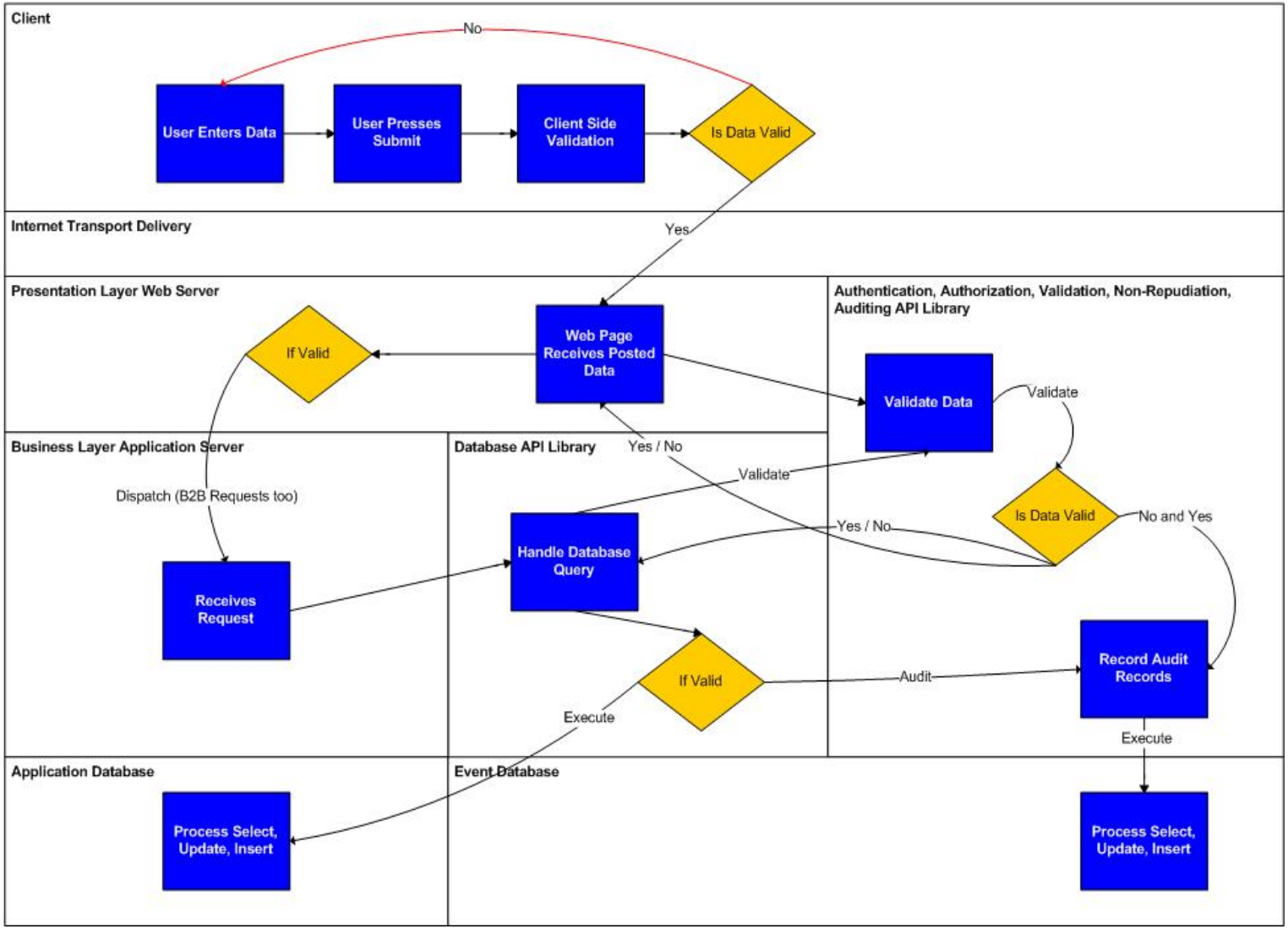

EDS an HP company

Benefits of Centralized Subsystems

- No code duplication!
- No code duplication!
- No code duplication!

- Fix defects for validation in one place and project across the enterprise

Subsystem Placement



“The chief condition on which, life, health and vigor depend on, is action. It is by action that an organism develops its faculties, increases its energy, and attains the fulfillment of its destiny.”

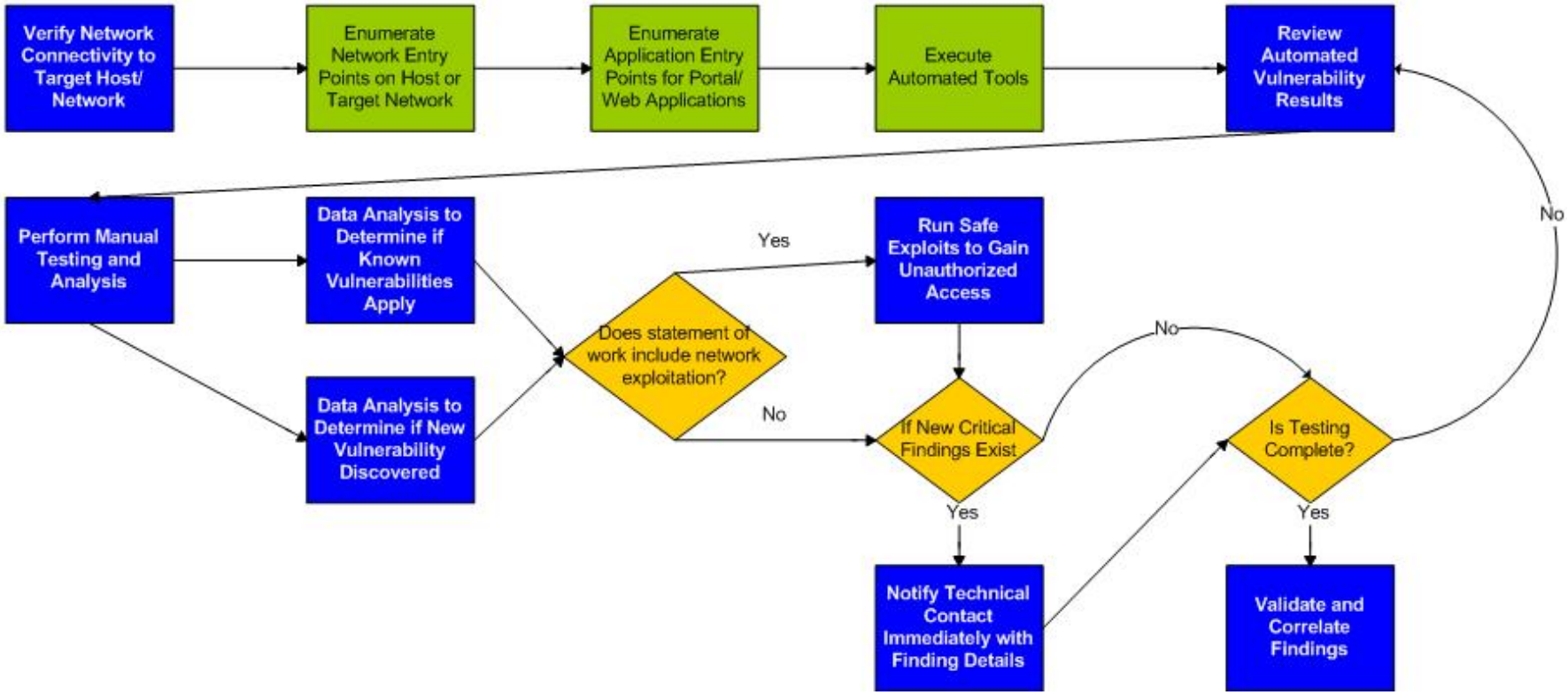
– Colin Powell

Non-Kinetic Warfare

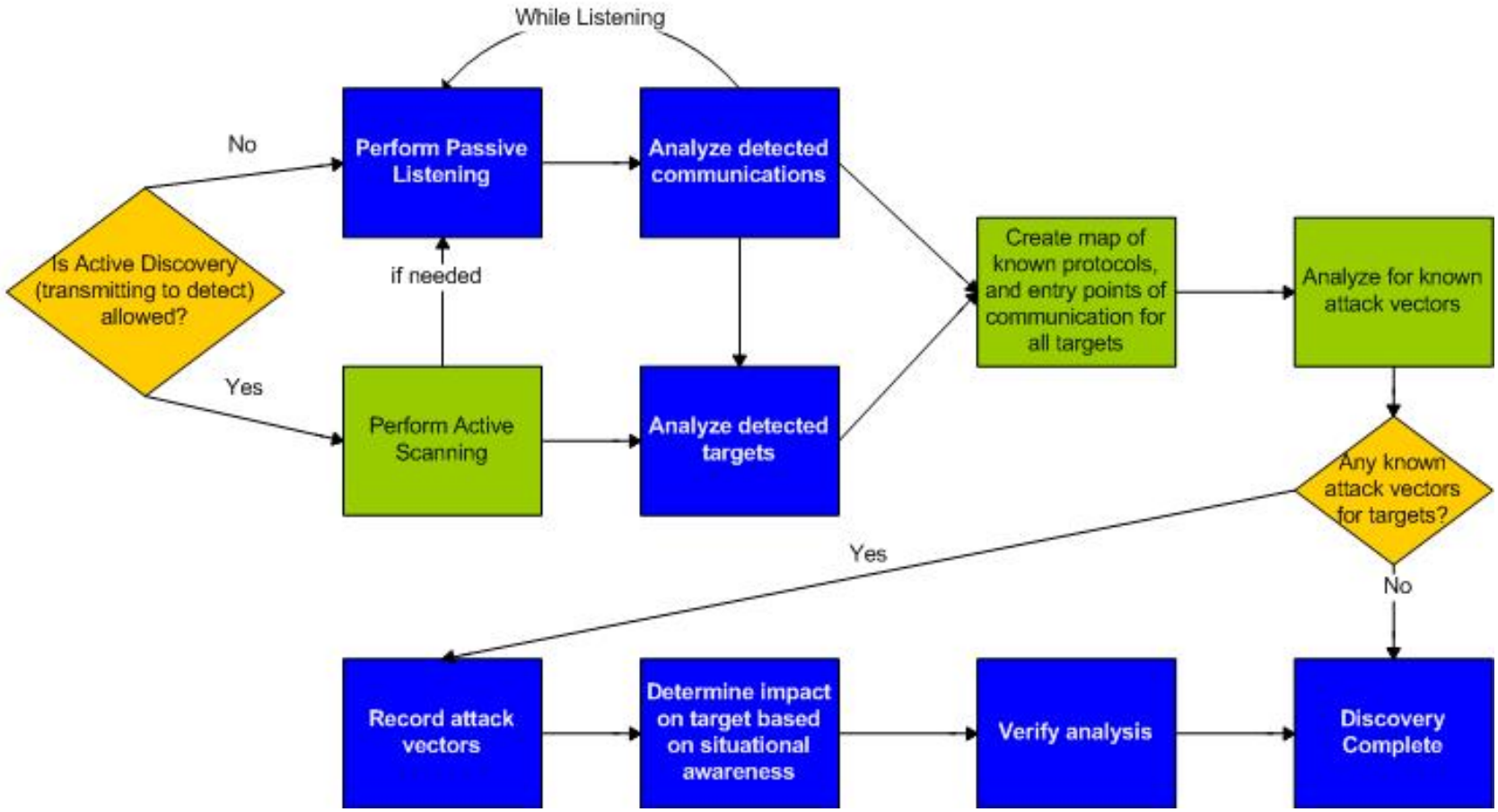
Non-Kinetic Warfare – Hacking

- Really just Security Testing?
- Zero-Day Exploits
  - Arming intelligence communities
  - Supports defending U.S. resources

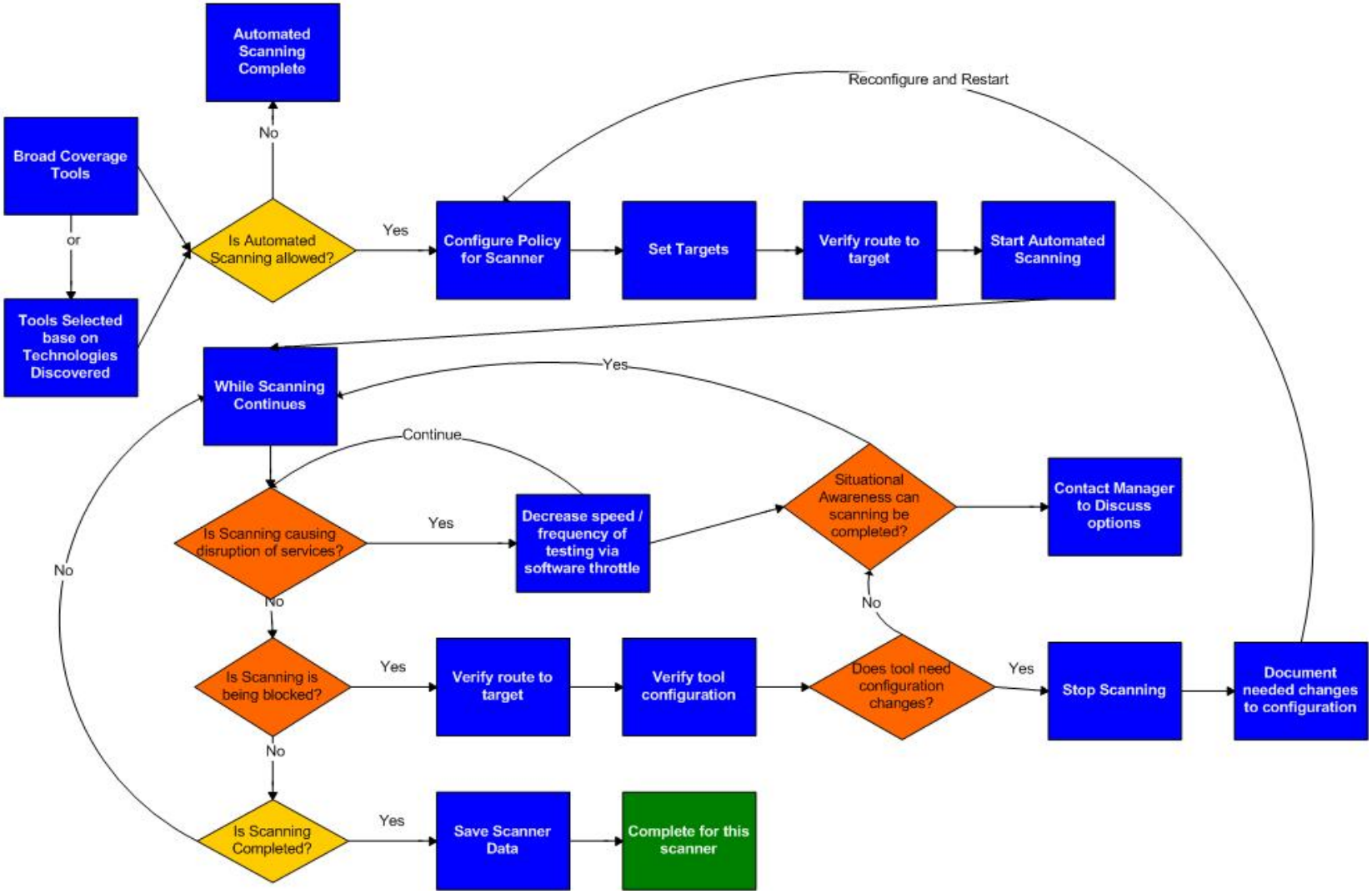
Application Testing…



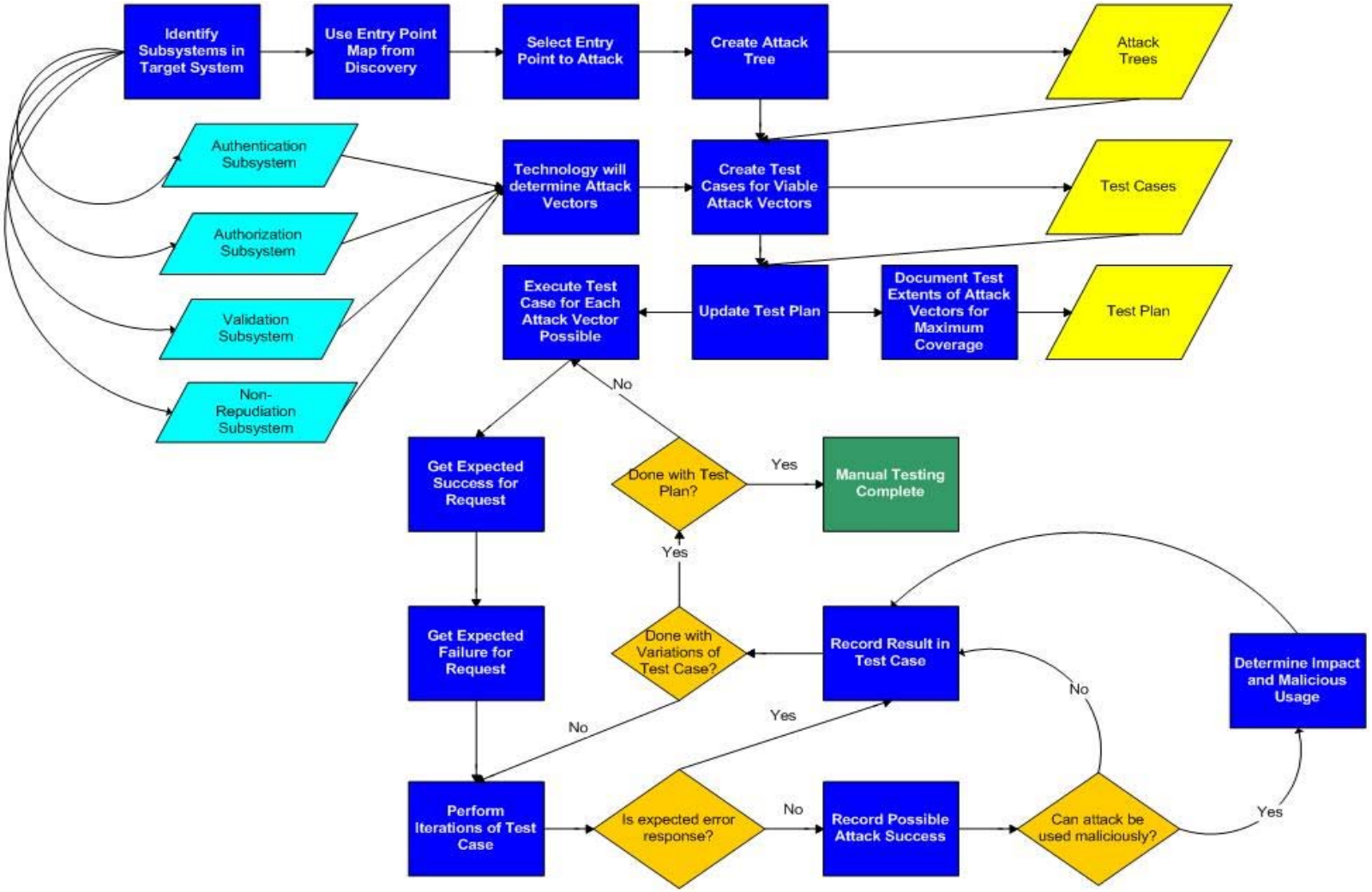
Active/Passive Discovery (Surveillance)



Not So Automated-Automated Testing



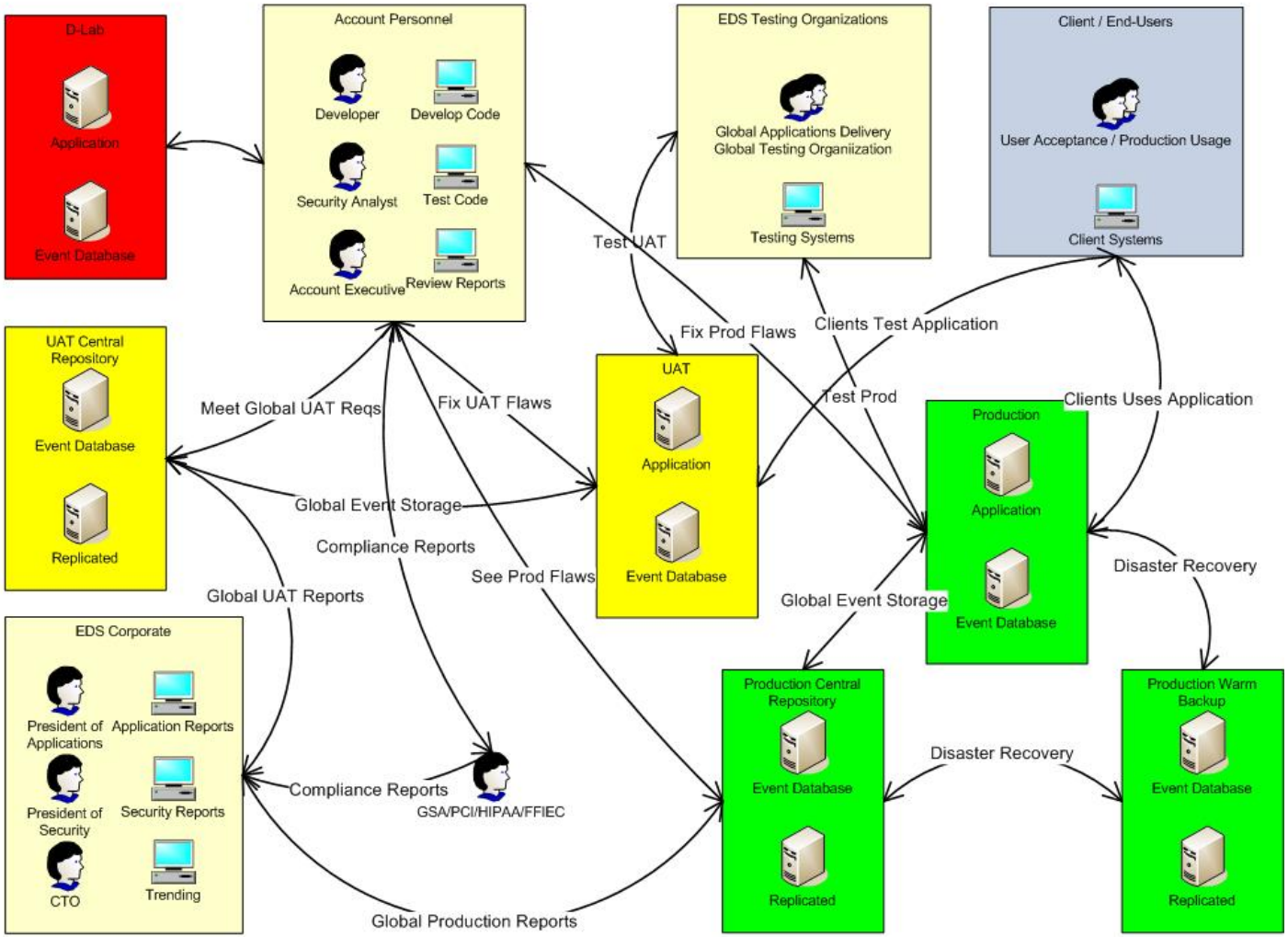
Cyber Offense (Using/Finding Zero-Day Exploits)



Outcome of Testing

- Attempted attacks can cause failure case during
  - Authentication
  - Authorization
  - Validation
  - Non-repudiation
- Attack vector proven secure
- New metrics to show
  - Coverage
  - Attack surface
  - Context surrounding attacks

Does the Lifecycle really change?



“You cannot command what you do not control.”

– Dr. David J. Kilcullen

“You cannot control what you cannot measure.”

– Tom DeMarco

Command & Control: Software Style

Application Integrated Cyber Intelligence

- Time based security model
- WIKIDS information hierarchy
- Application situational awareness
- Cyber space situational awareness
- Research and development

Time-Based Security

- TBS tenets
  - Protection fails
  - Attacks succeed
  - Detection must occur before protection fails
  - Response must occur before attack succeeds
- Not a new concept
  - Winn Schwartau
    - 1999 book
  - Some metrics in use circa 1998
    - Mean time between incidents
    - Mean time to detection
    - Mean time to respond
    - Mean time to escalate
- Not new does not mean in widespread use

Time-Based Security: Basics

- Basic time based security calculation for success
  - DRT < AST
  - PFT > DRT
- Basic time based security calculation for failure
  - (Detection + Response) > Protection > Attack
- Reality check
  - Many attacks happen in “real time”
  - Detection and response must be combined with protection

Time-Based Security: Examples

- Castle
  - Protection = Wall, moat, gate, etc
  - Detection = Watchmen
  - Response = Archers, hot tar
- Bank
  - Protection = Safe
  - Detection = Alarm
  - Response = Armed guards
- Network
  - Protection = Firewall
  - Detection = NIDS
  - Response = Manual Analysis & Response / NIPS

Time-Based Security: Input Validation

- Simple overflow validation
- Poor time based security implementation

```
if( input.Length < 16 )
{
        //do work
}
```

- What's missing?
  - Hint: You are protected

Time-Based Security: Validation

```
//comparison upper bounds extent
if( input.Length < 16 )
{
        //do work
}
else
{
        //detect defect
        //escalate event
        //respond to condition
}
```

Time-Based Security

- Requires detection and response
- There is no reason not to do this
- This code must be engineered not “coded”:
  - Centralized, Modular, etc.

WIKIDS Basics

- Symbols
  - Glyphs that represent a value
- Data
  - Raw value without context
- Information
  - Data in context used to convey meaning
- Knowledge
  - Collected, understood and utilized information
- Intelligence
  - Uncertain deductions resulting from knowledge
- Wisdom
  - Application of intelligence to actions

WIKIDS: Computers, Humans and Symbols

- Data values are absolute
- Symbols are context sensitive
- Humans
  - Cognize: know or understand
  - Cognize context based symbols
  - Assume context based symbol value
- Computers
  - Interpret: present in understandable terms
  - Interpret values to symbols for human consumption
  - Simply perform the action which they are told to perform

WIKIDS: Homoglyph Attack

- Symbolic glyph
  - ‘a’
- Possible glyph values
  - Unicode 0x0430 == Cyrillic Alphabet
  - Unicode 0x0061 == Basic Latin Alphabet
- Which glyph value is used below?
  - www.paypal.com
- What value is “understood” by the user?
- What value is used by the browser?
  - Should software be able to determine the appropriateness of the underlying value?
    - *I hope you were listening to Drew about contextual validation =)*

WIKIDS: Homoglyph Attack Detection

```
UnicodeEncoding unicode = new UnicodeEncoding();

Byte[] encodedBytes = unicode.GetBytes(unicodeInput);

for( nIndex=0; nIndex=input.Length; nIndex++ )
{
        //Latin-1 printable, non-white space char set
        if((0x0021 <= encodedBytes[nIndex])
                && (encodedBytes[nIndex] <= 0x007E))
        {
                // do work
        }
        else
        {
                //detect, escalate, respond as appropriate
        }
}
```

Application Situational Awareness

- Is this data value appropriate in this context?
  - Yes or No
  - This gets us one half way through the cycle
- Event escalation
  - Gets us to the point of scope specific application intelligence

Application WIKIDS	
Wisdom	?
Intelligence	Event Escalation
Knowledge	Failure Detection
Information	Validation Failed
Data	0x0430
Symbol	‘a’

Cyber Space Situational Awareness

- Inputs
  - Application Meta Data
  - Application Event Data
  - Application Event Analysis
  - Application Event Aggregation
  - Application Event Correlation
- Integrate with system and network inputs

Application Meta Data

- Built into application monitoring system
  - Application owner
  - Application function
  - Application dependencies
  - Application stake holders
  - Application outage implication statement

Application Event Data

- So data must an “application security event” provide?
- Whatever the application knows =)
  - Timestamp= 9/29/2008 8:10:39 PM
  - Hostname= noble001
  - Application= Demo
  - UTCTicks= 633583014397812500
  - Priority= High
  - RunningAs= NOBLE001\rcm
  - UserContext= User=NOBLE001\rcm Authenticated=True Method=NTLM
  - AttackSRCIP= 127.0.0.1
  - AttackDSTIP= 127.0.0.1
  - InputField= 'Login username' (System.String)
  - EventType= Underflow
  - AppDefined= 0
  - Action= Expected at least 8 bytes. Received 1 bytes.
  - Data= 61

Application Event Analysis

- For anyone that has done network analysis life just got easy
  - Security and operations personnel
- All events generated this way are true, no false positives
  - Go/No-Go
- Conclusions do not have to be inferred
  - True positive, raw data element failed value

Application Event Aggregation

- Direct SIM reporting
  - Application => SIM
- Centralized reporting
  - Application => Database => SIM
- OS logging facilities
  - Application => Event Viewer => Agent => SIM
- Flat text file
  - Application => File => Agent => SIM
- Actual practice prefers centralized reporting
  - Allows data mining outside of SIM data

Application Event Correlation

- Possibilities
  - Same event payload different source addresses
  - Same event payload varying time at different application
  - Same source different user contexts within small timeframe
  - Same event type different payloads from many source addresses
  - Same event payload different applications
- Statistical Methods
- Large Scale Monitoring
- Correlations are not causal relationships

Quick Review

- Time based security
  - Attacks must be detected and responded to before protection fails
- WIKIDS
  - Data is the quanta of attack and defense
  - If you don't know what your data element values are you will lose
- Application Situational Awareness
  - Instrument your applications
  - Aggregate the events
  - Correlate the events
  - Monitor the events
- Cyber Space Situational Awareness
  - Integrate multiple levels of data and meta data
  - Provides strategic intelligence

Questions / Comments

EDS
5400 Legacy Drive
Plano, TX 75024
Joseph.Miller@eds.com, Richard.MacVarish@eds.com

EDS and the EDS logo are registered trademarks of Hewlett-Packard Development Company, LP. HP is an equal opportunity employer and values the diversity of its people. ©2008 Hewlett-Packard Development Company, LP.

Behind success there's EDS an HP company