CWE, CAPEC Integration in Risk Based Threat Modeling

Tony Uceda Velez
CEO, VerSprite

August 31, 2015
Introduction

Tony UcedaVélez ("Tony UV")
- CEO, VerSprite – Global Security Consulting Firm
- Chapter Leader – OWASP Atlanta (past 7 years)
What Threat Are You Protecting Against?

- Do you know who may attack you?
- Do you know why they may attack you?
- Do you know what evidence support your threat claims?
- Use MITRE’s CAPEC & CWE to organize your attack and weakness libraries
What is PASTA?

- Process for Attack Simulation & Threat Analysis
  - Risk centric threat modeling methodology
  - Collaborative; great for business integration
  - 7 stages building up to impact of threat to application & business.
- Aimed at addressing most viable threats & building security in
Threat Modeling

**Threat Dissection**

- Application
- Component
- Some data
- A data store
- Do Something

**Targeted Analysis**

- Focused on understanding targeted threats
- Focus on attacks that are supported via viable threat patterns (considering multiple vectors)
- Threat motives may be data (e.g. - PII, IP), disruption based (hacktivism), IP
Threat. A threat is an undesired event. A potential occurrence, often best described as causal factors that may manifest into attacks that compromise an asset or objective. Relative to each site, industry, company; more difficult to uniformly define.
Risk Centric Threat Modeling

Risk Management

- Needs to substantiate risks
  - No one believes your risk scores
- Substantiate vulnerable findings w/ threat modeling stages
  - 3 (app decomposition)
  - 4 (threat analysis)
  - 5 (vuln detection)
  - 6 (exploitation)
- Vulnerabilities begin to ‘mean’ something to those who have to remediate them

Attack Tree
LEVERAGING CAPEC & CWE
What Is CWE?

Targeted to developers and security practitioners, the Common Weakness Enumeration (CWE) is a formal list of software weakness types created to:

- Serve as a common language for describing software security weaknesses in architecture, design, or code.
- Serve as a standard measuring stick for software security tools targeting these weaknesses.
- Provide a common baseline standard for weakness identification, mitigation, and prevention efforts.
What is CAPEC?

Objective
The objective of the Common Attack Pattern Enumeration and Classification (CAPEC™) effort is to provide a publicly available catalog of common attack patterns classified in an intuitive manner, along with a comprehensive schema for describing related attacks and sharing information about them.
<table>
<thead>
<tr>
<th>Primary Schema Elements</th>
<th>Supporting Schema Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Identifying Information</strong></td>
<td><strong>Describing Information</strong></td>
</tr>
<tr>
<td>- Attack Pattern ID</td>
<td>- Injection Vector</td>
</tr>
<tr>
<td>- Attack Pattern Name</td>
<td>- Payload</td>
</tr>
<tr>
<td><strong>Describing Information</strong></td>
<td>- Activation Zone</td>
</tr>
<tr>
<td>- Description</td>
<td>- Payload Activation Impact</td>
</tr>
<tr>
<td>- Related Weaknesses</td>
<td><strong>Diagnosing Information</strong></td>
</tr>
<tr>
<td>- Related Vulnerabilities</td>
<td>- Probing Techniques</td>
</tr>
<tr>
<td>- Method of Attack</td>
<td>- Indicators-Warnings of Attack</td>
</tr>
<tr>
<td>- Examples-Instances</td>
<td>- Obfuscation Techniques</td>
</tr>
<tr>
<td>- References</td>
<td><strong>Enhancing Information</strong></td>
</tr>
<tr>
<td><strong>Prescribing Information</strong></td>
<td>- Related Attack Patterns</td>
</tr>
<tr>
<td>- Solutions and Mitigations</td>
<td>- Relevant Security Requirements</td>
</tr>
<tr>
<td><strong>Scoping and Delimiting Information</strong></td>
<td>- Relevant Design Patterns</td>
</tr>
<tr>
<td>- Typical Severity</td>
<td>- Relevant Security Patterns</td>
</tr>
<tr>
<td>- Typical Likelihood of Exploit</td>
<td></td>
</tr>
<tr>
<td>- Attack Prerequisites</td>
<td></td>
</tr>
<tr>
<td>- Attacker Skill or Knowledge Required</td>
<td></td>
</tr>
<tr>
<td>- Resources Required</td>
<td></td>
</tr>
<tr>
<td>- Attack Motivation-Consequences</td>
<td></td>
</tr>
<tr>
<td>- Context Description</td>
<td></td>
</tr>
</tbody>
</table>
Use Case

Functional, as designed function of an application.
Abuse Case.

Deliberate abuse of functional use cases in order to yield unintended results.
**Attack Surface**

Logical area (browser stack, infrastructure components, etc) or physical area (hotel kiosk).

Where do you define this in risk assessments or compliance audits?
Attack Vector

Point & channel for which attacks travel over (card reader, form fields, network proxy, client browser, etc.)
Attack Trees

Attack Tree. Helpful diagram of relationship amongst asset-actor-use case-abuse case-vuln-exploit-countermeasur e
CAPEC + CWE Use in Attack Trees

PhySec Client Software

Use Case: Run Security Status

CAPEC-0014 Client Side Injection Induced Buffer Overflow

CWE-0119 Buffer Errors

CWE-0074 Injection

CWE-0020 Input Validation
Analysis Of Attacks Using Attack Trees

Fraudster

Upload Malware on Vulnerable Site
- Drive-by Download/Malicious Ads
- Steal Digital Certificates For Authentication
- Delete Cookies Forcing to Login To Steal Logins

Attack Victim's Vulnerable Browser
- Upload Banking Malware on Customer's PC
- Man In The Browser
- Modifies UI Rendered By The Browser
- Harvest Confidential Data/Credentials From Victim

Phishing Email, FaceBook Social Engineering
- Phish User To Click Link With Malware
- Steals Keystrokes with Key-logger
- Redirect Users To Malicious Sites
- Sends Stolen Data to Fraudster's Collection Server

Fraudster

Use Stolen Banking Credentials/Challenge C/Q
- Remote Access To Compromised PC Through Proxy
- Logs into Victim's Online Bank Account
- Perform Unauthorized Money Transfer to Mule
- Money Transferred From Mule to Fraudster
Attack Model

**Threat**
- Gain unauthorized router access over HTTP

**Asset/Target**
- Cisco Router and Security Device Manager (Asset/Target)

**Vulnerability**
- Unspecified Cross Site Scripting Vulnerability
  - CVE-2010-0594

**Attack**
- Malicious message from Attacker to Administrator containing suspect URL

**Security Process**
- IDs vuln on Router or Security Device Mgr via scans
- Communicates Vuln to Network Engineering
- Security Operations

**Reveals weaknesses in patching network equipment**

**VerSprite**
Layered Attacks to Thwart Countermeasures

1. **User**
   - Login With UserID and password over SSL
   - Trust connection by IP and machine tagging/browser attributes

2. **Fraudster**
   - Communicate with fraudster C&C

3. **Key logger/From grabber captures keystrokes incl. credentials**
   - Enter One Time Password (OTP) to authenticate transaction
   - Enter Challenge Question (C/Q) to authenticate transaction

4. **Drops Banking Malware on victims/PC**
   - Man In The Browser Injected HTML to capture C/Q

5. **Set IP with Proxy/MiTM to same IP gelocation of the victim**
   - Hijacks SessionIDs, Cookies, Machine Tagging

6. **Includes**
   - Capture OTP on web channel and authenticate on behalf of the user
   - Capture C/Qs in transit and authenticate on behalf of user
   - Key logger/From grabber captures keystrokes incl. credentials
OWASP Tie-In

**OWASP WASC Web Hacking Incidents Database Project**
- project dedicated to maintaining a list of web applications related security incidents.
- [https://www.owasp.org/index.php/OWASP_WASC_Web_Hacking_Incidents_Database_Project](https://www.owasp.org/index.php/OWASP_WASC_Web_Hacking_Incidents_Database_Project)

**OWASP Security Knowledge Framework**
- a tool that is used as a guide for building and verifying secure software. It can also be used to train developers about application security.

**Incorporates Applications Security Verification Standard**
Security Convergence via PASTA

Impact Landscape
- Data Losses
- Online Fraud
- Card Fraud
- Denial of Service
- Defacing
- Reputation Loss
- Client Lawsuits
- Unlawful Non Compliance

Controls Landscape
- Anti-malware
- Anti-automation
- Virtual Browsing
- Strong Authentication
- Transaction Verification
- Maker/Checker Process
- Anomaly Detection

Asset Landscape
- Customer Data:
  - Credit/debit card Data
  - Bank Account Data
  - Confidential-PII Data
- Application Data:
  - Logging Credentials
  - Challenge/Questions
  - Passwords
  - Transaction Data
  - Session Tokens

Threat Landscape
- Attack Customers:
  - Phishing Emails
  - Malicious URLs
  - Virus Infected Documents
  - Social Engineering
- Attack the Browser:
  - Drive by Download
  - Click Jacking
  - HTML injection
  - Man in the Browser
- Attack the Web Application:
  - Vulnerability Exploits
  - Business Logic/Flaws Attacks
  - Session Hijacking
  - Man in The Middle

PASTA METHODOLOGY
Stage 1 - Understand Biz Objectives behind Security, Compliance

Input:
- Business requirement documents
  - Functional requirement documents
  - Information Security Policies
  - Regulatory Compliance
  - Security Standards & Guidelines
  - Data Classification Documents

(1.0) Stage 1: Define the Objectives
- (1.1) Define Business Objectives
- (1.2) Define Security Requirements
- (1.3) Define Compliance Requirements
- (1.4) Perform preliminary Business Impact Analysis (BIA)

Output:
- Description of the application functionality
  - List of business objectives
  - Definition of the application security and compliance requirements
  - Business Impact Analysis Report
Baking in GRC

- Serve as inherent countermeasures in the form of people, process, technology
  - Policies (for people)
  - Standards (for technology)
- Prior risk assessments help build app risk profile
  - Historical RAs provide prior risk profile of app
- Regulatory landscape taken into consideration, but not the driver
  - Key here is to not retrofit compliance; more costly
- Web Related Example:
  - Tech: Using Nessus OWASP template to audit for PHP & ColdFusion hardening guidelines
  - OWASP Input Validation Cheat Sheets
  - CIS Web Technology Benchmarks
**Threat Modeling Stage 1 Artifact**

### Application Profile: Online Banking Application

| General Description | The online banking application allows customers to perform banking activities such as financial transactions over the internet. The type of transactions supported by the application includes bill payments, wires, funds transfers between customer’s own accounts and other bank institutions, account balance-inquires, transaction inquires, bank statements, new bank accounts loan and credit card applications. New online customers can register an online account using existing debit card, PIN and account information. Customers authenticate to the application using username and password and different types of Multi Factor Authentication (MFA) and Risk Based Authentication (RBA) |
| Application Type | Internet Facing |
| Data Classification | Public, Non Confidential, Sensitive and Confidential PII |
| Inherent Risk | HIGH (Infrastructure, Limited Trust Boundary, Platform Risks, Accessability) |
| High Risk Transactions | YES |
| User roles | Visitor, customer, administrator, customer support representative |
| Number of users | 3 million registered customers |
Stage 2 Walkthru – Define Tech Scope

(2.0) Stage II: Define the Technical Scope

(2.1) Identify Application Boundaries
(2.2) Identify Application Dependencies From Network Environment
(2.3) Identify Application Dependencies from Servers/Infrastructure
(2.4) Identify Application Dependencies from Software

Input

High level design documents, Sketches from white-board exercises, Network diagrams, Logical and physical architecture diagrams, Software and technical specifications

Output

High level, end-to-end view List of all protocols and data List of all the application servers List of all hosts and servers and type of software/technology dependencies List of all network devices/appliances
The Application Architecture Scope
Technical Scope Definition

Define the scope from design artifacts:

- **Application components** with respect to the application tiers (presentation, application, data)
- **Network topology**
- **Protocol/services** being used/exposed from/to the user to/from the back end (e.g. data flow diagrams)
- **Use case scenarios** (e.g. sequence diagrams)

Model the application in support of security architecture risk analysis:

- **The application assets** (e.g. data/services at each tier)
- **The security controls of the application** (e.g. authentication, authorization, encryption, session management, input validation, auditing and logging)
- **Data interactions** between the user of the application and between servers for the main use case scenarios (e.g. login, registration, query etc)
Stage 3– App Decomposition

Input:
Architecture diagrams-design documents, Sequence diagrams, Use cases, Users, roles and permissions, Logical diagrams, Physical-network diagrams

Process:
(3.0) Stage III: Decompose the Application
(3.1) Data Flow Diagramming & Trust Boundaries
(3.2) Identify Users-Agents and their Roles-Permissions
(3.3) Identify Assets, Data, Services, Hardware and Software
(3.4) Identify Data Entry Points and Trust Levels

Output:
Data Flow Diagrams
Access control matrix
List of assets including data and data sources
List of interfaces and trust levels
Mapping of use cases with actors and assets
Data Flow Diagramming (DFD)
Online Banking Application Example

User/Browser

Web Server

HTTPs Request

HTTPs Responses

DMZ (User/Web Server Boundary)

Internal (Web Server/App & DB Server Boundary)

Application Server

Application Calls (.do)

XML/HTTPS

Financial Transactions (ACH, wires, external transfer)

Restricted Network (App & DB Server/Financial Server Boundary)

Messaging Bus

XML/HTTPS

Message: XML/JMS

Service Message Response

SQL Query Call/JDBC

Restricted Network (App & DB Server/Financial Server Boundary)

Authentication Credential Store

MFA RBA/Fraud Detection

User Authentication Credential

Authentication Credential Store

Financial Transaction Processing MainFrame

Internal (Web Server/App & DB Server Boundary)

Application Responses

Internal (Web Server/App & DB Server Boundary)

Application Server

HTTPs Responses

HTTPs

VerSprite
<table>
<thead>
<tr>
<th>Transaction</th>
<th>Risk</th>
<th>Data Classification</th>
<th>Security Functions Invoked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password Reset</td>
<td>HIGH</td>
<td>Sensitive</td>
<td>Debit Card, PIN, Account#&lt;br&gt;Challenge/Questions Risk Interdicted&lt;br&gt;Pre-Auth/Bank Customer&lt;br&gt;Pre-auth SessionID/Cookie&lt;br&gt;HTTPS&lt;br&gt;Custom Errors &amp; Messages&lt;br&gt;Application, Fraud Detection</td>
</tr>
<tr>
<td>Username Recovery</td>
<td>HIGH</td>
<td>Sensitive</td>
<td>Debit Card, PIN, Account#&lt;br&gt;Challenge/Questions Risk Interdicted&lt;br&gt;Pre-Auth/Bank Customer&lt;br&gt;Pre-auth SessionID/Cookie&lt;br&gt;HTTPS&lt;br&gt;Custom Errors &amp; Messages&lt;br&gt;Application, Fraud Detection</td>
</tr>
<tr>
<td>Registration</td>
<td>MEDIUM</td>
<td>Confidential PII &amp; Sensitive&lt;br&gt;PII (e.g. SSN), Demographics</td>
<td>OOB/Confirmation&lt;br&gt;Visitor&lt;br&gt;Pre-auth SessionID/Cookie&lt;br&gt;HTTPS&lt;br&gt;Custom Errors &amp; Messages&lt;br&gt;Application</td>
</tr>
<tr>
<td>Logon</td>
<td>HIGH</td>
<td>Confidential PII &amp; Sensitive&lt;br&gt;Username/Password</td>
<td>Single Auth + Challenge/Questions Risk Interdicted&lt;br&gt;Post-Auth/Bank Customer&lt;br&gt;Post-auth SessionID Mgmt&lt;br&gt;HTTPS/3DES Token&lt;br&gt;Custom Errors &amp; Messages&lt;br&gt;Application, Fraud Detection</td>
</tr>
<tr>
<td>Wires</td>
<td>HIGH</td>
<td>Confidential PII &amp; Sensitive&lt;br&gt;Amount, Account#, IBAN/BIC</td>
<td>Single Auth + C/Q Risk Interdicted + OTP&lt;br&gt;Post-Auth/Bank Customer&lt;br&gt;Post-auth SessionID Mgmt&lt;br&gt;HTTPS&lt;br&gt;Custom Errors &amp; Messages&lt;br&gt;Application, Fraud Detection</td>
</tr>
<tr>
<td>Bill Pay</td>
<td>HIGH</td>
<td>Confidential PII &amp; Sensitive&lt;br&gt;Amount, Payee Account#</td>
<td>Single Auth + C/Q Risk Interdicted + OTP&lt;br&gt;Post-Auth/Bank Customer&lt;br&gt;Post-auth SessionID Mgmt&lt;br&gt;HTTPS&lt;br&gt;Custom Errors &amp; Messages&lt;br&gt;Application, Fraud Detection</td>
</tr>
</tbody>
</table>
Stage 4 Threat Intelligence/Analysis

(4.0) Stage IV: Analyze the Threats

- (4.1) Analyze Probabilistic Attack Scenarios
- (4.2) Analyze Incidents and Fraud-Case Management Reports
- (4.3) Analyze Application Logs And Security Events
- (4.4) Correlate Incidents and Fraud with Threat Intelligence

Input:
- Threat agents and motives
- Security incidents (SIRT) report
- Fraud detection report
- Secure incident event monitoring (SIEM) reports
- Application and server logs
- Threat intelligence reports

Output:
- Attack scenario-landscape report
  - List of threat agents and attacks
  - Report on incidents-events relevant to the likelihood of threats and attack scenarios
  - Correlation to threat intelligence reports for attack scenarios
Threat Intelligence is Golden

**Threat Enumeration Based upon Good Intel**
- Threats based upon known intel
- Prior assessment info (where applicable & useful)
- Other application assessments from 3rd parties
- SIEM feeds/ Syslog data/ Application Logs/ WAF logs
  - Denote attacks but will reveal overarching threats
- Threat Intel/ Feeds
- Security Operations/ Incident Reports
  - Personnel/ Infrastructure

**Threat examples:**
- IP Theft
- Data Theft
- Sabotage
- Infrastructure compromise
- Ransom
Threat Analysis Prefaces Attack Enumeration

• Threat analysis will lead to attack enumeration
  – PII theft
    – XSS
    – SQL Injection
    – MITM
  – Sabotage driven threats
    – CMS exploits to web application (Zope, Joomla, Mambo, etc)
    – FTP Brute Force attacks
    – iFrame Injection attacks
  – Malware upload
• Identify most likely attack vectors
  – Address entire application footprint (email, client app, etc)
  – Web Forms/ Fields
  – WSDLs/ SWF Objects
  – Compiled Libraries/ Named Pipes
Stage 5 Walkthru – Vuln Analysis

- Library of threat trees
  Attack scenarios (from Stage IV)
  Vulnerability Assessment reports
  Standards for vulnerability enumeration (MITRE CWE, CVE)
  Standards for vulnerability scoring (CVSS,CWSS)

(5.0) Stage Y: Vulnerabilities & Weaknesses Analysis

- (5.1) Correlate Vulnerabilities to Application Assets
- (5.2) Map Threat to Vulnerabilities Using Threat Trees
- (5.3) Map Threat To Security Flaws Using Use and Abuse Cases
- (5.4) Enumerate and Score Vulnerabilities

Map of existing vulnerabilities to the nodes of a threat tree
Enumeration of these vulnerabilities using CVE-CWE
Scoring of these using CVSS-CWSS
List of threats-attacks-vulnerabilities-assets
SecOps Convergence of Vulnerability Mgt.
Stage 6 Walkthru – Attack Enumeration

Application Technical Scope (Stage II) and Application Decomposition (Stage III) - Attack libraries-patterns, list of threats, attacks and vulnerabilities to the application assets (Stage V)

Input

(6.0) Stage VI: Model The Attacks

(6.1) Identify Application Attack Surface
(6.2) Derive Attack Trees For Threats and Assets
(6.3) Map Attack Vectors To Nodes of Attack Trees
(6.4) Identify Exploits and Attack Paths using Attack Trees

Output

Application attack surface - Attack trees with attack scenarios for targeted assets, attack tree mapping to vulnerabilities for impacted assets, list of possible attack paths to exploits including the attack vectors
Stage 7- Residual Risk Analysis

Preliminary BIA (Stage I)  
Technical Scope (Stage II)  
Application Decomposition (Stage III)  
Threat Analysis (Stage IV)  
Vulnerability Analysis (Stage V)  
Attack Analysis (Stage VI)  
Mapping of attacks to controls  
Technical standards for controls

(7.0) Stage VII: Risk and Impact Analysis

(7.1) Qualify and Quantify Business Impacts  
(7.2) Identify Gaps in Security Controls  
(7.3) Calculate Residual Risks  
(7.4) Identify Risk Mitigation Strategies

Application risk profile  
Quantitative and qualitative risks report  
Threat matrix with threats, attacks, vulnerabilities, business impact  
Residual risk to business  
Risk mitigation strategy-options
Users
Request
Web Server
ESAPI/ISAPI Filter
Custom errors
Application Calls
XSS, SQL Injection, Information Disclosure Via errors
Application Responses
DMZ (User/Web Server Boundary)
Message Call
Account/Transaction Query Calls
Web Server
Application Server
Encryption + Authentication
Internal (Web Server/App & DB Server Boundary)
Insecure Direct Obj. Ref, Insecure Remote File Inclusion
Message Response
Message Encryption + Authentication
(Data + DB Server/Financial Server Boundary)
Financial Server
Broken Authentication, Federation, Mutual Authentication
Trusted Authentication, Federation, Mutual Authentication
Customer Financial Data
Account/Transaction Query Calls
Encrypt Confidential PII in Storage/Transit
Financial Data
Insecure Crypto Storage
Phishing, Privacy Violations, Financial Loss
Identity Theft System Compromise, Data Alteration, Destruction
"././././etc/passwd %00"
OR 'I'='I--'
Cmd=%3B+mkdir+hackerDirectory
http://www.abc.com?RoleID
../../../../etc/passwd%00
SqlCommand=%3B+mkdir+hackerDirectory
http://www.abc.com?RoleID

"<SCRIPT>alert("Cookie"+document.cookie)</SCRIPT>"
Focus on the application as business asset target
- Risk \( \neq t \times v \times i \)
- Risk \( \neq t \times v \times i \times p \)

- Attack simulation enhances \( p \) probability coefficients
- Considers both inherent countermeasures & those to be developed
- Focused on minimizing risks to applications and associated impacts to business

\[
R_{\text{risk}} = \left(\frac{t_p \times v_p}{c}\right) \times i
\]