

Secure Coding Practices

Introductions

Justin Goodhart

Developer in an Information Security Team

Learning how to be a Security Analyst

Generally, it is much less expensive to build secure software than to correct security issues after the software package has been completed, not to mention the costs that may be associated with a security breach.

https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf

Confidentiality, Integrity, and Availability (CIA)

The goal of software security is to maintain the **confidentiality, integrity, and availability** of information resources in order to enable successful business operations.

Confidentiality

To ensure that information is disclosed only to authorized parties.

Integrity

The assurance that information is accurate, complete and valid, and has not been altered by an unauthorized action.

Availability

A measure of a system's accessibility and usability.

Other Popular Trilogies

- Information Objectives (Already mentioned)
Confidentiality, Integrity, and Availability (CIA)
- Scenario Goals
Prevent, Detect, and Respond
- Management Methodology
People, Process, and Technology

General Security Principles

Least privilege

The principle of least privilege means giving a user account only those privileges which are essential to that user's work and nothing more.

Simple is more secure

The larger and more complex that a system becomes the harder it becomes to secure. Larger systems have more areas of concern. More complex systems increase the likelihood of bugs or of making mistakes. Simpler is always more secure.

Never trust users

You should consider and be on guard against basic human mistakes. In general, you should be paranoid. Most users aren't out to get you, but one in 10,000 might be, and the thing is, you can't tell the difference, especially ahead of time.

Expect the unexpected

Security's not like chess where you can react to someone else's moves. You have to assume that you will be hacked and your job is to figure out how that will happen ahead of time. You have to prevent the crime before it happens. It's sometimes called a mystery in reverse.

Defense in depth

Originally defense in depth was a military term. The idea is to slow the advance of an attacker because over time an attack loses momentum, and therefore it's not as effective.

Defense in depth continued...

When we're talking about computers, we're talking about redundant security. There are three main areas that you'll want to focus on for defense in depth. And you want to have defense in all of these areas, and have it deeply in all of these areas as well.

- People
- Technology
- Operations

Defense in depth (People)

Defense in depth for people means:

- Writing a security policy
- Getting everyone educated
- Getting them to follow best practices
- Assigning responsibilities

Defense in depth (Technology)

Now when we talk about technology we're talking about having security throughout your entire technology stack. That is:

- Hardware
- Software
- Acquisition
- Maintenance

Defense in depth (Technology) continued...

Defense in depth in technology touches on:

- Firewalls
- Intrusion detection
- Server hardware
- Software

Defense in depth (Operations)

Defense in depth for operations means:

- Periodic security reviews
- Data handling procedures
- Monitoring responsibilities
- Establishing how do you respond to threats.

Multi-layered defense

Layered security, also known as layered defense, describes the practice of combining multiple mitigating security controls to protect resources and data.

The term bears some similarity to defense in depth, a term adopted from a military strategy that involves multiple layers of defense that resist rapid penetration by an attacker but yield rather than exhaust themselves by too-rigid tactics.

Layered security is regarded by some as merely a delaying tactic used to buy time to bring security resources to bear to deal with a malicious security cracker's activities.

Security through obscurity

The less information you give out the better. More information benefits hackers. Hackers rely on exposed information and feedback from their actions.

Security through obscurity continued...

Information helps the hacker by narrowing the field of possible exploits. So you want to limit exposed information. Don't report any more information than is absolutely necessary. It's similar to the idea of least privilege, but this is least information.

Security through obscurity continued...

However, some people misinterpret security obscurity to mean that it's a good idea to use random or fake filenames or directory names to confuse hackers.

There's a huge penalty to you and anyone who's working on the site if things are confusing, and possibly even to your overall security because this violates our other principle, that **simple is more secure**.

Security through obscurity continued...

As a side note to security through obscurity:

Information that is shared through poor configuration management of the environment can add up to an exploit. The information you share through the application code is not everything the attacker is going to go after.

Blacklisting

Blacklisting is listing what is forbidden.

We have to remember to add to our blacklist when new items come up.

Whitelisting

Whitelisting is listing what is permitted instead of what is forbidden.

Whitelisting means restricted by default (more secure than blacklisting). We don't have to do anything extra when new items are discovered.

Blacklisting and whitelisting

The choice to use whitelisting or blacklisting comes up in many different security areas. Learn to recognize it as a pattern when you see it, so that you can make a smart choice about which one to use.

When possible it is recommended to use the whitelisting approach. Security by default.

Map exposure points and data passageways

Mapping the exposure points and passage ways is a big part of the awareness side of things. If we could be aware of where they are, then we can make sure that they're protected.

It also helps us to expect the unexpected.

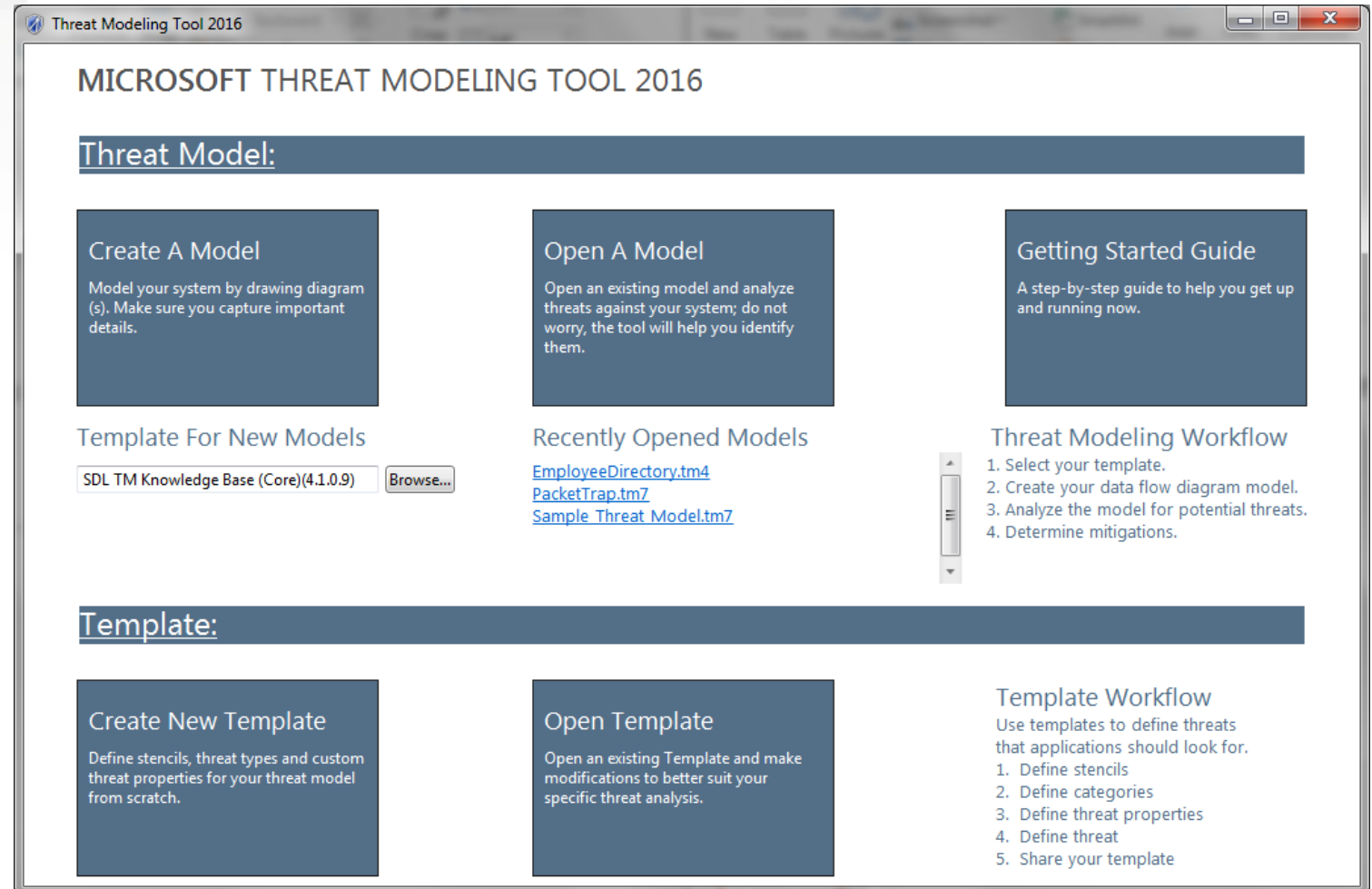
Threat Modeling



Microsoft Threat Modeling Tool 2016



Microsoft Threat Modeling Tool 2016



Fill in the threat model information

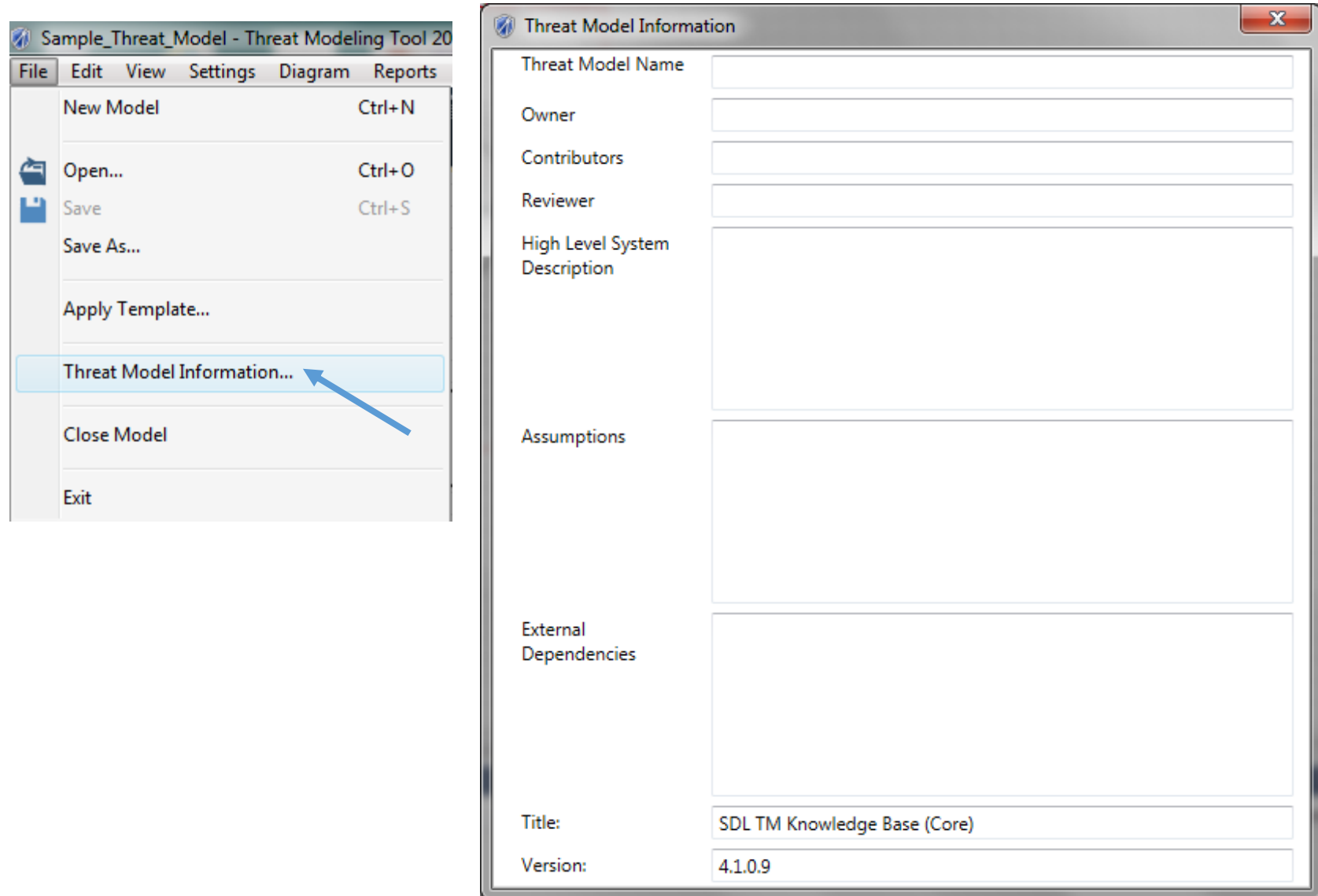
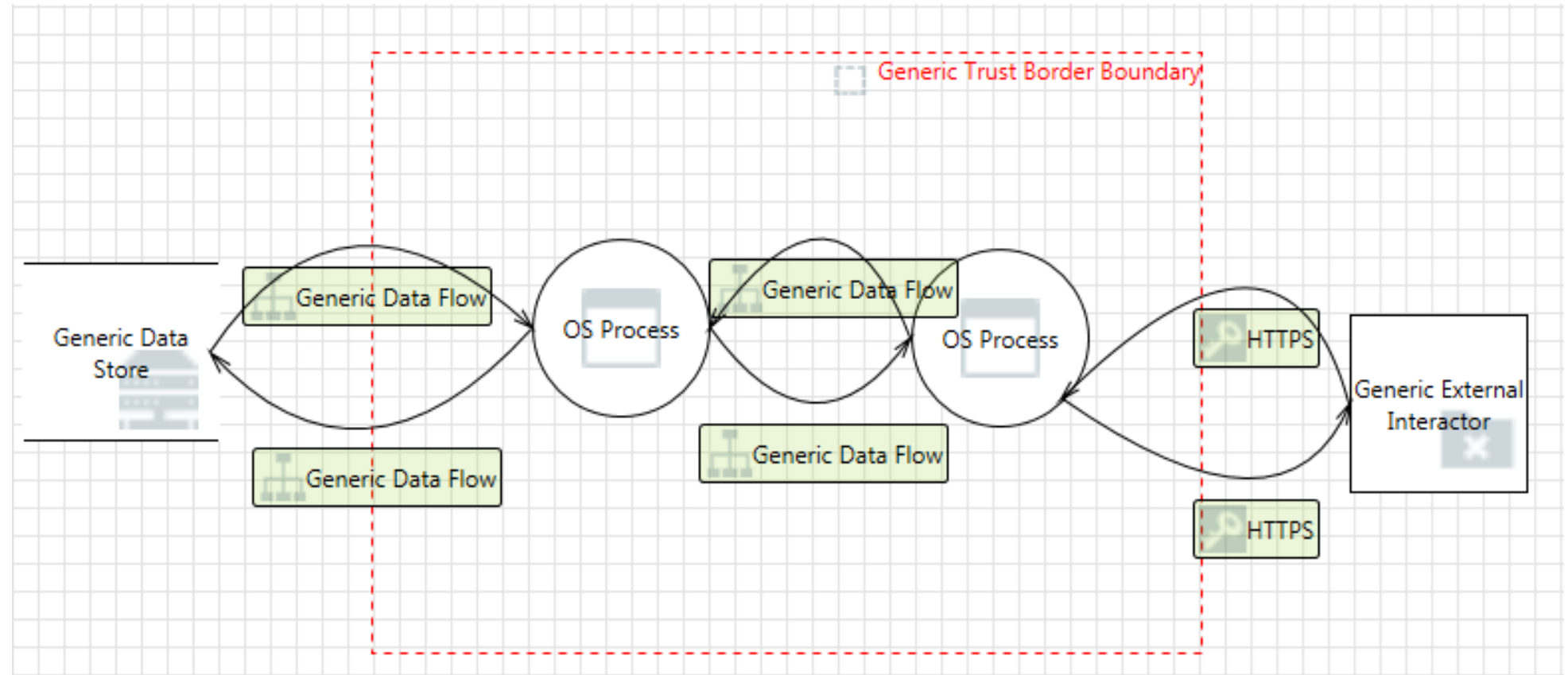
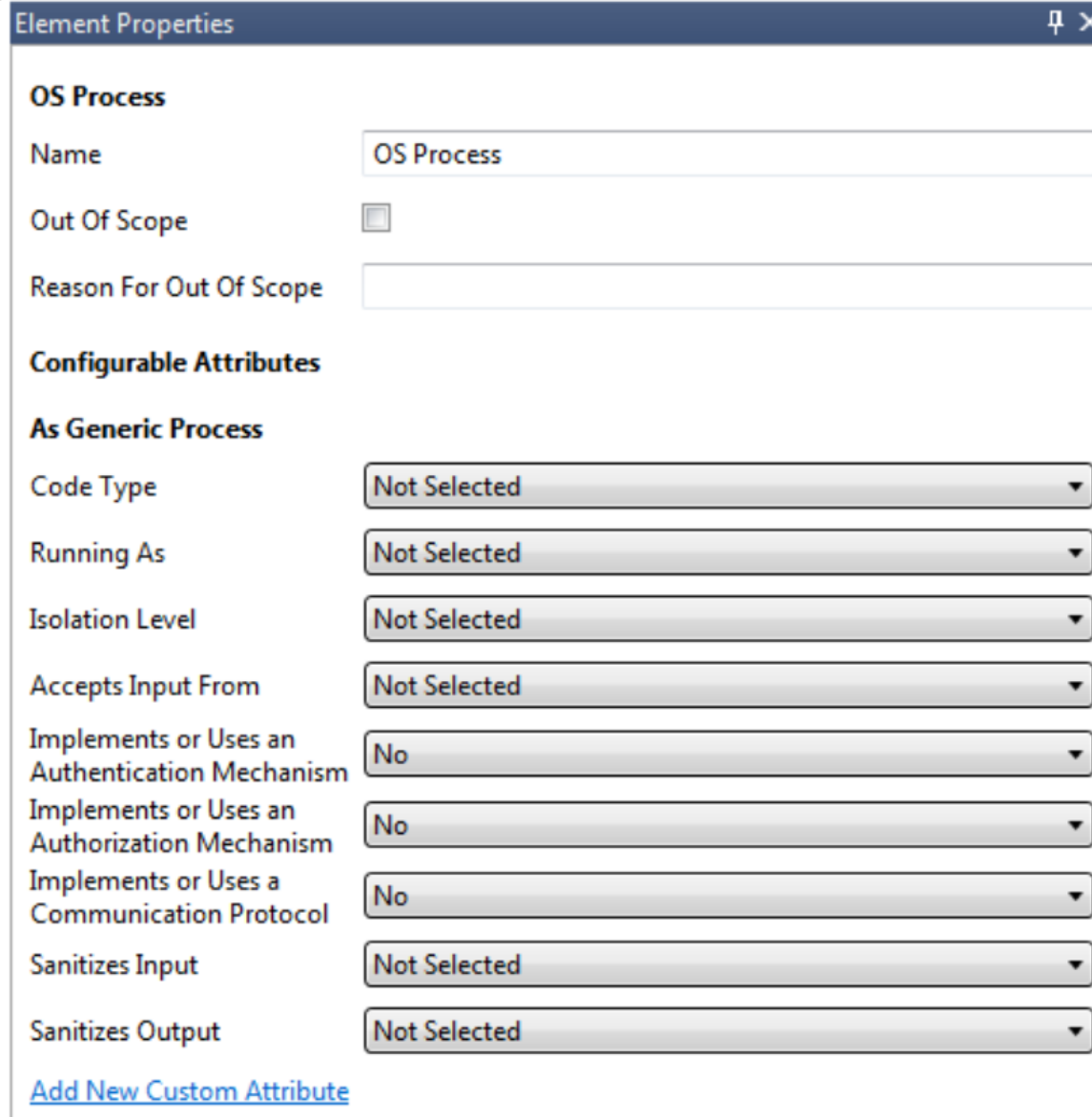


Diagram the system



Fill in the properties



The screenshot shows a software configuration window titled "Element Properties". It contains several sections for defining an "OS Process".

OS Process

- Name:** A text field containing "OS Process".
- Out Of Scope:** An unchecked checkbox.
- Reason For Out Of Scope:** An empty text field.

Configurable Attributes

As Generic Process

- Code Type:** A dropdown menu with "Not Selected" selected.
- Running As:** A dropdown menu with "Not Selected" selected.
- Isolation Level:** A dropdown menu with "Not Selected" selected.
- Accepts Input From:** A dropdown menu with "Not Selected" selected.
- Implements or Uses an Authentication Mechanism:** A dropdown menu with "No" selected.
- Implements or Uses an Authorization Mechanism:** A dropdown menu with "No" selected.
- Implements or Uses a Communication Protocol:** A dropdown menu with "No" selected.
- Sanitizes Input:** A dropdown menu with "Not Selected" selected.
- Sanitizes Output:** A dropdown menu with "Not Selected" selected.

[Add New Custom Attribute](#)

Determine and rank threats

Switch from Design View to Analysis View to see the identified threats.

Ranking the Threats is a simple dropdown for each of the generated threats in the Threat Modeling Tool and if there are questions on the rank level we can work with the Security Team.

STRIDE

- **Spoofing**
- **Tampering**
- **Repudiation**
- **Information disclosure**
- **Denial of service**
- **Elevation of privilege**

Spoofing

To illegally access and use another user's credentials, such as username and password.

Mitigations:

- Strong authentication
- Hash validation
- Protecting data in transit
- Protecting data at rest

Tampering

To maliciously change/modify persistent data, such as persistent data in a database, and the alteration of data in transit between two computers over an open network, such as the Internet.

Mitigations:

- Access control lists
- Digital signatures
- Message authentication codes
- Integrity controls

Repudiation

To perform illegal operations in a system that lacks the ability to trace the prohibited operations.

Mitigations:

- Strong authentication
- Secure logging and auditing
- Digital signatures
- Time stamps
- Trusted third party

Information disclosure

To read a file that one was not granted access to, or to read data in transit.

Mitigations:

- Protecting data in transit
- Protecting data at rest
- Access control lists
- Encryption

Denial of service

To deny access to valid users, such as by making a web server temporarily unavailable or unusable.

Mitigations:

- Access control lists
- Filtering
- Load balancing
- Authorization
- High-availability designs
- Quotas and rate limits

Elevation of privilege

To gain privileged access to resources for gaining unauthorized access to information or to compromise a system.

Mitigations:

- Input validation
- Setting permissions
- Group or role membership
- Access control lists

Security Controls/Mitigation

Input Validation

A user or client will not always submit data your application will expect. By building robust applications that do not trust user input by default, you ensure the application will be able to handle unexpected data gracefully.

Field Validation:

- Data Type Validation
- Required Validation
- Regular Expression Validation

Output Encoding

Encoding, closely related to Escaping is a powerful mechanism to help protect against many types of attack, especially injection attacks and Cross-site Scripting (XSS). Essentially, encoding involves translating special characters into some equivalent that is no longer significant in the target interpreter.

Treat content as content and not executable code.

Parameterized Queries

A parameterized query is a query in which placeholders are used for parameters and the parameter values are supplied at execution time.

Examples:

- Stored Procedures (without string concatenation)
- SQL Commands with SQL Parameters instead of string concatenation
- Data Access Frameworks that use parameterized queries

Still treating content as content and not executable code.

SQL Injection

Reasons not to stop SQL Injection:

- Laziness?
- Stubbornness?

As simple as using parameterized queries yet year after year injection is the number one threat.

Other Types of Injection

- Xpath Injection
- LDAP Injection
- NoSQL Injection
- OS Commands
- XML Parsers
- SMTP Headers

Code injection is the exploitation of a computer bug that is caused by processing invalid data. Injection is used by an attacker to introduce (or "inject") code into a vulnerable computer program and change the course of execution.

Request Tokens

- Characteristics of a Request Token
 - Unique per user & per user session
 - Tied to a single user session
 - Large random value
 - Generated by a cryptographically secure random number generator
- The request token is added as a hidden field for forms or within the URL if the state changing operation occurs via a GET
- The server rejects the requested action if the request token fails validation

Authentication

U.S. National Institute of Standards and Technology (NIST) Special Publication 800-53 (SP800-53) defines authentication as verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system.

Authorization

Ensuring that access to assets is authorized and restricted based on business and security requirements. Areas to apply Access Control Lists (ACLs):

- applications or APIs;
- specific application screens or functions;
- specific data fields;
- memory;
- private or protected variables;
- storage media;
- transmission media;
- In short, any object used in processing, storage or transmission of information.

Cookie Management

It's possible for an attacker to steal and reuse session identifiers or other sensitive cookie values when they are stored or transmitted insecurely.

It's possible for an attacker to tamper with session identifiers or other sensitive cookie values when they are stored or transmitted insecurely.

Protect and validate cookies.

Session Management

The disclosure, capture, prediction, brute force, or fixation of the session ID will lead to session hijacking (or sidejacking) attacks, where an attacker is able to fully impersonate a victim user in the web application.

Protect and validate session variables.

Configuration Management

There are a wide variety of server configuration problems that can plague the security of a site. These include:

- Unpatched security flaws in the server software
- Server software flaws or misconfigurations that permit directory listing and directory traversal attacks
- Unnecessary default, backup, or sample files, including scripts, applications, configuration files, and web pages
- Improper file and directory permissions
- Unnecessary services enabled, including content management and remote administration

Configuration Management Continued...

- Default accounts with their default passwords
- Administrative or debugging functions that are enabled or accessible
- Overly informative error messages
- Misconfigured SSL certificates and encryption settings
- Use of self-signed certificates to achieve authentication and man-in-the-middle protection
- Use of default certificates
- Improper authentication with external systems

Error Handling

An important aspect of secure application development is to prevent information leakage. Error messages give an attacker great insight into the inner workings of an application.

The purpose of reviewing the Error Handling code is to assure the application fails safely under all possible error conditions, expected and unexpected. No sensitive information is presented to the user when an error occurs.

Logging and Auditing

- Auditable – all activities that affect user state or balances are formally tracked
- Traceable – it's possible to determine where an activity occurs in all tiers of the application
- High integrity – logs cannot be overwritten or tampered with by local or remote users

Fail safe – do not fail open

Cryptography

Cryptography (or crypto) is one of the more advanced topics of information security

It is difficult to get right because there are many approaches to encryption

In addition, serious cryptography research is typically based in advanced mathematics and number theory, providing a serious barrier to entry.

The proper and accurate implementation of cryptography is extremely critical to its efficacy.

A small mistake in configuration or coding will result in removing a large degree of the protection it affords and rendering the crypto implementation useless against serious attacks.

Least Privilege

The principle of least privilege recommends that accounts have the least amount of privilege required to perform their business processes. This encompasses user rights, resource permissions such as CPU limits, memory, network, and file system permissions.

Resources

- Build Security In (DHS)
- Open Web Application Security Project (OWASP)
- Microsoft's Security Development Lifecycle (SDL)
- Microsoft Threat Modeling Tool 2016

Build Security In



Build Security In is a collaborative effort that provides practices, tools, guidelines, rules, principles, and other resources that software developers, architects, and security practitioners can use to build security into software in every phase of its development.

<https://buildsecurityin.us-cert.gov/>

Build Security In continued...

Consistent with this list is the Top 10 Project by the Open Web Application Security Project (OWASP). OWASP's report captures the top ten risks associated with the use of web applications in an enterprise.

OWASP

The Open Web Application Security Project (OWASP) is a 501(c)(3) worldwide not-for-profit charitable organization focused on improving the security of software. Our mission is to make software security visible, so that individuals and organizations worldwide can make informed decisions about true software security risks.

<https://www.owasp.org>

OWASP Top 10

The OWASP Top Ten is a powerful awareness document for web application security. The OWASP Top Ten represents a broad consensus about what the most critical web application security flaws are. Project members include a variety of security experts from around the world who have shared their expertise to produce this list.

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

Security Development Lifecycle (SDL) [Microsoft]

The Security Development Lifecycle (SDL) is a software development process that helps developers build more secure software and address security compliance requirements while reducing development cost

- <https://www.microsoft.com/en-us/sdl/>

Microsoft Threat Modeling Tool 2016

Microsoft Threat Modeling Tool 2016 is a tool that helps in finding threats in the design phase of software projects.

Microsoft Threat Modeling Tool 2016 Download:

<http://www.microsoft.com/en-us/download/details.aspx?id=49168>