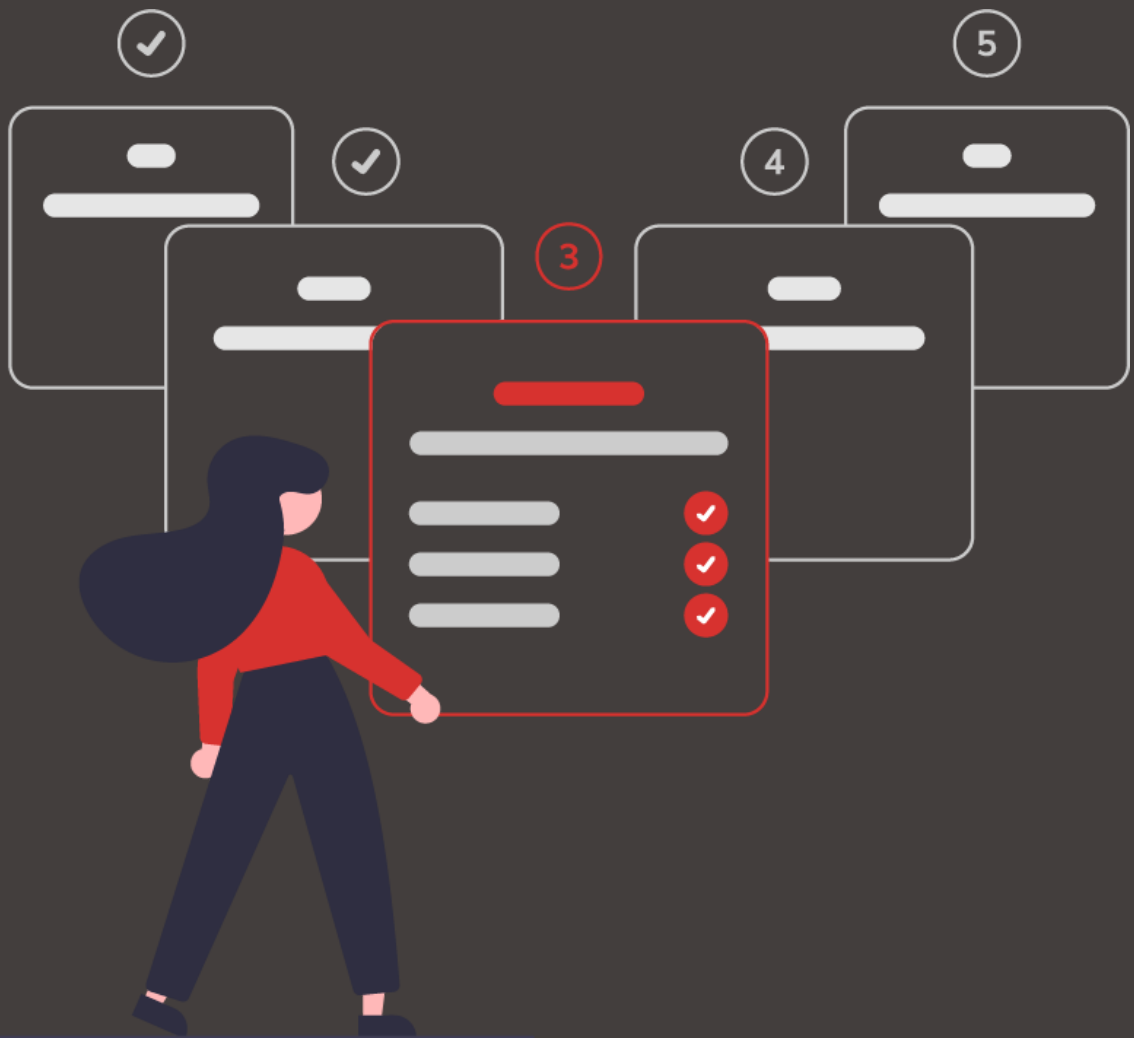
ALTERSOLUTIONS
DEUTSCHLAND

Security Threat Modeling

Eine Einführung



Einleitung

Security Threat Modeling (kurz: STM), ist, wie die Übersetzung des Namens schon sagt, eine Analyse von Bedrohungen bzgl. der IT-Sicherheit gegenüber einem System, einer Applikation, einer Schnittstelle, einem Betriebsmodell (Cloud vs. OnPrem), etc. Diese Art der Bedrohungsanalyse ist dabei nicht ausschließlich auf IT-Systeme beschränkt, sondern immer dann relevant, wenn Unternehmen und Organisationen es mit schützenswerten Assets zu tun haben.

Ein simples Beispiel außerhalb der IT-Welt: Wenn ein Wohnungseigentümer einen Rauchmelder in seinem Wohnzimmer angebracht hat, der bei Rauchentwicklung Alarm schlagen soll, hat der Wohnungseigentümer die Bedrohung einer Rauchgasvergiftung durch einen Brand „analysiert“ und mit dem Rauchmelder diese Bedrohung „mitigiert“.

Übertragen auf die Welt der Informationstechnologie bedeutet es, dass mögliche Bedrohungen zu einem IT-System analysiert und mitigiert werden. Dies geschieht am besten bereits vor der Entwicklung des Systems. In unserem Beispiel mit der Rauchentwicklung bzw. dem Wohnungsbrand wären z.B. bei dem Bau der Wohnung auf schwer entzündbare Materialien zu achten, eine mögliche „Mitigierung“. Das Wohnungsbeispiel verdeutlicht auch, dass nach dem Bau die Bedrohung u.U. nicht mehr abgewendet werden kann.

Halten wir also fest: Eine Bedrohungsanalyse sollte ein zentraler Bestandteil des Software-Entwicklungsprozesses sein, damit erkannte Bedrohungen schon während des „Baus“ des Systems adressiert werden können.

Warum Security Threat Modeling?

Wenn wir bei unserem Wohnungsbeispiel bleiben: Früh getroffene Entscheidungen können sich drastisch auf die Sicherheit der Immobilie auswirken. Natürlich kann (und sollte) man immer noch Rauchmelder installieren – aber wäre es nicht besser, wenn man von Vornherein auf schwer entflammbare Baumaterialen setzt?

Eine Bedrohungsanalyse kann ungünstige oder falsche IT-Architekturentscheidungen vermeiden, und zwar schon bevor eine Line of Code erzeugt wurde. Außerdem kann eine Bedrohungsanalyse hilfreich sein, um Sicherheits-Anforderungen an das System zu verstehen und diese zu ergänzen.

Ein Beispiel aus der Welt der IT: Müssen die Daten verschlüsselt werden? Vielleicht nicht unbedingt auf einem Server, aber bei Nutzung auf einem mobilen Endgerät schon eher. Wie wir sehen, existiert also eine enge Abhängigkeit zwischen Bedrohungsszenario, Maßnahmen zur Vermeidung (Mitigation) und System-Anforderungen. Durch die Betrachtung der (Sicherheits-) Anforderungen, bereits im frühen Solution Design, kann ein viel sichereres System bereitgestellt werden.

Wie führt man Security Threat Modeling in der Praxis durch?

Einfach gesagt ist Security Threat Modeling nichts anderes als die frühzeitige Überlegung, was alles an welcher Stelle in Bezug auf IT-Sicherheit passieren kann und das entsprechende Definieren einer darauf basierenden Schutzstrategie, um das Worst-Case-Szenario zu vermeiden. Dafür werden folgende Fragen beantwortet:

1. Was wird entwickelt?
2. Was kann passieren?
3. Was kann dagegen getan werden?
4. Wie wird die Gesamtlage bewertet?
5. Wie war die Qualität der Analyse?

Jede dieser fünf Fragen ist als ein Prozessschritt der Bedrohungsanalyse zu betrachten.

Schritt 1: Was wird entwickelt?

Ein gängiger Weg ein IT-System zu beschreiben, sind technische Diagramme. Ideal für unseren Kontext sind Datenfluss-Diagramme. In so einem Diagramm werden alle Schnittstellen und Datenflüsse visualisiert. Denn nur wo Daten vorhanden und/oder „fließen“, entstehen Bedrohungen. Als Beispiel schauen wir uns eine „typische“ Webanwendung an:
Unsere Anwendung wird in einem Rechenzentrum betrieben und besteht aus einem Proxy, einer Web Application und einer Datenbank. Natürlich hat jede Webanwendung auch mindestens einen Actor, nämlich den User, der über das Internet Zugriff auf den Proxy und damit auf Web Application und die Datenbank hat.

Für das Erstellen des Datenfluss-Diagramms sollten neben dem verantwortlichen Architekten auch immer ein oder mehrere Entwickler und System-Admins konsultiert werden. Nur so entsteht am Ende dieses Schrittes ein Diagramm, das in der Meinung aller Beteiligten korrekt ist – und ein korrektes Diagramm ist elementar wichtig für den weiteren Prozess des Security Threat Modeling.

Schritt 2: Was kann passieren?

Nachdem das Datenfluss-Diagramm des Systems erstellt worden ist, überlegt man sich im nächsten Schritt, was alles passieren kann. Hierbei ist es hilfreich, Annahmen zu treffen, um etwaige Bedrohungsszenarien einzugrenzen.

Beispiel: Wenn man das System-Beispiel aus Schritt 1 betrachtet, können durch die Annahme, dass ein aktuelles Betriebssystem verwendet wird, einige Bedrohungen von vornherein ausgeschlossen werden. Aber Achtung: Solche Annahmen sollten auch stets überprüft und bei der Bedrohungsanalyse berücksichtigt werden.

Einige Fragen sind bzgl. „was kann passieren“ offensichtlich:

1. Ist der Web-Server wirklich der, als der er sich zu erkennen gibt?
2. Können die Daten zwischen Proxy und Web Application manipuliert werden?
3. Wie stellt die Web Application sicher, dass die Datenbank die Daten auch erhalten hat?
4. Kann ein Benutzer durch die Web Application auch die Architektur dahinter sehen?
5. Kann die Web Application (oder die Datenbank) durch zu hohe Last in die Knie gezwungen werden?
6. Kann ein User in Bezug auf die Web Application Dinge tun, zu denen er nicht berechtigt ist?

Diese Liste kann man natürlich nahezu beliebig fortgesetzt werden. Entsprechend schnell kann es passieren, dass wichtige Dinge übersehen werden. Daher sollte eine Methodik bei diesen Fragen benutzt werden. Wir setzen hier auf die sogenannte STRIDE-Methode, die von Microsoft entwickelt wurde. Dabei steht STRIDE für:

S: Spoofing
T: Tampering
R: Repudiation
I: Information Disclosure
D: Denial of Service
E: Elevation of Privilege

Mit dieser Methode können wir unsere „Was kann passieren?“-Fragen kategorisieren und durch ein strukturiertes Vorgehen die Gefahr reduzieren, dass etwas übersehen wird. Genau betrachtet kann jede, oben als Beispiel aufgeführte, Frage diesen Kategorien zugeordnet werden.

Schritt 3: Was kann dagegen getan werden?

Am Ende der „Was kann passieren?“-Frage sollte eine Liste von Bedrohungen zu allen, im Datenfluss-Diagramm aufgeführten, Interaktionen und Daten vorliegen. Im nächsten Schritt gehen wir die Liste durch und adressieren alle gefundenen Bedrohungen. Das geht auf vier verschiedene Arten:

Mitigation bedeutet, Maßnahmen zu ergreifen, um eine Bedrohung abzuschwächen bzw. es schwieriger zu machen, dass eine Bedrohung ausgenutzt wird.
Beispiel: Passwörter für eine administrative Schnittstelle mitigieren die Bedrohung, dass sich jemand vermeintlich als Admin ausgeben könnte.

Eliminierung erreicht man häufig durch Ablehnung von Funktionalitäten.
Beispiel: Man kann die administrative Schnittstelle mit Passwörtern schützen, aber die Bedrohung, dass sich jemand als Admin ausgeben könnte, ist immer noch vorhanden. Eine Deaktivierung der administrativen Schnittstelle würde diese Bedrohung verschwinden lassen.

Transfer erreicht man, wenn man die Bedrohung auf jemand anderen transferiert.
Beispiel: Man könnte die Authentifizierung für die administrative Schnittstelle einem Active Directory Service überlassen.

Akzeptieren ist, wenn man entscheidet, nichts gegen eine identifizierte Bedrohung zu unternehmen. Dies kann unterschiedliche und auch durchaus valide Gründe haben. Häufig sind das auch Kosten-Nutzen Entscheidungen.
Nun hat man eine Liste von Bedrohungen und unterschiedliche Strategien zur Adressierung dieser Bedrohungen. Im nächsten Schritt müssen die Mitigationen bewertet und priorisiert werden. Die dann ausgewählten Mitigationen werden in den Softwareentwicklungsprozess aufgenommen. Hierzu kann man z.B. User-Stories oder auch einfach Bugs verwenden.
Beispiel: Ein Hacker kann durch zu viele automatisierte Anfragen die Datenbank zum Absturz bringen.

Am Ende dieser Aktivität hat man aller Voraussicht nach eine Menge To-Dos, die umgesetzt werden müssen. Doch vor der Umsetzung bewerten wir nun alle Bedrohungen und Maßnahmen, um eine Kosten-Nutzen-Analyse durchzuführen.

Schritt 4: Wie wird die Gesamtlage bewertet?

Nach den vorherigen beiden Aktivitäten hat man eine Menge an Bedrohungen und möglichen Gegenmaßnahmen identifiziert. Jetzt geht es darum, diese Bedrohungen bzw. die Gegenmaßnahmen zu priorisieren. Hierzu bestimmt man für jede Bedrohung das sogenannte Bedrohungsrisiko.
Eine Möglichkeit zur Bestimmung ist folgende Formel:

Bedrohungsrisiko = Eintrittswahrscheinlichkeit X Auswirkung

Hier gibt es unterschiedliche Ansätze, wie man solche Bedrohungsrisiken bewertet, z.B.: Microsoft Bug Bar oder CVSS (Common Vulnerability Scoring System).

In größeren Unternehmen wird dies häufig auch in den Security Policies des Unternehmens festgehalten. Um es simpel zu halten, kann man die Eintrittswahrscheinlichkeit und die Auswirkung in vier gängige Kategorien der Risikobewertung einordnen: *Sehr hoch*, *Hoch*, *Mittel*, *Niedrig*

Die Möglichkeiten zur Bewertung der Bedrohung sind sehr vielfältig. Im ersten Schritt ist nur wichtig, ein abgestimmtes Vorgehen zur Bewertung der Bedrohungen zu nutzen, das dauerhaft und einheitlich angewandt wird.

Nachdem alle Bedrohungen ohne Gegenmaßnahmen bewertet werden, sollte man diese Bedrohungen mit Gegenmaßnahme(n) erneut bewerten. Letzteres dient aber eher zum Zwecke der Dokumentation und ist nützlich, um zu einem späterem Zeitpunkt Entscheidungen nachzuvollziehen.

Schritt 5: Wie war die Qualität der Analyse?

Der letzte Schritt im Security Threat Modeling ist die Qualitätsbewertung der Bedrohungsanalyse.

Überprüfung der Architektur
Zuerst geht man zurück zu seinem Datenfluss-Diagramm und überprüft, ob es noch korrekt ist. Es ist nicht selten, dass Architekturentscheidungen kurzfristig geändert werden: Funktionen kommen hinzu, vorher nicht bedachte Use-Cases bringen andere Datensätze mit sich, Interaktionen zwischen Systemen ändern sich. Also muss das Datenfluss-Diagramm aktualisiert bzw. ergänzt werden.

Überprüfung der Bedrohungen
Natürlich müssen neue Bedrohungen, die durch eine geänderte Architektur entstehen, identifiziert werden. Danach sollten noch einmal alle identifizierten Bedrohungen hinsichtlich ihrer Mitigationen überprüft werden. Dafür müssen wir die Liste der Bedrohungen erneut durchgehen und dabei folgende Fragen stellen: Wurde jede einzelne Bedrohung adressiert?Wurde jede einzelne Bedrohung richtig mitigiert?

Tests für die Maßnahmen
Für alle Maßnahmen sollten Tests durchgeführt werden. Diese können manuell oder auch automatisiert exekutiert werden. Es muss sichergestellt werden, dass alle Maßnahmen, die umgesetzt werden sollen, auch qualitätsgesichert werden. Idealerweise sollten diese Tests in ein bereits vorhandenes Testing-Framework bei der Entwicklung integriert werden

Fazit

Natürlich ist ein „reales“ Security Threat Modeling noch um einiges detaillierter und aufwändiger als der hier skizzierte Ablauf. Es sollte dennoch deutlich geworden sein, wie wichtig es ist, Sicherheit bereits konzeptionell mit in den Entwicklungsprozess zu integrieren und in regelmäßigen Abständen – bspw. bei der Entwicklung neuer Features oder der Änderung des Betriebsmodells - zu aktualisieren Die hier gezeigte Analyse-Methode mit ihren Mitigations- und Kosten-Nutzen-Überlegungen sorgt für mehr Sicherheit von Beginn an.

Alter Solutions Deutschland GmbH
Klaus-Bungert-Str. 6a
40468 Düsseldorf
Tel.: +49 211 54 56 019-0
E-Mail: digitial@alter-solutions.com

ALTER SOLUTIONS
DEUTSCHLAND