Schnittstellen als Einfallstor in die interne IT?

WIE BEDROHUNGS-MODELLIERUNG DIE SICHERE IMPLEMENTIERUNG VON SCHNITTSTELLEN FÖRDERT

Bei einem Penetrationstest eines Onlineshops zeigte sich ein alarmierendes Bild: Das System bot nicht nur direkte Angriffsflächen, sondern ermöglichte über unzureichend gesicherte Schnittstellen auch den Zugriff auf interne Enterprise Resource Planning (ERP) und Business-Intelligence-(BI)-Systeme. Threat Modeling kann solche Schwachstellen frühzeitig identifizieren und absichern.

Oftmals sind Webanwendungen nur ein Nebengeschäft oder dienen einem Marketingzweck, sei es der Onlineshop als Ergänzung zum stationären Handel oder die Webseite als digitale Visitenkarte. In der Praxis werden solche Systeme daher häufig als Projekt betrachtet, das nach dem Go-live abgeschlossen ist. Die zur Verfügung stehenden finanziellen und personellen Ressourcen werden anschließend weitestgehend zurückgefahren. Selten gibt es einen Pentest oder einen Wartungsvertrag, der alle Sicherheitsaspekte adäquat abdeckt. So geht von der einst mühevoll aufgebauten Webanwendung schnell ein hohes Sicherheitsrisiko aus.

Besonders kritisch wird es, wenn solche Systeme über Schnittstellen mit internen Anwendungen verbunden sind. Was beim isolierten Website-Baukasten eines Massenhosters für manche Betreiber vertretbar wirkt, wird beispielsweise bei einem Onlineshop schnell zum Potenzial für Reputationsschäden und Datenschutzverletzungen:

- personenbezogene Daten von Kunden
- sensitive Unternehmensinformationen wie Umsätze oder Kostenstruktur
- Schnittstellen zur Übertragung von Bestellungen in das ERP

HERAUSFORDERUNGEN BEI DER SCHNITTSTELLENENTWICKLUNG

Es liegt in der Natur der Sache, dass bei der Implementierung von Schnittstellen zwei unterschiedliche Domänen auf einen gemeinsamen Nenner gebracht werden müssen. In der Idealvorstellung sind die beteiligten Personen mit beiden Domänen vertraut und wissen schon in der Konzeptionsphase, worauf es zu achten gilt und welche Besonderheiten die Systeme haben. In der Praxis häufiger anzutreffen ist jedoch einseitiges Domänenwissen: Während die Onlineshop-Agentur bestens mit dem Shopsystem vertraut ist, kennt der ERP-Berater jeden Winkel seines ERP-Systems. Sind dann noch unterschiedliche Infrastrukturbetreiber wie die interne IT für die demilitarisierte Zone (DMZ) und der Hoster für die Webanwendung beteiligt, steigen Kommunikations-Overhead, Komplexität und Fehleranfälligkeit weiter an.

Die Entwicklung von Schnittstellen wird dadurch häufig zur Kompromisslösung: Man beschäftigt sich genauso viel (oder wenig) mit dem anderen System, wie es für die Lösung der konkreten Aufgabe notwendig ist. Das kann zu zahlreichen Problemen und Angriffsvektoren führen, unter anderem:

- falsch konfigurierte Zugriffsberechtigungen
- Systemausfälle aufgrund zu hoher Schnittstellenlast
- unnötig offene Ports in die DMZ
- Cross-Site-Scripting oder andere Injections aufgrund von Unterschieden bei der Datenvalidierung und -bereinigung
- Datenlecks durch fälschlicherweise offene Endpunkte

Häufig liegen die Ursachen hierfür nicht in mangelnden technischen Fähigkeiten, sondern in fehlendem Systemverständnis, Missverständnissen bei der Kommunikation oder unklaren Zuständigkeiten.

BEDROHUNGEN SYSTEMATISCH IDENTIFIZIEREN

Wann immer es um Risiken geht, lautet die Standardantwort: Risikomanagement nach dem Risk-Management-Lifecycle (Risikoidentifizierung, Risikobewertung, Risikobehandlung und Risikokontrolle). Für den gezielten Umgang mit Bedrohungen existiert jedoch eine spezifischere Methodik: das Threat Modeling.

Dabei handelt es sich um einen strukturierten Ansatz, um Bedrohungen zu identifizieren, zu verstehen und zu behandeln. Die „Open Web Application Security Project (OWASP)"-Foundation definiert mit dem „Four Questions Framework" einen strukturierten Ansatz für den Aufbau eines Bedrohungsmodells:

- Umfang und Kontext definieren: „Woran arbeiten wir?"
- Bedrohungen identifizieren: „Was kann schiefgehen?"
- Maßnahmen festlegen: „Was werden wir dagegen tun?"
- Maßnahmen bewerten: „Haben wir gute Arbeit geleistet?"

Bedrohungen entstehen besonders beim Datenaustausch – einschließlich Metadaten wie HTTP-Request-Informationen. Daher bilden Datenflussdiagramme eine gute Basis für den Aufbau eines Threat Models. In einem oberflächlichen Diagramm lassen sich grundlegend die beteiligten Prozesse, Entitäten und Datenspeicher ablesen (siehe Abbildung 1).

Vertrauensstufen (Trust Levels) definieren die Berechtigungen von Entitäten und Prozessen. Ein Onlineshop läuft beispielsweise als www-data-Systembenutzer, während das ERP-System einen

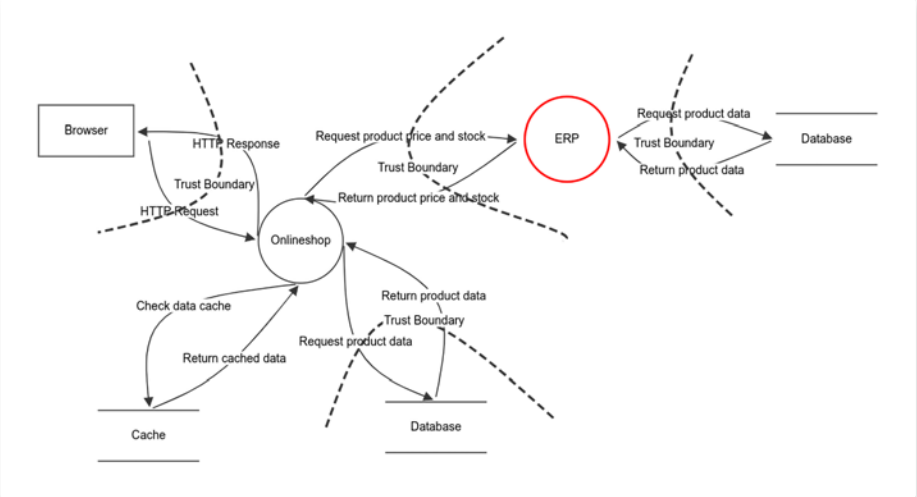

Abbildung 1: Darstellung eines beispielhaften und einfach gehaltenen Threat Models einer Schnittstelle zwischen Onlineshop und ERP aus Threat Dragon (https://owasp.org/www-project-threat-dragon/). (Bild: Dev Specialists GmbH)

Bild: ImageFlow | Maryna - stock.adobe.com

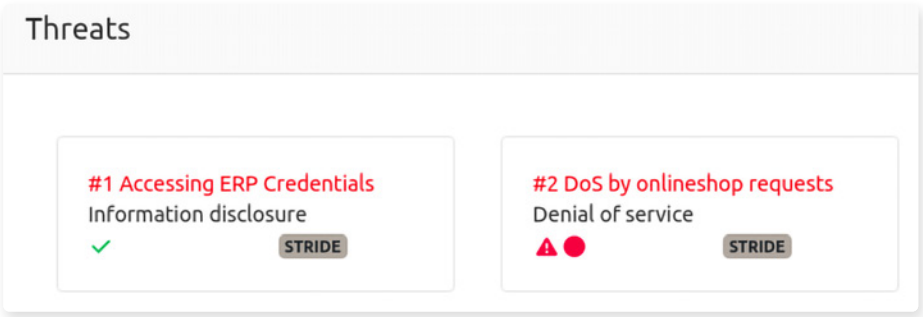

Abbildung 2: Darstellung von beispielhaften Risiken am ERP-Prozess aus Threat Dragon. (Bild: Dev Specialists GmbH)

speziellen Schnittstellenbenutzer benötigt. Ändern sich die Trust Levels, wird dies über Vertrauensgrenzen (Trust Boundaries) kenntlich gemacht, die dadurch kritische Datenflüsse kennzeichnen.

Auf Basis von Klassifikationsmodellen wie STRIDE oder LINDDUN lassen sich nun spezifische Bedrohungen für Entitäten, Prozesse und Datenspeicher definieren (siehe Abbildung 2). In detaillierteren Modellen können auch komplexere Architekturen wie Microservices abgebildet werden.

PRAKTISCHER NUTZEN VON THREAT MODELS

Ein Threat Model vereint Infrastruktur, Architektur und Datenflüsse und ermöglicht die systematische Klassifikation von Bedrohungen. Als Anbieter einer Schnittstelle, die von Dritten genutzt und implementiert wird, lassen sich Risiken sowohl in einem allgemeinen als auch in einem spezifischen Kontext – beispielsweise endpunktbezogen – modellieren und Vorgaben zur Mitigation ableiten.

Eine typische Bedrohung ergibt sich aus dem Umgang mit den Zugangsdaten für Schnittstellen. Gerade in Onlineshops und ähnlichen Anwendungen sind diese unverschlüsselt in Konfigurationsdateien, in der Datenbank oder leider auch hartkodiert im Quelltext zu finden. Klare Vorgaben zur Risikominderung, die bereits im Threat-Modeling-Prozess definiert werden, können hier Abhilfe schaffen.

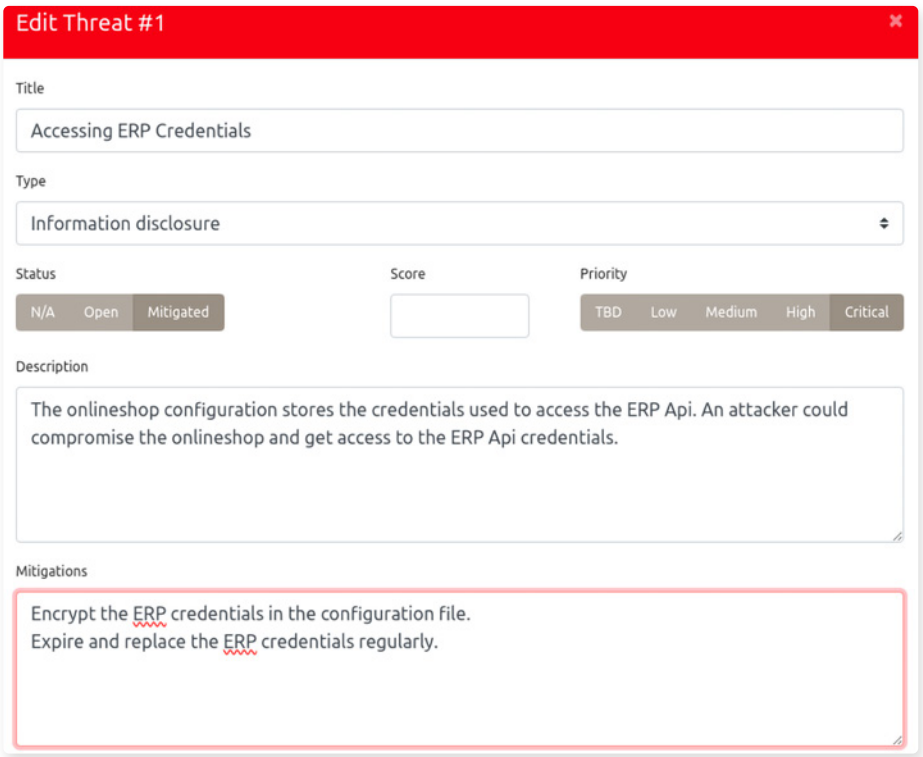

Abbildung 3: Bearbeitungsfenster eines beispielhaften Risikos in Threat Dragon. (Bild: Dev Specialists GmbH)

Anstatt nur auf Anbietervorgaben zu setzen, können Implementierer Bedrohungen auch eigenverantwortlich adressieren. Das Threat Model hilft hier bei der Schaffung von Awareness für die Besonderheiten des Systems, als Fahrplan zur Schnittstellenabsicherung und zur Dokumentation der Mitigation.

Grundsätzlich ist jedoch zu beachten, dass die Menge an Bedrohungen, die durch den Implementierer mitigiert werden müssen, minimal zu halten ist. Threat Models dürfen nicht zum Instrument werden, die eigene Verantwortung an Dritte abzuwälzen. Eine Delegation an den Entwickler darf nur bei Bedrohungen erfolgen, bei denen der Anbieter keine andere Mitigationsmöglichkeit hat.

Auch mit einem umfassenden Bedrohungsmodell bleiben externe Sicherheitstests unverzichtbar. Die implementierten Schutzmaßnahmen müssen überprüft werden, und die unabhängige Perspektive kann neue Bedrohungen aufdecken, die das Modell weiter verbessern.

FAZIT: MEHR ALS EIN PAPIERTIGER

Threat Models erscheinen zunächst als zusätzlicher bürokratischer Aufwand. Richtig eingesetzt entwickeln sie sich jedoch zu wirksamen Werkzeugen für bessere Anwendungssicherheit. Besonders für externe Entwickler bieten sie erhebliche Vorteile: Sie schaffen Systemverständnis, liefern Handlungsempfehlungen und definieren klare Anforderungen. So verhindern sie, dass Schnittstellen zum Einfallstor in die interne IT werden. ■

DOMINIK STRAUSS
ist CEO und Senior Solutions Architect bei der Dev Specialists GmbH