Bezpieczeństwo w inżynierii oprogramowania

Modelowanie zagrożeń

Arkadiusz Chrobot, Adam Malicki

Katedra Systemów Informatycznych

26 października 2024

Plan

Wstęp

Modelowanie zagrożeń (ang. *Threat Modelling*) jest zbiorem ustrukturyzowanych, powtarzalnych metod, które pozwalają w sposób racjonalny podejmować decyzje dotyczące zabezpieczenia oprogramowania i systemów [2]. Metody te pozwalają aktywnie eliminować podatności w oprogramowaniu, poznać i zrozumieć wymagania związane z bezpieczeństwem, ograniczyć ryzyko, projektować i dostarczać lepsze produkty oraz symulować potencjalne ataki. Stosowanie modelowania zagrożeń staje się nieodzownym elementem współczesnych ▸ standardów bezpieczeństwa. Z tych powodów każdy nowoczesny inżynier oprogramowania powinien dysponować nawet podstawową wiedzą na temat tego procesu.

Definicje

Zagrożenie

Zagrożeniem jest każde zdarzenie, które ma niekorzystny wpływ na oprogramowanie, system i zasoby [1]. Jego wystąpienie wiąże się ze *stratą*, która ma najczęściej charakter finansowy.

Atak

Wykorzystanie podatności do celowego spowodowania wystąpienia zagrożenia nazywa się atakiem [1].

Ryzyko

To iloczyn prawdopodobieństwa materializacji zagrożenia i miary jego skutków (np. nieistotne, małe, średnie, duże, katastrofalne) [1].

Definicje

Modelowanie zagrożeń

Identyfikację typów ataków, które potencjalny intruz może wykorzystać do naruszenia zabezpieczeń oprogramowania lub systemu nazywa się modelowaniem zagrożeń. Jest ono najczęściej wykonywane przez inżynierów i/lub ekspertów od bezpieczeństwa [2].

Zarządzanie ryzykiem

Dla kontrastu, decydowanie o sposobach radzenia sobie ze zidentyfikowanymi problemami i zagrożeniami oraz podejmowanie niezbędnych kompromisów nazywa się zarządzaniem ryzykiem. Za tę czynność są najczęściej odpowiedzialni prawnicy i personel zarządzający [2].

Przykłady zagrożeń

Do zagrożeń można zaliczyć:

- awarie oprogramowania, czasową niedostępność systemu, błędy administratorów, użytkowników, utrzymania;
- przeciążenie systemu w skutek braku zabezpieczeń przed atakami DoS/DDoS, braku monitorowania wykorzystania zasobów, braku planów modernizacji infrastruktury;
- ograniczona dostępność personelu i brak możliwości utrzymania systemu na skutek rotacji pracowników, chorób, niedoborów osób o wymaganych kwalifikacjach;
- nieautoryzowany dostęp do przesyłanych danych, kradzież nośników z poufnymi danymi, brak procedur niszczenia nośników danych;
- włamania do systemów lub baz danych, kradzież danych, phishing, nieautoryzowany dostęp do sieci.

Podstawy modelowania zagrożeń

Istnieje wiele metod modelowania zagrożeń [1, 3, 4]. Każda z nich dostosowana jest do określonego celu, takiego jak np. ochrona prywatności, infrastruktury lub identyfikacja i analiza potencjalnych przeciwników. Jednak fundamentem wszystkich tych metod i całego procesu modelowania zagrożeń są cztery pytania [5] zaproponowane przez Adama Shostacka [2], które stały się także podstawą opublikowanego w 2020 roku *Manifestu Modelowania Zagrożeń* [6]:

1. Nad czym pracujemy? (ang. *What are we working on?*)
2. Co złego może się wydarzyć? (ang. *What can go wrong?*)
3. Co z tym zrobimy? (ang. *What are we going to do about it?*)
4. Czy wykonaliśmy dostatecznie dobrze nasze zadanie? (ang. *Did we do a good enough job?*)

Podstawy modelowania zagrożeń

Pierwsze pytanie dotyczy struktury i dziedziny zastosowania oprogramowania, dla którego jest przeprowadzane modelowanie zagrożeń. Jego celem jest identyfikacja *aktywów*, czyli najcenniejszych zasobów, *granic zaufania*, czyli interfejsów między komponentami o różnym stopniu zaufania oraz *powierzchni i wektorów ataku*. Do uzyskania na nie odpowiedzi niezbędne są dokumenty projektowe, takie jak diagramy UML, diagramy przepływu danych, opisy protokołów komunikacyjnych.
Drugie pytanie ma na celu identyfikację potencjalnych zagrożeń, poprzez określenie, kto może być potencjalnym intruzem, poznanie zaprojektowanego sposobu działania oprogramowania, zbadanie czy istnieje możliwość zmuszenia go do zadziałania niezgodnie z intencjami projektantów oraz ustalenie, czy takie zachowanie może być uzyskane w sposób intencjonalny lub przypadkowy.

Podstawy modelowania zagrożeń

Trzecie pytanie dotyczy środków zaradczych, które powinny być zastosowane wobec zidentyfikowanych zagrożeń. Ogólnie, możliwe działania można podzielić na cztery kategorie:

Łagodzenie zagożeń (ang. *mitigating threats*) — uczynienie zagrożeń trudniejszymi do wykorzystania przez intruza,

Eliminowanie zagrożeń (ang. *eliminating threats*) — usunięcie interfejsów lub komponentów, które tworzą zagrożenie,

Przeniesienie zagrożeń (ang. *transferring threats*) — przekazanie odpowiedzialności za przeciwdziałanie jednostkom, które dysponują lepszymi informacjami w tym zakresie, np. uczynienie klienta odpowiedzialnym za dostosowanie konfiguracji oprogramowania do jego infrastruktury,

Akceptacja zagrożenia (ang. *accepting threats*) — możliwa, jeśli czas i wysiłek konieczne do złagodzenia lub wyeliminowania zagrożenia mogłyby zagrozić realizacji głównego celu projektu.

Podstawy modelowania zagrożeń

Do środków zaradczych można zaliczyć:

- tworzenie kopii zapasowych,
- zapewnianie redundancji i środowiska odtworzeniowego (ang. *Disaster Recovery — (DRC)*),
- wdrożenie polityki haseł,
- dedykowane uprawnienia dla określonej grupy użytkowników oraz skuteczny mechanizm autoryzacji,
- monitorowanie obciążenia zasobów,
- testowanie zmian w odpowiednich środowiskach, zanim trafią do środowiska produkcyjnego,
- szyfrowanie przesyłanych i przechowywanych danych wrażliwych.

Podstawy modelowania zagrożeń

Czwarte pytanie ma na celu zainicjowanie w procesie modelowania zagrożeń retrospekcji zmierzającej do oceny jakości wykonanej pracy. W szczególności należy upewnić się, że uzyskany model odpowiada tworzonemu produktowi, posiłkując się w tym działaniu diagramami i innymi dokumentami projektowymi. Następnie trzeba sprawdzić, czy każde odpowiednio poważne zagrożenie został zidentyfikowane i czy zastosowano dla niego odpowiednie środki zapobiegawcze. Warto też zadbać, aby powstał odpowiedni zbiór automatycznych lub manualnych testów, które pozwolą zweryfikować, czy problem może wystąpić i czy przyjęte mechanizmy zabezpieczające są poprawnie zastosowane.
Modelowanie zagrożeń ma charakter cykliczny. Nie powinno być wykonywane jednokrotnie, ale zawsze wtedy, gdy wprowadzane są nawet drobne zmiany w oprogramowaniu lub zmienia się kontekst jego użytkowania.

Przykład

W przykładzie udostępnionym przez organizację *SAFECode* przedstawiono proces modelowania zagrożeń dla bliżej nieokreślonej aplikacji internetowej, w którym wykorzystano metodologię STRIDE, ułatwiającą identyfikację zagrożeń [7]. Jako podstawę do przeprowadzenia modelowania użyto diagramu przepływu danych przedstawionego na Rysunku 1. Na tym diagramie zaznaczono także znalezione granice zaufania oraz podano przykładowe zagrożeniami dla poszczególnych komponentów.

Przykłady

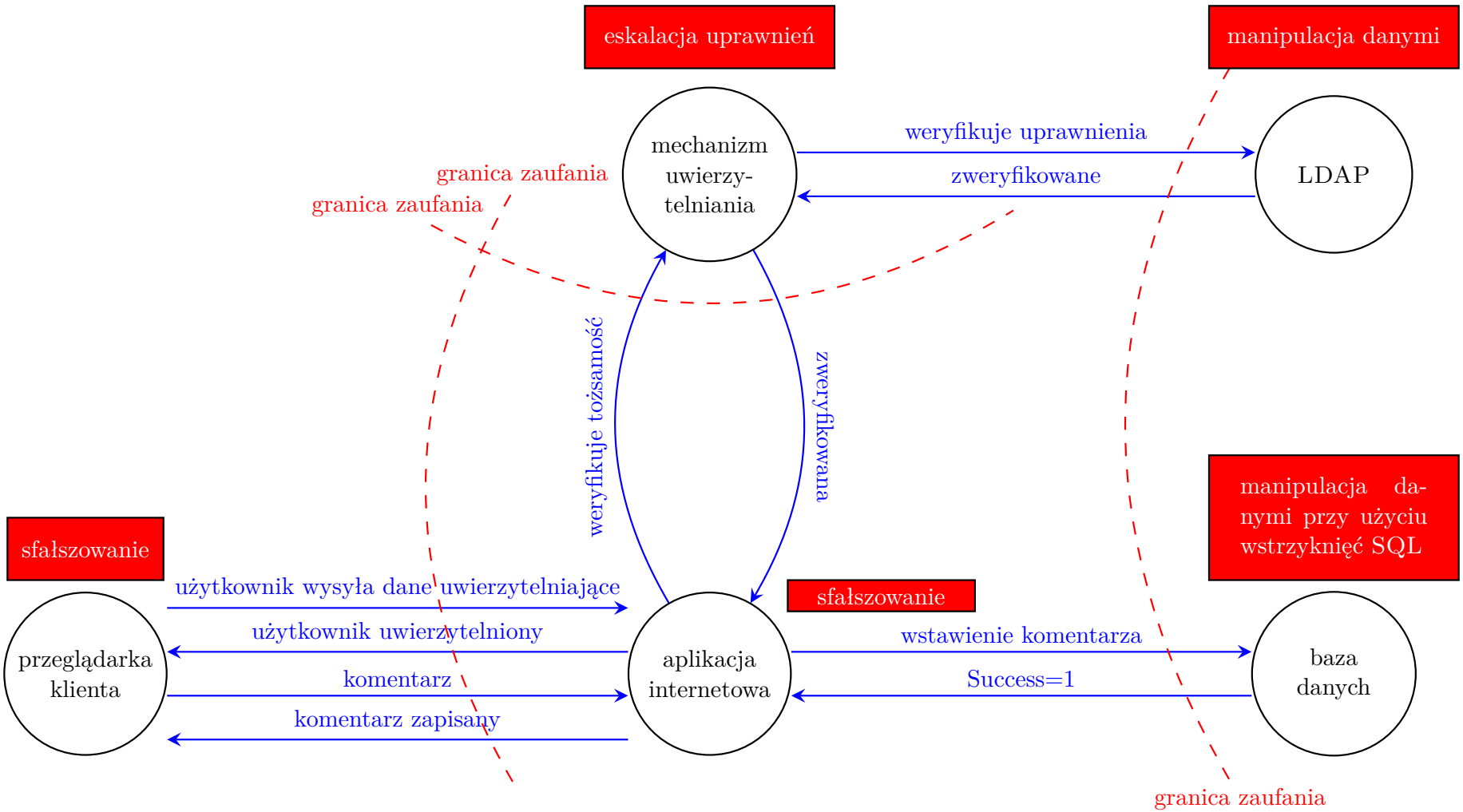

Diagram przepływu danych (ang. *Data Flow Diagram*) w modelowaniu zagrożeń (źródło: [7])

Spoofing

Intruz może podszyć się pod aplikację internetową (ang. *spoof*), co pozwoli mu na nieautoryzowany dostęp do przeglądarki użytkownika. *W trakcie dyskusji z projektantami ustalono, że wydaje się to niemożliwym, z uwagi na zastosowany mechanizm autoryzacji serwera.*

Intruz może podszyć się pod bazę danych, co będzie skutkowało tym, że zapisywane dane będą trafiać do niego, a nie do uprawnionego magazynu danych. *Po naradzie z projektantami ustalono, że jest to bardzo mało prawdopodobne, bo połączenie z bazą danych jest uwierzytelniane.*

Tempering

Wstrzyknięcie SQL, to atak polegający na dodaniu złośliwego (ang. *malicious*) kodu do ciągów znaków, które następnie są przekazane do instancji serwera bazy danych celem przetworzenia (ang. *parsing*) i wykonania.
Wstrzyknięcie LDAP jest możliwe ponieważ część procesu uwierzytelniania użytkownika obejmuje zapytanie do LDAP.
Intruz może sfałszować (ang. *temper*) dane przesyłane między komponentami aplikacji internetowej. Brak odpowiedniej walidacji danych jest główną przyczyną wielu podatności. Rozważ wszystkie ścieżki przepływu danych i to w jaki sposób te dane są przetwarzane. Sprawdź, czy dane są weryfikowane pod względem poprawności przy zastosowaniu metody *białej listy*.

Repudiation

Nie ma możliwości weryfikacji czy baza danych zapisała dane pochodzące od komponentów znajdujących się za granicą zaufania. Rozważ rejestrowanie danych na potrzeby audytu, obejmujące źródło, czas i podsumowanie (np. skrót) otrzymywanych danych.
Nie ma możliwości weryfikacji czy aplikacja internetowa otrzymała dane od procesu znajdującego się za granicą zaufania. Rozważ rejestrowanie danych na potrzeby audytu, obejmujące źródło, czas i podsumowanie (np. skrót) otrzymywanych danych.
Nie ma możliwości weryfikacji czy LDAP zapisał dane pochodzące od komponentów znajdujących się poza granicą zaufania. Rozważ rejestrowanie danych na potrzeby audytu, obejmujące źródło, czas i podsumowanie (np. skrót) otrzymywanych danych.

Information disclosure

Dane przesyłane między komponentami aplikacji internetowej mogą zostać podsłuchane (ang. *sniffed*) przez intruza. Ujawnione dane mogą posłużyć intruzowi do atakowania innych komponentów oprogramowania lub mogą stanowić wyciek wrażliwych informacji i być dowodem na brak zgodności systemu ze standardami bezpieczeństwa. Rozważ szyfrowanie przesyłanych danych.
Niewłaściwe zabezpieczenia LDAP mogą pozwolić intruzowi na odczyt danych wrażliwych. Wykonaj przegląd ustawień autoryzacji.

Denial of service

Przeglądarka klienta może ulec awarii, zatrzymaniu lub może działać powoli, prowadząc do naruszenia ustalonych metryk dostępności. Zewnętrzny czynnik (intencjonalny lub nieintencjonalny) blokuje dostęp do danych po drugiej stronie granicy zaufania.

Elevation of privilege

Przeglądarka klienta może zostać użyta do przejęcia kontekstu aplikacji internetowej, celem uzyskania dodatkowych uprawnień.
Intruz może przesłać dane przeglądarki klienta celem takiej zmiany działania tego programu, aby pracował na jego korzyść.

Przykład — podsumowanie

Przedstawione modelowanie zagrożeń nie jest kompletne. Zostało zakończone na etapie identyfikacji. Po ustaleniu co grozi aplikacji należy podjąć decyzję jakie środki zapobiegawcze powinny zostać podjęte i przejść do retrospekcji.

Narzędzia

Opracowano wiele narzędzi, które ułatwiają przeprowadzenie procesu modelowania zagrożeń [4, 3]. Należą one zarówno do kategorii oprogramowania komercyjnego, jak i *open source*. Do najciekawszych i najbardziej użytecznych z tej drugiej grupy można zaliczyć ▸ Microsoft Threat Modelling Tool oraz ▸ OWASP Threat Dragon. Pierwsze narzędzie dostosowane jest do metody SDL firmy Microsoft, automatyzuje czynności związane z tworzeniem modelu oraz wspiera klasyfikację zagrożeń STRIDE. Drugie ułatwia przeprowadzenie i dokumentowanie modelowania zagrożeń w zakresie tworzenia diagramów przepływu danych, sugerowania możliwych zagrożeń oraz wprowadzania mechanizmów łagodzących i zapobiegawczych. W grupie oprogramowania komercyjnego interesującym rozwiązaniem jest ▸ Irius risk. Jest to w zasadzie system ekspertowy, który zadając szereg pytań przeprowadza użytkownika przez proces modelowania zagrożeń.

Źródła I


Michał Sajdak i inni. *Wprowadzenie do bezpieczeństwa IT*. Tom 1. Securitum, Kraków, 2023.


Shostack+Associates. The Ultimate Beginner's Guide to Threat Modelling. URL: https://shostack.org/resources/threat-modeling (termin wizyty 23. 10. 2024).


Awesome Threat Modelling. URL: https://github.com/hysnsec/awesome-threat-modelling (termin wizyty 23. 10. 2024).


Threat Modelling Cheat Sheet. URL: https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html (termin wizyty 23. 10. 2024).


OWASP Threat Modelling Project. URL: https://owasp.org/www-project-threat-model/ (termin wizyty 23. 10. 2024).

Źródła II

- Threat Modelling Manifesto. URL: https://www.threatmodelingmanifesto.org/ (termin wizyty 23. 10. 2024).
- SAFECode. Tactical Threat Modelling. URL: https://safecode.org/wp-content/uploads/2017/05/SAFECode_TM_Whitepaper.pdf (termin wizyty 23. 10. 2024).

Pytania

?

Koniec

Dziękuję Państwu za uwagę!