Praktyczne modelowanie zagrożeń dla systemów teleinformatycznych z wykorzystaniem modelu STRIDE

Tomasz Kruk
Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, Instytut Automatyki i Informatyki Stosowanej, ul. Nowowiejska 15/19, 00-665 Warszawa

Streszczenie: W niniejszym artykule opisano praktyczne podejście do zagadnienia stanowiącego aktualnie nieodłączną część wytwarzania nowoczesnych zaawansowanych i złożonych systemów informatycznych – do modelowania zagrożeń teleinformatycznych. W artykule przedstawiono praktyczną zasadność i przebieg procesu modelowania zagrożeń, a następnie opisano jedną z najpopularniejszych metod identyfikacji i analizy zagrożeń – tak zwany model STRIDE.

Słowa kluczowe: cyberbezpieczeństwo, bezpieczne wytwarzanie oprogramowania, modelowanie zagrożeń, model STRIDE

1. Wprowadzenie

Modelowanie zagrożeń teleinformatycznych stało się nieodłącznym elementem wytwarzania profesjonalnego oprogramowania. Zwiększono nacisk na zapewnianie poza podstawową funkcjonalnością, spełniania niezbędnych wymagań niefunkcjonalnych wytwarzanego rozwiązania informatycznego.

O ile dojście do zakładanej celem projektu funkcjonalności oprogramowania jest procesem dającym się precyzyjnie zdefiniować, a ocena osiągnięcia tego stanu wydaje się w pełni weryfikowalna przez zastosowanie odpowiedniego katalogu testów akceptacyjnych przed wdrożeniem rozwiązania w środowisku docelowym u klienta, o tyle zapewnienie podstawowych wymagań niefunkcjonalnych wymaga nadal troskliwej pracy analitycznej.

Z wymagań niefunkcjonalnych w ostatnich latach na czoło zdecydowanie wysunęło się zapewnienie właściwego poziomu bezpieczeństwa wytwarzanego rozwiązania informatycznego. Ataki na systemy informatyczne stały się równie częste jak samo użytkowanie tych systemów. Dziś wdrożenie rozwiązania informatycznego bez zapewnienia właściwej dojrzałości z punktu widzenia bezpieczeństwa, to poważny błąd projektowy, najpewniej doprowadzi do problemów, które w takim scenariuszu zmaterializują się bardzo szybko. Kosztem kompromitacji systemu może być utrata reputacji, wyciek danych, czasem paraliż funkcjonowania instytucji a czasem wręcz upadek firmy.

Autor korespondujący:
Tomasz Kruk, tomasz.kruk@pw.edu.pl

Artykuł recenzowany
nadesłany 04.10.2021 r., przyjęty do druku 03.12.2021 r.

Sprawdzoną techniką tworzenia bezpiecznych rozwiązań jest zastosowanie analizy ryzyka, w tym przede wszystkim modelowania zagrożeń teleinformatycznych i podjęcie z wyprzedzeniem działań neutralizujących możliwe ataki teleinformatyczne.

W teorii zarządzania analiza ryzyka jest elementem składowym grupy działań współtworzących razem zarządzanie ryzykiem. Poza identyfikacją i analizą ryzyka teoria oczekuje jeszcze: określenia celu zarządzania ryzykiem, oczywistego w kontekście analizy zagrożeń teleinformatycznych, wyboru metod zarządzania ryzykiem oraz monitorowania, rozumianego jako ocena efektywności wybranych metod zarządzania ryzykiem.

Artykuł nie ma charakteru sprawozdania z wykonanych badań. Artykuł ma charakter wprowadzający i przeglądowy koncentrując się na praktycznych aspektach wykonania analizy ryzyka zgodnie z modelem STRIDE.

2. Praktyka modelowania zagrożeń

U podstaw modelowania zagrożeń w systemach teleinformatycznych leżą cztery proste pytania:

1. Jaki jest przedmiot projektu?
2. Co może pójść nie tak?
3. Co zamierzamy z tym zrobić?
4. Czy właściwie wykonano ewaluację?

O potencjalnych intruzach, którzy mogą wykorzystać podatności wiadomo mniej niż o samym projekcie chronionego rozwiązania, stąd próba trafnego opisania możliwych zachowań potencjalnych intruzów z założenia będzie wysoce niedoskonała. Warto natomiast skoncentrować się na elemencie dobrze rozpoznanym, czyli samym realizowanym projekcie. Łatwiej jest opisać w pewnym wymiarze dobrze zdefiniowany projekt niż możliwe ledwie mgliście zarysowane zachowania potencjalnych intruzów. W kontekście modelowania zagrożeń i analizy ryzyka dobrym podejściem wydaje się próba zdefiniowania możliwych do wystąpienia podatności realizowanego rozwią-

zania oraz, na zasadzie analogii, możliwych metod eksploatacji takich podatności.

Pierwsze pytanie w modelowaniu zagrożeń brzmi: co jest zasadniczym przedmiotem projektu? Zagadnienie należy rozpatrywać w szerszej perspektywie. Powinna ona obejmować nie tylko architekturę naszego rozwiązania, ale również doprecyzować jego otoczenie oraz interakcje wewnętrzne i zewnętrzne. Stąd dobrym punktem wyjścia do analizy poziomu bezpieczeństwa jest rozpoczęcie od stworzenia pewnej reprezentacji graficznej rozwiązania w postaci diagramu. Diagramu, który z jednej strony zobrazuje w sposób wykorzystywalny na rzecz analizy bezpieczeństwa architekturę przedmiotowego rozwiązania, a zarazem wskaże interakcje zewnętrzne oraz wpływ dostępności i poprawności działania elementów zewnętrznych. O ile teoretycznie rozpatruje się różne podejścia, w praktyce zawsze warto zacząć od bardzo ogólnego i stosunkowo prostego modelu i diagramu. Z założenia nie będzie on doskonały, natomiast warto pamiętać, że żaden diagram nie odda w pełni precyzyjnie całej architektury i zasady działania rozwiązania.

Po wytworzeniu wersji inicjalnej diagramu, w trakcie pracy nad analizą podatności i zagrożeń, naturalnym jest iteracyjne pogłębianie, doprecyzowywanie tworzonej reprezentacji graficznej. Diagram w trakcie prac nad modelem zagrożeń ulega nie tylko doprecyzowywaniu – ale i zmianom. Jest to proces naturalny. Zmiany są dowodem użyteczności przyjętego sposobu działania – gdyby nie narysowano poprzedniej, jak się okazuje pierwotnie niedoskonałej wersji architektury jako diagramu, nie byłoby możliwości korekty percepcji postrzegania analizowanej architektury.

Po narysowaniu diagramu zaczyna się uzupełniać katalog zagrożeń. Zazwyczaj osoba rysująca ma już na samym początku kilka z takich zagrożeń na myśli. Ich źródłem mogą być podobieństwa wytwarzanego rozwiązania do analizowanych uprzednio pod względem modelowania zagrożeń rozwiązań podobnych, albo świadomość aktualnie popularnych i wykorzystywanych metod ataku systemów informatycznych. Okazuje się bowiem, że w przypadku cyberbezpieczeństwa również można mówić o pewnej sezonowości występowania różnych typów ataków. Zazwyczaj bieżąca moda na konkretny wektor ataku wynika z katalogu zidentyfikowanych w niedalekiej przeszłości podatności oraz dostępności do wykorzystania gotowych fragmentów oprogramowania możliwych do natychmiastowej eksploatacji znanych podatności.

Znalezienie kilku oczywistych i kilku nieoczywistych podatności prowadzi do kolejnego pytania, które współstanowi istotę modelowania zagrożeń: co można i co planuje się zrobić z każdym ze zidentyfikowanych zagrożeń?

Bez wątpienia pierwszą czynnością, którą względem każdego ze zidentyfikowanych zagrożeń należy wykonać – jest odnotowanie w katalogu zagrożeń do dalszej analizy. Gromadzenie zestawienia można rozpocząć przy krótkim zestawieniu w postaci na przykład współdzielonego przez analityków zasobu (dokument, biała tablica itp.). Gdy katalog wydaje się dopełniony, jego zawartość stanowi wkład do systemu zagadnień projektowych dotyczących bezpieczeństwa, narzędzia wspierającego planowe, terminowe i częściowo zautomatyzowane rozwiązywanie zagadnień projektowych, typu: system zarządzania zleceniami (ang. ticketing system), czy system śledzenia postępów w realizacji projektów informatycznych, jak na przykład oprogramowanie Jira.

Zebrany katalog często na tym etapie obejmuje również jako przedmiot analizy wybrane elementy czy scenariusze, które dopiero w wyniku pogłębionego przebadania okazują się nie być w rzeczywistości problematyczne z punktu widzenia bezpieczeństwa. Dotychczasowe uwzględnienie ich wśród analizowanych zagadnień nie było jednak błędem – umożliwiło poddanie zagadnienia pod rozwagę, dzięki czemu przedmiotowy scenariusz jest świadomie odrzucany z dalszej analizy, gdyż nie wymaga żadnych działań neutralizujących. Modelowanie zagrożeń obejmuje również właśnie uwzględnianie scenariuszy, których negatywny wpływ nie musi być jednoznacznie oczywisty. Jeżeli nie jest to rzeczywisty problem bezpieczeństwa, można taki fakt udokumentować odpowiednim komentarzem, a w wybranych sytuacjach można przykładowo napisać kod testowy wykazujący prawidłowe działanie środków zaradczych. Ważne jest, by z założenia kwestiami bezpieczeństwa teleinformatycznego zarządzać równie kompleksowo jak kompleksowo zarządza się innymi wymiarami realizacji projektów i tworzenia produktów w danym przedsiębiorstwie.

Zasadniczym efektem końcowym fazy modelowania zagrożeń a zarazem wkładem etapu w proces dostarczania klientom procesów i usług jest przede wszystkim sama lista zidentyfikowanych do dalszej ewaluacji zagrożeń.

Poprzedzone identyfikacją zagrożeń zaplanowanie działań neutralizujących nie kończy procesu modelowania zagrożeń. Kolejnym zadaniem, które należy zrealizować jest możliwie rzetelna ocena jakości dotychczas przeprowadzonych działań. Podczas sprawdzania, co mogło zostać opisane nieprawidłowo, ważne jest, by wyszukiwać zagrożenia w każdym jednym elemencie diagramu DFD lub każdej części diagramu, która znajduje się wewnątrz odpowiednich granic zaufania. Stąd kolejne pytanie, na które odpowiedź koniecznie trzeba zweryfikować, brzmi: czy dla każdego z elementów diagramu rzeczywiście przeanalizowano możliwość wystąpienie każdego typu zagrożenia, jakkolwiek prawdopodobnego w kontekście typu danego elementu diagramu i roli tego elementu w całym analizowanym systemie.

Dla każdego z wypisanych zagrożeń dokładność opisu jest na tym etapie mniej istotna niż sam fakt umieszczenia w zestawieniu niekorzystnego scenariusza, który mógłby zajść. Oczywiście są pewne granice upraszczania opisu – zagrożenie opisane zbyt ogólnie przeważnie uniemożliwia wdrożenie realnie skutecznych środków zaradczych. Przykładowo, zagrożenie opisane jako „ktoś może manipulować treścią pliku" jest istotnie mniej użyteczne niż opis „ktoś może manipulować treścią plików bazy danych w pomocniczym katalogu klienta i spowodować niecelowe wyświetlanie na stronie pierwotnie nieprzeznaczonych do wyświetlania treści". Przy modelowaniu zagrożeń oczekiwane jest właściwie konkretne dodefiniowanie charakteru i kontekstu zagrożenia. Co mogłoby się wydarzyć w scenariuszu przełamania zabezpieczeń i gdzie dokładnie w systemie taki scenariusz może wystąpić.

Warto wspomnieć, że modelownie zagrożeń może do późnego etapu - a czasem zupełnie – abstrahować od atrybucji źródeł ataków, czyli pomijać identyfikację potencjalnych rzeczywistych aktorów realizujących dany atak, o ile zaproponowane metody neutralizacji ataków danego typu są w stanie również abstrahować od takiej atrybucji bez wpływu na skuteczność metod przeciwdziałania. Przykładowo, wdrożenie do zapewnienia poufności transmisji szyfrowania kanału komunikacyjnego można uznać za skuteczne działania neutralizujące zagrożenie wycieku informacji podczas transmisji niezależnie od tego, kto próbowałby taką transmisję podsłuchiwać.

Można zatem nie mieć pewności co do podmiotu, źródła takiej aktywności. Można nie mieć pewności co do celu takich działań. Na tym etapie może się również zdarzyć, że po w miarę kompletnym zidentyfikowaniu charakteru zagrożenia nie widać jeszcze oczywistej ścieżki jego neutralizacji. Na tym etapie najważniejsze jest, by samo zagrożenie zostało chociaż zidentyfikowane i wstępnie zaewidencjonowane – stanie się wkładem do późniejszej analizy zagrożeń i metod ich przeciwdziałania.

Klasycznie modelowanie zagrożeń stanowi element bardziej pojemnego zagadnienia określanego mianem analizy ryzyka. Analiza ryzyka przeważnie kładzie silny nacisk na uwzględ-

nianie prawdopodobieństwa wystąpienia czy inaczej materializacji analizowanych niebezpiecznych scenariuszy. Nie warto zajmować się zagadnieniami wysoce nieprawdopodobnymi, bo szansa ich wystąpienia jest znikoma. Z drugiej strony trzeba uważać, by nie wpaść w pułapkę nieuzasadnionego merytorycznie upraszczania, czyli próby klasyfikowania scenariuszy jako nieprawdopodobnych, w rzeczywistości tylko dlatego, że zespół identyfikujący zagrożenia choć jest zagrożenia świadom, to nie ma wiedzy czy środków, które wymagane byłyby dla neutralizacji zagrożenia, a samo zagrożenie w rzeczywistości zgodnie z wiedzą zespołu jest jak najbardziej prawdopodobne.

3. Reprezentacje wizualne oparte na diagramach DFD

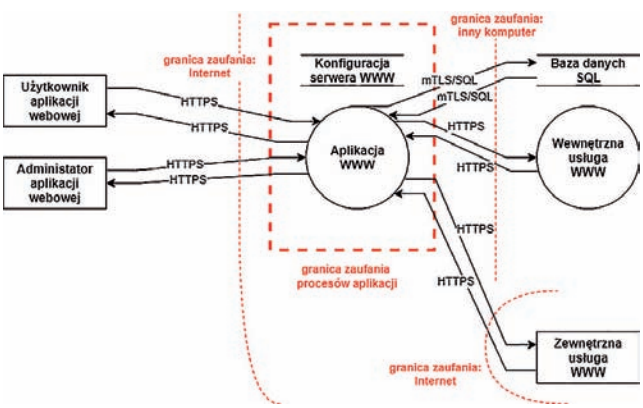
Metody modelowania zagrożeń rozwinęły wizualną reprezentację aplikacji i infrastruktury wykorzystującą diagramy przepływu danych DFD (ang. *data flow diagram*). Diagramy DFD zostały opracowane w latach 70. XX wieku jako narzędzie dla inżynierów systemowych do wysokopoziomowej reprezentacji sposobu w jaki aplikacja zarządza danymi: jak realizuje przepływy, przechowywanie oraz manipulowanie danymi w infrastrukturze, na której się wykonuje.

Tradycyjnie, diagramy DFD wykorzystywały zaledwie cztery symbole: przepływy danych, magazyny danych, procesy i aktorów. Potem dodano dodatkowy symbol, tak zwane granice zaufania, by umożliwić wykorzystanie diagramów DFD właśnie do modelowania zagrożeń teleinformatycznych.

Celowo ograniczony graficzny język opisu oraz ograniczona liczba dostępnych typów elementów sprzyja utrzymaniu dyscypliny opisu, a także wspiera uniwersalność i komunikatywność diagramów – następnie często badanych przez analityków bezpieczeństwa, którzy nie byli przecież ich autorami. Poprzez zastosowanie stosunkowo prostej reprezentacji graficznej unika się ryzyka wystąpienia przerostu formy prezentacji diagramu nad jego treścią.

Po rozpisaniu systemu na pięć typów elementów, eksperci do spraw bezpieczeństwa analizują każdy zidentyfikowany punkt wejścia zagrożenia pod kątem wszystkich znanych kategorii zagrożeń. Po zidentyfikowaniu potencjalnych zagrożeń można ustalić środki zaradcze ograniczające zagrożenia lub przeprowadzić dodatkową analizę.

Podsumowując, diagram przepływu danych może zawierać następujące typy elementów:
- procesy
- przepływy danych
- interaktorzy
- magazyny danych
- granice zaufania



Rys. 1. Przykładowy schemat przepływu danych
Fig. 1. Example of a data flow diagram (DFD)

4. Metodyka STRIDE w modelowaniu zagrożeń

Model STRIDE wywodzi swoją nazwę z zestawienia różnych metod ataku na systemy informatyczne, katalog zagrożeńobejmuje:
1. **S**poofing
2. **T**ampering
3. **R**epudiowanie
4. **I**nformation disclousure
5. **D**enial of Service
6. **E**levation of Privilige

Mnemonik STRIDE ułatwia systematyczną kompletną analizę potencjalnych zagrożeń na rzecz wybranych, bądź często wszystkich, elementów składowych diagramu DFD. Przyłożenie sześciu typów potencjalnych zagrożeń do danego elementu pozwala nam założyć, że element ten jako komponent badanego systemu został wyczerpująco przeanalizowany pod względem możliwych potencjalnych zagrożeń bezpieczeństwa. Poniżej rozwinięto opis poszczególnych składowych zestawienia typów zagrożeń stanowiącego model STRIDE.

4.1. Spoofing (podszywanie się, np. spoofing określonego serwera)

Scenariusz określany mianem spoofing to sytuacja, gdy wystąpiło podszycie się pod czyjąś tożsamość, gdy nieprawidłowo zweryfikowano tożsamość. Zjawisko może dotyczyć naruszenia autentyczności źródła strony internetowej, która ładuje się w wyniku wybrania pewnego adresu URL. Własność autentyczności polega na tym, że nazwa odpowiada pewnym przyjętym przez użytkownika oczekiwaniom. W sytuacji, gdy atakujący wykonuje spoofing, dostarcza podróbkę, treść nieautentyczną, niezakładaną w miejsce treści prawdziwej.

4.2. Tampering (manipulacja, np. manipulowanie plikiem)

T w STRIDE oznacza tampering, czyli nieautoryzowaną modyfikację. Nieautoryzowana modyfikacja dotyczyć może zarówno zmiany zawartości pewnego pliku jak i podmienienie części informacji w realizowanej komunikacji sieciowej. W prawdziwym ataku typu man-in-the-middle, pierwszym krokiem jest przekonanie jednego z komputerów na diagramie, że wskazany komputer jest najlepszym pośrednikiem sieciowym, routerem dla jego pakietów. Takie przekierowanie może być zaskakująco skuteczne w przejęciu kontroli nad ruchem sieciowym, umożliwiając nowemu pośrednikowi w komunikacji na odczyt, zmianę, odrzucanie lub tworzenie dowolnych pakietów stanowiących element przejętej komunikacji. Dzięki skutecznemu przekierowaniu atakujący nie musi znajdować się wewnątrz obszaru wzajemnego zaufania.

Rozwiązanie problemu manipulacji jest relatywnie prostsze niż rozwiązanie problemu ataku poprzez spoofing. W przypadku plików lokalnych, uprawnienia systemu operacyjnego są solidne, zakładając, że zostały poprawnie skonfigurowane. Podobnie w chmurze, należy korzystać z możliwości zabezpieczania, które zapewnia system. W przypadku komunikacji sieciowej należy używać kryptograficznej ochrony integralności transmisji, na przykład poprzez wykorzystanie protokołu TLS.

4.3. Repudiaton (wyparcie się, np. odrzucenie zamówienia)

R w STRIDE oznacza wyparcie się, odrzucenie. Odrzucenie ma odmienną specyfikę niż pozostałe typy zagrożeń stanowiące model STRIDE. Jest to rzadziej występujące określenie i oznacza zrzeczenie się, zaprzeczenie lub jakikolwiek sposób

przekazania, że nie jest się za coś odpowiedzialnym. Przykładem wyparcia się, odrzucenia jest stwierdzenie o treści „Nie otrzymałem takiej wiadomości" a także „Nie można stwierdzić, czy na pewno otrzymałem taką wiadomość".

Przyczyną materializacji takiego zagrożenia może być brak wdrożonych rejestrów przesyłanych wiadomości albo braki takich dzienników systemowych za wybranych podokres. W celu neutralizacji zagrożenia niezbędne jest prowadzenie niepodważalnych rejestrów, na przykład dzienników systemowych, których zawartość zostanie dostarczona jako kontrargument w sytuacji podważania przez uczestnika pewnej aktywności swego udziału w tej aktywności.

4.4. Ujawnianie informacji

Litera I w modelu zagrożeń STRIDE oznacza ujawnianie informacji (ang. *information disclousure*).

W sieci, najlepsza poufność pochodzi z kryptografii. W rzeczywistości kryptografia jest najlepszym sposobem na ochronę każdego sekretu, ale wtedy trzeba byłoby zarządzać ogromną liczbą kluczy, a to jest skomplikowane. TLS w większości przypadków zajmuje się zarządzaniem kluczami za użytkownika. W ramach systemu, łatwiejsze może być użycie uprawnień. Większość serwerów WWW umieszcza zarządzanie użytkownikami i plikami w serwerze WWW. Ponieważ użytkownik loguje się do serwera WWW, nie może zalogować się do niego przez SSH. Wadą tego rozwiązania jest konieczność wyboru i zarządzania mechanizmem uprawnień

Poufność może być wymagana w odniesieniu do treści lub metadanych komunikacji. Czasami obie te informacje muszą pozostać poufne. Spółki konkurują ze sobą pod względem sposobu, w jaki chronią informacje o użytkowniku przed niewłaściwym wykorzystaniem lub ujawnieniem.

4.5. Odmowa świadczenia usług

D w STRIDE oznacza denial-of-service (odmowa usługi). Istnieją ataki typu denial-of-service (lub DoS) przeciwko procesorom, sieciom i pamięci masowej. Najprostsze ataki denial-of-service to po prostu brute-force. Przy dużej ilości żądań obejrzenia danej reklamy, sieć się przepełnia. Albo z powodu połączeń przychodzących, albo, co bardziej prawdopodobne, danych wychodzących.

Najprostszym sposobem obrony przed atakami typu denial-of-service jest obfitość zasobów, które są trudne do wyczerpania przez atakujących. Jest to również właściwe rozwiązanie do szybkiego obsłużenia dużej liczby klientów choć jest oczywiście drogie. Obrona przed rozproszonymi atakami jest czymś, co najlepiej zrobić na poziomie sieci lub dostawcy chmury. Obrona przed sprytnymi atakami wymaga profilowania aplikacji i wiedzy o tym, jak będą się one zachowywać. Podobnie jak ujawnianie informacji dotyczy bezpieczeństwa i prywatności, utrzymywanie dostępności systemów jest zarówno właściwością bezpieczeństwa, jak i niezawodności.

4.6. Zwiększenie poziomu uprzywilejowania

E w STRIDE oznacza podniesienie przywilejów, czyli zmniejszenie zestawu ograniczeń stosowanych wobec konta użytkownika. Jeśli więc obecnie użytkownik ma ograniczenie do wysyłania pakietów sieciowych do serwera AD, może podnieść swoje uprawnienia do uruchamiania kodu lub nawet do poziomu uprawnień administratora. Jeśli ktoś może coś zmienić, a jego reakcja jest taka, że nie powinien móc tego zrobić, może to oznaczać atak typu „elevation of privilege". Niektóre problemy związane z podnoszeniem przywilejów dotyczą zasobów chronionych tylko przez nieprzejrzystość, jak przykładowo udostępnienie panelu kontrolnego administratora serwera WWW pod nieoczywistą ścieżką dostępu.

Wiele innych ataków na podniesienie przywilejów dotyczy sposobu przetwarzania nieuprzywilejowanych danych wejściowych lub pomylenia różnicy między kodem a danymi. Przykładowo, atak SQL injection podnosi przywilej poprzez uruchomienie kodu, który serwer WWW przekazuje do bazy danych, gdzie serwer WWW pobrał dane wejściowe i pozwolił, aby niektóre z nich były traktowane jako kod.

Podobnie, atak typu cross-site scripting daje atakującemu przywilej uruchamiania kodu i prawdopodobnie innych aktywności. Istnieje również wiele klasycznych ataków z przestrzeni procesów użytkowników do poziomu konta root, które zazwyczaj wykorzystują programy z ustawionym SUID.

Unikanie ataków typu „elevation of privilege" wymaga dobrze zaprojektowanego i wdrożonego systemu autoryzacji.

4.7 Zakres stosowalności modelu

Nie wszystkie zagrożenia ujęte w modelu STRIDE mają zastosowanie do wszystkich typów elementów diagramów przepływu danych. Najwięcej typów zagrożeń, a zarazem największy wysiłek analityczny, związany jest z opisem procesów składowych jako elementów diagramu DFD. Pełne zestawienie stosowalności analizy poszczególnych typów zagrożeń modelu STRIDE na rzecz różnych typów składowych architektury systemu informatycznego opisanej diagramem DFD zestawiono na rysunku 2.

ELEMENT	S	T	R	I	D	E
Interaktorzy	√		√			
Procesy	√	√	√	√	√	√
Magazyny danych		√	?	√	√	
Przepływy danych		√		√	√	

Rys. 2. Elementy STRIDE stosowane do poszczególnych typów elementów diagramów DFD
Fig. 2. STRIDE elements used for individual types of DFD diagram elements

4.8. Narzędzia dedykowane do modelowania zagrożeń

Istnieją narzędzia informatyczne dedykowane do modelowania zagrożeń, a w szczególności wspomagające poprawne wytwarzanie diagramów przepływu danych. Najpopularniejsze ogólnodostępne pakiety to:

- Microsoft Threat Modeling Tool, https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool
- OWASP Threat Dragon, OWASP, https://owasp.org/www-project-threat-dragon/

5. Podsumowanie

Modelowanie zagrożeń jest niezwykle wydajne na początku projektu. Umożliwia systematyczne uwzględnianie w architekturze systemu niezbędnych kompromisów wynikających z wymagań bezpieczeństwa. Więcej ograniczeń pojawia się, gdy modelowanie zagrożeń rozpoczyna się pod koniec projektu lub jest wprowadzane podczas aktualizacji istniejącego produktu lub usługi. Trudniej jest zadawać pytania podstawowe.

Systematyczne, ustrukturyzowane i kompleksowe podejście do modelowania zagrożeń prowadzi do osiągania rzetelnych wniosków w bardziej przewidywalnych ramach czasowych.

Rozpatrując przedmiot każdego nowego przedsięwzięcia czy projektu warto jak najszybciej do aktywności projektowych wdrożyć analizę zagrożeń, czyli zadawanie sobie pytania, co w tym przedsięwzięciu może zajść niezgodnie z planem, co może wpłynąć na niezawodność, często dostępność w przy-

padku usług, czy bezpieczeństwo tworzonego rozwiązania. Warto również sięgać do katalogów zagrożeń i środków zaradczych zidentyfikowanych podczas poprzednich tematycznie zbliżonych projektów oraz do sprawdzonych praktycznie metod modelowania zagrożeń, takich jak opisana w artykule metodyka STRIDE.

Bibliografia

1. Shostack A., *Threat Modeling: Designing for Security*, John Wiley & Sons, 2014.
2. UcedaVelez T., Morana M.M., *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*, John Wiley & Sons, 2015.
3. Shevchenko N., *Threat Modeling: 12 Available Methods*, Carnegie Mellon University, 2018, https://insights.sei.cmu.edu/blog/threat-modeling-12-available-methods/.
4. Jagannathan V., *Threat Modeling, Architecting and Designing with Security in Mind*, 2016, https://owasp.org/www-pdf-archive/AdvancedThreatModeling.pdf
5. Gumbley J., *A Guide to Threat Modelling for Developers*, 2020, https://martinfowler.com/articles/agile-threat-modelling.html
6. Jelacic B., Rosic D., Lendak I., Stanojevic M., Stoja S. (2018) *STRIDE to a Secure Smart Grid in a Hybrid Cloud*. [In:] Katsikas S. et al. (eds) Computer Security, SECPRE 2017, CyberICPS 2017. "Lecture Notes in Computer Science", Vol. 10683. Springer, Cham. DOI: 10.1007/978-3-319-72817-9_6.
7. Khan R., McLaughlin K., Laverty D., Sezer S., (2018). *STRIDE-based Threat Modeling for Cyber-Physical Systems*. [In:] Proceedings IEEE of 2017 IEEE PES: Innovative Smart Grid Technologies Conference Europe (ISGT-Europe): DOI: 10.1109/ISGTEurope.2017.8260283
8. Kaneko T., *Threat analysis using STRIDE with STAMP/STPA*, CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2809, 2018.
9. de Souza N.P., César C.D.A.C., Bezerra J.D.M., Hirata C.M., *Extending STPA with STRIDE to identify cybersecurity loss scenarios*. "Journal of Information Security and Applications". Vol. 55, 2020, DOI: 10.1016/j.jisa.2020.102620.
10. Sattar D., Vasoukolaei A.H., Crysdale P., Matrawy A., *A STRIDE Threat Model for 5G Core Slicing*, 2021 IEEE 4th 5G World Forum (5GWF), 2021, 247–252, DOI: 10.1109/5GWF52925.2021.00050.

Practical Modelling of Threats to ICT Systems Using the STRIDE Model

Abstract: This article describes a practical approach to the issue which is currently an integral part of the development of modern advanced and complex information systems – ICT threat modelling. The article presents the practical validity and process of threat modelling and then describes one of the most popular methods of threat identification and analysis – the so-called STRIDE model.

Keywords: cyber security, secure software development, threat modeling, STRIDE model

dr inż. Tomasz Kruk
tomasz.kruk@pw.edu.pl
ORCID: 0000-0002-4907-7688

Ekspert informatyki specjalizujący się w bezpieczeństwie IT oraz projektowaniu systemów informatycznych dużej skali. Od 2001 r. wykładowca informatyki na Politechnice Warszawskiej, laureat studenckiej Złotej Kredy (2014, 2017, 2019). W latach 2010–2016 dyrektor operacyjny w instytucie badawczym NASK. W latach 2010–2015 członek Komitetu Sterującego NCBiR do spraw badań naukowych i prac rozwojowych w obszarze bezpieczeństwa i obronności państwa. W latach 2012–2016 członek Rady Polskiej Izby Informatyki i Telekomunikacji. Od 2021 r. również inżynier bezpieczeństwa w Amazon/AWS.