Bezpieczeństwo systemów operacyjnych

Wprowadzenie

Arkadiusz Chrobot

Katedra Systemów Informatycznych

2 października 2026

Plan

1. Wstęp
   - Kontakt
   - Literatura
   - Inicjatywy
2. Modelowanie zagrożeń
   - Nad czym pracujemy?
   - Co może pójść źle?
   - Co możemy z tym zrobić?
   - Czy dobrze wykonaliśmy zadanie?

Wstęp

Motto

At the start of this century, security technology was an archipelago of mutually suspicious islands — the cryptologists, the operating system protection people, the burglar alarm industry, right through to the chemists who did banknote inks. We all thought the world ended at our shore. By 2010, security engineering was an established and growing discipline; the islands were being joined up by bridges as practitioners realised we had to look beyond our comfort zones.

Ross Andreson, „Security Engineering: A Guide To Building Dependable Distributed Systems”

W latach 90. poprzedniego wieku badania nad cyberbezpieczeństwem dotyczyły głównie kryptologii oraz systemów operacyjnych. Współcześnie poświęcone są one większej liczbie zagadnień, włączając w to zastosowania sztucznej inteligencji, a nawet ekonomię i psychologię. Niemniej, system operacyjny nadal odgrywa kluczową rolę w zabezpieczaniu systemu komputerowego i przetwarzanych oraz przechowywanych przez niego danych.

Kontakt

- dr inż. Arkadiusz Chrobot
- pokój: 3.23 D
- telefon: 41 34-24-185
- e-mail: a.chrobot@tu.kielce.pl
- termin konsultacji: poniedziałek, 12:00 – 14:00
- www: https://achilles.tu.kielce.pl/portal/Members/84df831b59534bdc88bef09b15e73c99

Literatura

- Ross Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 2020. URL: https://www.cl.cam.ac.uk/~rja14/book.html.
- Michał Sajdak i inni. *Wprowadzenie do bezpieczeństwa IT*. Tom 1. Securitum, Kraków, 2023.
- Michał Sajdak i inni. *Wprowadzenie do bezpieczeństwa IT*. Tom 2. Securitum, Kraków, 2024.
- Andrew S. Tanenbaum i Herbert Boss. *Systemy operacyjne*. Helion, Gliwice, 2016.
- Abraham Silberschatz i inni. *Podstawy systemów operacyjnych*. Tom 2. Polskie Wydawnictwo Naukowe PWN SA, Warszawa, 2021.

Inicjatywy

- ▸ Kursy IBM,
- IBM Innovation Project, IBM AI Racing League (mgr inż. Konrad Sich),
- projekt ▸ Akademia_CYBER.MIL,
- ▸ Erasmus+ Internship,
- projekt CriTech (FERS).

Modelowanie zagrożeń

Jest wiele metod prowadzenia modelowania zagrożeń, jednak najprostszą i najbardziej ogólną jest udzielenie odpowiedzi na cztery poniższe pytania:

1. Nad czym pracujemy?
2. Co może pójść źle?
3. Co możemy z tym zrobić?
4. Czy dobrze wykonaliśmy zadanie?

Pierwsze pytanie wymaga poznania oprogramowania, dla którego określany jest model zagrożeń. Drugie dotyczy potencjalnych zagrożeń, które mogą go dotyczyć, a trzecie środków zaradczych, które będą im przeciwdziałać. W końcu ostatnie pytanie związane jest z *retrospekcją*, która w tym kontekście oznacza sprawdzenie, czy żadne istotne szczegóły nie zostały pominięte.

Nad czym pracujemy?

W przypadku tego wykładu, oprogramowaniem, dla którego będą modelowane zagrożenia, jest system operacyjny. Zazwyczaj pierwszy krok modelowania zagrożeń obejmuje analizę dokładnego projektu oprogramowania, w postaci diagramów przedstawiających jego architekturę, przepływ danych i sterowania. Tym razem będzie on jednak przebiegał nieco inaczej, gdyż nie będzie dotyczył konkretnego systemu operacyjnego, ale ogólnie programów tego rodzaju. Zatem najpierw trzeba odpowiedzieć na pytanie „czym jest system operacyjny”? Okazuje się, że nie jest to proste, bo poszczególne egzemplarze takich systemów różnią się nie tylko budową, ale również zastosowaniem, które znacząco wpływa na ich konstrukcję. Prościej jest określić miejsce systemu operacyjnego w systemie komputerowym (Rysunek 1) i wskazać jakie funkcje (typowo) w nim pełni.

System komputerowy



Schemat systemu komputerowego

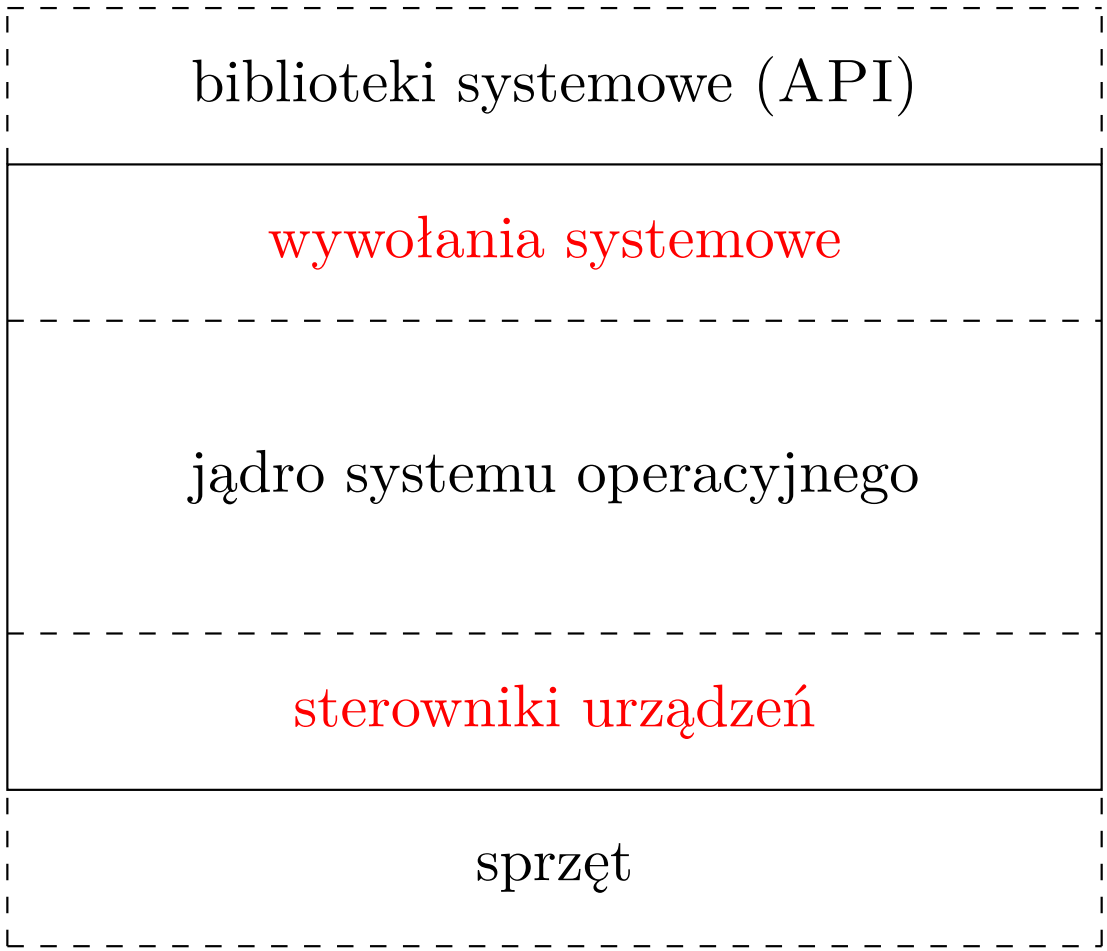
Funkcje systemu operacyjnego

Wśród funkcji systemu operacyjnego najczęściej:

1. tworzenie środowiska wykonawczego dla procesów użytkownika,
2. nadzór nad wykonaniem tych procesów,
3. zarządzanie zasobami systemu komputerowego.

Aby efektywnie je spełniać, najważniejszy komponent systemu, jego jądro (ang. *kernel*) wykonywane jest w trybie uprzywilejowanym procesora (fizycznego lub wirtualnego). Ten opis nie jest jednak wystarczający, aby przeprowadzić poprawnie modelowanie zagrożeń. Przyjmijmy zatem, że przeprowadzimy ten proces dla systemu operacyjnego bazującego na najpowszechniej stosowanej architekturze z jądrem monolitycznym (Rysunek 2).

System operacyjny



Schemat systemu operacyjnego z jądrem monolitycznym

Analiza budowy

Na Rysunku 2 ujawnione są dwie granice zaufania, czyli interfejsy, pośredniczące między komponentem o wyższym poziomie zaufania (jądro systemu) i komponentami o niższym poziomie (procesy użytkownika lub sterowniki urządzeń). Dlaczego granicę stanowią sterowniki? Ponieważ mogą być dostarczane przez producentów sprzętu, a nie programistów tworzących system, stanowią część jądra — mają ten sam poziom uprzywilejowania — ale mogą być do niego dodawane w sposób dynamiczny.
W pierwszym kroku modelowania zagrożeń ważne jest również określenie jakie aktywa (szczególnie ważne zasoby) powinny podlegać ochronie. W przypadku systemów operacyjnych będą to zasoby sprzętowe (procesor, urządzenia wejścia-wyjścia, pamięć operacyjna i masowa) oraz logiczne (pliki, łącza komunikacyjne, dane).

Co może pójść źle?

W tym kroku modelowania zagrożeń należy wskazać potencjalne zagrożenia, posługując się określoną metodą. W przypadku oprogramowania najczęściej stosuje się klasyfikację STRIDE. Każda litera w jej nazwie oznacza określoną klasę zagrożeń (Tabela 1).

Tabela: Klasyfikacja STRIDE zagrożeń

Zagrożenie
Spoofing (Podszywanie się)
Tampering (Manipulacja)
Repudiation (Zaprzeczalność)
Information disclosure (Ujawnienie informacji)
Denial of service (Odmowa usługi)
Elevation of privileges (Podniesienie uprawnień)

Podszywanie się

Jako źródła zagrożenia polegającego na podszywaniu się można wskazać procesy, sterowniki urządzeń i same urządzenia. Procesy mogą próbować podawać się za inne wykonujące się programy, celem uzyskania dostępu do przydzielonych im zasobów logicznych i fizycznych. Z kolei fałszywe/złośliwe sterowniki urządzeń mogą udawać prawdziwe sterowniki ładowane dynamicznie, aby otrzymać nieograniczony dostęp do zasobów przysługujący oprogramowaniu, które wykonuje się w trybie jądra. W przypadku urządzeń istnieją rozwiązania sprzętowe lub sprzętowo-programowe, które umożliwiają podszywanie się pod urządzenia wejścia-wyjścia, takie jak klawiatury, pamięci USB (ang. *pendrive*) i punkty dostępowe Wi-Fi, celem kradzieży danych.

Manipulacja

Na manipulację narażony jest proces aktualizacji i instalacji oprogramowania. Jej skutkiem jest przedostanie się do systemu komputerowego złośliwych programów (ang. *malware*), które mogą próbować w nieautoryzowany sposób modyfikować przechowywane, przetwarzane i transmitowane dane, w tym dane konfiguracyjne innych programów użytkowych i systemowych, a nawet samego jądra operacyjnego. Szczególnie narażonym elementem są w tym przypadku wywołania systemowe, stanowiące wektor ataku dla prób manipulacji.

Zaprzeczalność

Brak rejestracji informacji o zdarzeniach w systemie, w tym **danych audytowych** skutkuje niemożnością przeprowadzenia diagnostyki problemów oraz udowodnienia przeprowadzania nieuprawnionych, w tym jawnie złośliwych operacji ze strony użytkowników i działających w ich imieniu procesów. Należy zwrócić uwagę na dwa dodatkowe problemy związane z rejestrowaniem zdarzeń — jak bardzo *szczegółowe* powinny być o nich informacje i gdzie je należy *przechowywać*.

Ujawnienie informacji

Do ujawnienia informacji może dojść na skutek błędów w konfiguracji lub implementacji głównie wywołań systemowych, ale ten problem może również dotyczyć innych komponentów systemu operacyjnego. Przyczyną może być również brak mechanizmów weryfikacji tożsamości komponentów systemu komputerowego i użytkowników. Ujawnione mogą zostać informacje o konfiguracji sprzętowej i programowej, klucze kryptograficzne, hasła w jawnej postaci, organizacja pamięci procesu i dane poufne użytkowników.

Odmowa usługi

Odmowa usługi może być spowodowana przez procesy, które mogą zmonopolizować procesor lub pamięć operacyjną, co skutkuje stanem potocznie nazywanym „zawieszeniem się” komputera. Może ona też polegać na przejęciu przez złośliwy lub niepoprawnie działający proces kontroli nad określonym urządzeniem wejścia-wyjścia. Warto zwrócić uwagę, że źródłem tego zagrożenia mogą być niepoprawnie zaprojektowane lub skonfigurowane mechanizmy zabezpieczające (np. rejestrujące informacje o zdarzeniach).

Podniesienie uprawnień

Chyba najczęściej występującym zagrożeniem w kontekście systemów operacyjnych są wszelkiego rodzaju próby podniesienia uprawnień wykonywanych procesów. W szczególności celem ataków wykorzystujących to zagrożenie jest uzyskanie uprawnień jądra systemu operacyjnego, co daje nieograniczony dostęp do zasobów systemu komputerowego. Jednak przejęcie uprawnień dowolnych procesów, ale bardziej uprzywilejowanych lub dysponujących innym zestawem praw dostępu także jest korzystne z punktu widzenia intruzów.

Co możemy z tym zrobić?

Ten krok polega na zaproponowaniu rozwiązań dla zidentyfikowanych zagrożeń. Najczęściej te rozwiązania zaliczają się do trzech kategorii: łagodzenia (ang. *mitigation*), przeniesienia (ang. *transfer*) lub akceptacji (ang. *acceptance*) ryzyka. Nie zawsze całkowite usunięcie ryzyka jest możliwe, bo może prowadzić do sytuacji, w której system operacyjny nie będzie spełniał wymagań.

Podszywanie się

Ryzyko zagrożenia poszywaniem się jest łagodzone poprzez wprowadzenie mechanizmów uwierzytelniania (ang. *authentication*) — inaczej weryfikacji tożsamości — oraz autoryzacji (ang. *authorization*) czyli mechanizmów sprawdzania przypisanej roli i związanych z nią uprawnień. Jest to też rozwiązanie zaliczane do kategorii przeniesienia ryzyka, bo mechanizmy uwierzytelniania są implementowane zazwyczaj całkowicie przez oprogramowanie działające w trybie użytkownika, a ich bezpieczeństwo jest gwarantowane zastosowaniem odpowiednio zaawansowanych narzędzi kryptograficznych. Uwierzytelnienie może być też oddelegowane do innych systemów komputerowych, jak dzieje się np. w przypadku Active Directory. W przypadku autoryzacji implementacja jest przeprowadzana w obu rodzajach oprogramowania, w szczególności jądro systemu musi rozpoznawać poziom uprzywilejowania procesów użytkownika korzystających z wywołań systemowych. Przykładem może być choćby mechanizm możliwości (ang. *capabilities*) używany w Linuksie.

Podszywanie się

Weryfikację tożsamości można również zastosować do urządzeń, np. poprzez użycie protokołów typu challenge – response. Z kolei sterowniki tych urządzeń mogą być podpisywane kryptograficznie przez producentów/twórców. Warto zwrócić uwagę, że w egzekwowaniu separacji procesów dysponujących różnymi poziomami uprzywilejowania system operacyjny polega na rozwiązaniach sprzętowych, jak choćby dualny tryb pracy procesora lub mechanizmy wspomagające realizację pamięci wirtualnej.

Manipulacja

Rozwiązania kryptograficzne pomocne są również w łagodzeniu ryzyka związanego z manipulacją. Podpisy cyfrowe można zastosować do weryfikacji autentyczności instalowanego lub aktualizowanego oprogramowania (w tym wspomnianych na poprzednim slajdzie sterowników).

Zaprzeczalność

Ryzyko związane z możliwością zaprzeczenia (wyparcia) wykonania wrażliwych operacji jest łagodzone poprzez wprowadzenie ich autoryzacji oraz rejestracji zdarzeń. W przypadku tego drugiego mechanizmu, jak wcześniej wspomniano, kluczowe są decyzje odnośnie do szczegółowości rejestrowanych danych i miejsca ich przechowywania. Informacje o zdarzeniach muszą być na tyle szczegółowe, aby ustalić źródło i przebieg operacji, ale z drugiej strony nie mogą być zbyt obszerne, aby nie spowodować zagrożenia odmowy usługi na skutek wyczerpania miejsca w pamięci masowej. Jeśli nie można ograniczyć ich szczegółowości, to należy rozważyć okresowe ich usuwanie lub przechowywanie w osobnym systemie komputerowym. To drugi rozwiązanie jest także wskazane, jeśli istnieje ryzyko, że te dane mogą zostać zmanipulowane. Innym środkiem zapobiegawczym przed taką ewentualnością jest zastosowanie sygnatur kryptograficznych.

Ujawnienie informacji

Ochronę informacji zapewniają zarówno mechanizmy autoryzacji, jak i zabezpieczenia kryptograficzne. Dodatkowe rozwiązania, takie jak *Address Space Layout Randomization* (ASLR) lub *Kernel Address Space Layout Randomization* (KASLR) chronią przez specyficznymi rodzajami ataków, które mogą skutkować ujawnieniem informacji.

Odmowa usługi

Ryzyko odmowy usługi można łagodzić poprzez zastosowanie nadmiarowości (redundancji), ale najczęściej stosowanym podejściem jest jego akceptacja, jeśli pozwala na to względnie niski stopień krytyczności systemu komputerowego. *Nie oznacza to jednak, że to ryzyko jest po prostu ignorowane*. Zachowanie usług, które są istotne dla systemu może podlegać monitorowaniu i w przypadku stwierdzenia drastycznego obniżenia ich wydajności mogą być podjęte adekwatne do sytuacji działania. Inne rozwiązania polegają na zapewnieniu odpowiedniej ilości zasobów dla krytycznych komponentów, kosztem jej ograniczenia dla tych, które są mniej istotne, poprzez choćby wprowadzenie priorytetowości w zarządzaniu zasobami i procesami lub wątkami.

Podniesienie uprawnień

Ryzyko podniesienia uprawnień jest łagodzone przez mechanizmy autoryzacji, ale znaczącą rolę odgrywają także w tym przypadku wspomniane wcześniej mechanizmy ASLR i KASLR. W przypadku jądra systemu operacyjnego, każde wywołanie systemowe musi dokładnie weryfikować przekazane argumentu i uprawnienia wywołującego je procesu.

Czy dobrze wykonaliśmy pracę?

Odpowiedź na to pytanie zależna jest od kontekstu. Najczęściej wymaga ona dokładniejszej analizy każdego ▸ komponentu i każdego zagrożenia. Należy pamiętać także o tym, że bezpieczeństwo wymaga kompromisów. Przykładowo, zmiana architektury systemu operacyjnego z jądra monolitycznego na mikrojądro podnosi poziom bezpieczeństwa, ale może źle wpływać na wydajność, co nie zawsze jest akceptowalne. Projekty systemów o krytycznym znaczeniu można oprzeć o tzw. *Zaufaną Bazę Obliczeniową* (ang. *Trusted Computing Base*), czyli minimalny zestaw oprogramowania i sprzętu niezbędny do egzekwowania wszystkich reguł bezpieczeństwa [1, 4]. Wymagania bezpieczeństwa są wyższe dla systemów mobilnych i wbudowanych. Dlatego tam jako pierwsze pojawiły się rozwiązania takie jak osobne procesory dla operacji związanych z bezpieczeństwem (tzw. enklawy) i mechanizmy bezpiecznego rozruch (ang. *secure boot*). Z czasem zaczęły one przenikać do komputerów osobistych, w postaci np. układów TPM (ang. *Trusted Platform Module*).

Pytania

?

KONIEC

Dziękuję Państwu za uwagę!