MATEMATYCZNE MODELE BEZPIECZEŃSTWA KOMPUTERA

DLACZEGO MODELE SĄ WAŻNE.

Kiedy prowadzisz nowy samochód, szukasz określonych przedmiotów, które pomogą Ci sterować samochodem: pedału gazu, hamulca, zmiany biegów i kierownicy. Istnieją one we wszystkich samochodach i pełnią funkcję przyspieszania samochodu, zwalniania go i obracania w lewo iw prawo. Tworzy to model samochodu. Przy prawidłowym funkcjonowaniu tych przedmiotów możesz stworzyć przekonujący argument, że model prawidłowo opisuje, co musi posiadać samochód, aby móc się odpowiednio poruszać i sterować. Model bezpieczeństwa komputerowego służy temu samemu celowi. Przedstawia ogólny opis systemu komputerowego (lub kolekcji systemów). Model zawiera definicję „ochrony" (np. „Zachować poufność" lub „zapobiegać nieupoważnionej zmianie") oraz warunki, w których zapewniona jest ochrona. W przypadku modeli matematycznych można wykazać, że warunki zapewniają określoną ochronę. Zapewnia to wysoki stopień pewności, że dane i programy są chronione, przy założeniu, że model jest wdrożony prawidłowo. Ten ostatni punkt jest krytyczny. Aby powrócić do naszej analogii do samochodu, zwróć uwagę na frazę „przy prawidłowym działaniu tych elementów". Oznacza to również, że przeciętny kierowca musi być w stanie pracować prawidłowo. W większości, jeśli nie we wszystkich samochodach, model jest realizowany w oczywisty sposób: pedał przyspieszenia znajduje się po prawej stronie pedału hamulca i przyspiesza samochód; pedał hamulca zwalnia; a przekręcenie kierownicy przesuwa samochód w lewo lub w prawo, w zależności od kierunku, w którym obraca się koło. Przeciętny kierowca jest zaznajomiony z tą implementacją i może z niej korzystać poprawnie. Tak więc model i realizacja razem pokazują, że ten konkretny samochód może być napędzany. Załóżmy teraz, że elementy są implementowane inaczej. Wszystkie przedmioty są, ale kierownica jest zablokowana, więc nie można jej obrócić. Mimo że samochód ma wszystkie części, których wymaga model, nie działają one tak, jak wymaga tego model. Implementacja jest niepoprawna, a argument, który zapewnia model, nie ma zastosowania do tego samochodu, ponieważ model przyjmuje założenia - takie jak obracanie kierownicy - które są nieprawidłowe dla tego samochodu. Czytelnik powinien pamiętać o założeniach, które tworzą modele. Kiedy stosuje się te modele do istniejących systemów lub wykorzystuje je do projektowania nowych systemów, należy upewnić się, że założenia są spełnione, aby uzyskać pewność, że model zapewnia. Ta część przedstawia kilka modeli matematycznych, z których każdy służy innemu celowi. Możemy podzielić te modele na kilka typów. Pierwszy zestaw modeli służy do określenia, w jakich warunkach można udowodnić, że typy systemów są bezpieczne. Model macierzy kontroli dostępu przedstawia ogólny opis systemu komputerowego, z którego korzysta ten typ modelu, i da pewne wyniki na temat rozstrzygalności bezpieczeństwa ogólnie i dla poszczególnych klas systemów. Drugi typ modelu opisuje, w jaki sposób system komputerowy stosuje elementy sterujące. Obowiązkowy model kontroli dostępu i uznaniowy model kontroli dostępu stanowią podstawę dla następujących modeli. Model kontroli dostępu kontrolowany przez twórcę łączy kontrolę danych z twórcą, a nie właścicielem, i ma oczywiste zastosowania w systemach zarządzania prawami cyfrowymi. Model kontroli dostępu oparty na rolach wykorzystuje funkcję zadania, a nie tożsamość, w celu zapewnienia kontroli, a zatem może realizować zasadę najmniejszych przywilejów skuteczniej niż wiele modeli. Kolejne kilka modeli opisuje poufność i integralność. Model Bell-LaPadula opisuje klasę systemów zaprojektowanych w celu ochrony poufności i był jednym z najwcześniejszych i najbardziej wpływowych modeli bezpieczeństwa komputerowego. Ścisła polityka integralności modelu Biba jest ściśle związana z modelem Bell-LaPadula i jest obecnie szeroko stosowana; jest on stosowany do programów, aby określić, kiedy można zaufać ich wynikom. Model Clarka-Wilsona jest również modelem integralności, ale różni się zasadniczo od modelu Biby, ponieważ model Clarka-Wilsona opisuje integralność pod względem procesów i zarządzania procesami, a nie pod względem atrybutów danych. Czwarty typ modelu to model hybrydowy. Model ściany chińskiej analizuje konflikty interesów i stanowi interesującą

mieszankę wymagań dotyczących poufności i integralności. Ten typ modelu powstaje, gdy wiele rzeczywistych problemów jest abstrakcyjnych w matematycznych reprezentacjach, na przykład podczas analizy zabezpieczeń wymaganych dla dokumentacji medycznej i dla procesu rejestrowania nieruchomości.

MODELE I BEZPIECZEŃSTWO.

Niektóre terminy powtarzają się w naszej dyskusji na temat modeli.

* Podmiot jest aktywną jednostką, taką jak proces lub użytkownik.

* Obiekt jest bierną jednostką, taką jak plik.

* Prawo opisuje, co podmiot może zrobić z obiektem; na przykład prawo do odczytu zezwala podmiotowi na odczytanie pliku.

* Stan ochrony systemu odnosi się po prostu do praw wszystkich podmiotów w systemie.

Dokładne znaczenie każdego prawa różni się w zależności od systemu. Na przykład w systemach Linux, jeśli proces ma uprawnienia do zapisu pliku, proces ten może zmienić zawartość pliku. Ale jeśli proces ma prawo zapisu do katalogu, ten proces może tworzyć, usuwać lub zmieniać nazwy plików w tym katalogu. Podobnie, posiadanie praw do odczytu w procesie może oznaczać, że posiadacz może uczestniczyć jako odbiorca komunikatów komunikacji międzyprocesowej pochodzących z tego procesu. Chodzi o to, że znaczenie praw zależy od interpretacji danego systemu. Przypisanie znaczenia prawom użytym w modelu matematycznym nazywa się tworzeniem modelu. Pierwszy model, który zbadamy, stanowi podstawę wielu prac nad podstawową trudnością analizy systemów w celu ustalenia, czy są one bezpieczne.

MODEL MACIERYZ KONTROLI DOSTĘPU.

Model macierzy kontroli dostępu jest prawdopodobnie najprostszym modelem bezpieczeństwa komputera. Składa się z macierzy, której wiersze odpowiadają tematom, a kolumny odpowiadają jednostkom (podmiotom i przedmiotom). Każdy wpis w macierzy zawiera zestaw praw, które podmiot (wiersz) ma nad jednostką (kolumną). Na przykład macierz kontroli dostępu w

	Process 1	Process 2	File 1	File 2
Process 1	own	read	read, execute	read, write, own
Process 2	write	own	read, write, execute, own	read

pokazuje system z dwoma procesami i dwoma plikami. Pierwszy proces ma własne prawa do siebie; prawa do odczytu w drugim procesie; odczytywać i wykonywać prawa do pierwszego pliku; oraz odczytywać, zapisywać i posiadać prawa do drugiego pliku. Drugi proces może pisać do pierwszego procesu; jest właścicielem; może czytać, pisać, wykonywać i posiadać pierwszy plik; i może odczytać drugi plik. Macierz kontroli dostępu rejestruje stan ochrony systemu. Ale systemy ewoluują; ich stan ochrony nie pozostaje stały. Zatem zawartość macierzy kontroli dostępu musi się zmienić, aby odzwierciedlić tę ewolucję. Prawdopodobnie najprostszym zbiorem reguł zmiany macierzy kontroli dostępu są te podstawowe operacje:

* Utwórz temat s tworzy nowy wiersz i kolumnę, obie oznaczone s

* Utwórz obiekt o tworzy nową kolumnę oznaczoną o

* Wpisz r w A [s, o] dodaje prawy r do wpisu w wierszu s i kolumnie o; Odpowiada to nadaniu podmiotowi prawa do podmiotu o

* Usuń r z A [s, o] usuwa prawe r z pozycji w wierszu s i kolumnie o; Odpowiada to usunięciu prawa podmiotu nad podmiotem o

* Zniszcz przedmioty s usuwa wiersz i kolumnę oznaczone s

* Zniszcz obiekt o usuwa kolumnę oznaczoną o

Operacje te można łączyć w polecenia. Kolejne polecenie tworzy plik f i daje procesowi p odczytanie i własne prawa do tego pliku:

command createread(p, f)

create object f

enter read into A[p, f]

enter own into A[p, f]

end.

Polecenie mono-operacyjne składa się z pojedynczej operacji podstawowej. Na przykład polecenie

command grantwrite(p, f)

enter write into A[p, f]

end.

który daje prawo do zapisu p na f, jest mono-operacyjny. Polecenia mogą zawierać warunki. Na przykład następne polecenie daje podmiotowi p uprawnienia do wykonywania pliku f, jeśli p ma prawa do odczytu nad f:

command grantexec(p, f)

if read in A[p, f] then

enter execute into A[p, f]

end.

Jeśli p nie ma praw do odczytu na f, gdy to polecenie jest wykonywane, nic nie robi. To polecenie ma jeden warunek, a więc jest nazywane monokondycyjnym. Komendy dwuwarstwowe łączą dwa warunki:

command copyread(p, q, f)

if read in A[p, f] and own in A[p, f] then

enter read into A[q, f]

end.

To polecenie daje podmiotowi q prawa do odczytu obiektu f, jeśli podmiot p jest właścicielem f i ma prawa do odczytu na f. Polecenia mogą mieć warunki tylko na początku, a jeśli warunek jest fałszywy,

polecenie kończy się. Polecenia mogą zawierać inne polecenia, a także operacje pierwotne. Jeśli wszystkie polecenia w systemie są monoperacyjne, mówi się, że system jest monooperacyjny; jeśli wszystkie polecenia są monoconditional lub biconditional, to system mówi się, że jest monoconditional lub biconditional, odpowiednio. Wreszcie, jeśli system nie ma poleceń, które używają usuwania lub niszczenia pierwotnych ozperacji, system jest uważany za monotoniczny. Macierz kontroli dostępu stanowi teoretyczną podstawę dla dwóch szeroko stosowanych mechanizmów bezpieczeństwa: list kontroli dostępu i list możliwości. W dziedzinie modelowania zapewnia narzędzie do analizy trudności w określeniu, jak bezpieczny jest system.

HARRISON, RUZZO I ULLMAN ORAZ INNE WYNIKI.

Pytanie o to jak sprawdzić, czy systemy są bezpieczne, ma kluczowe znaczenie dla zrozumienia bezpieczeństwa komputera. Zdefiniuj bezpieczne w najprostszy możliwy sposób: System jest bezpieczny w odniesieniu do ogólnego prawa r, jeśli to prawo nie może być dodane do jednostki w macierzy kontroli dostępu, chyba że kwadrat już go zawiera. Innymi słowy, system jest bezpieczny w odniesieniu do r, jeśli r nie może przeciec do nowego wpisu w macierzy kontroli dostępu. Pytanie staje się wtedy:

Pytanie bezpieczeństwa. Czy istnieje algorytm określający, czy dany system ze stanem początkowym secure jest bezpieczny w odniesieniu do danego prawa?

W ogólnym przypadku:

Twierdzenie . Pytanie bezpieczeństwa jest nierozstrzygalne. Dowodem jest ograniczenie problemu zatrzymania do pytania dotyczącego bezpieczeństwa. Oznacza to, że jeśli kwestia bezpieczeństwa byłaby rozstrzygalna, problem z zatrzymaniem byłby taki. Jednak nierozwiązywalność problemu zatrzymania jest dobrze znana, więc problem bezpieczeństwa musi również wystąpić być nierozstrzygalnym. Wyniki te oznaczają, że nie można opracować ogólnego algorytmu do określania, czy systemy są bezpieczne. Można to jednak zrobić w ograniczonych przypadkach i modele, które podążają za przykładami takich przypadków. Cechy, które muszą spełniać klasy systemów, aby pytanie bezpieczeństwa było rozstrzygalne, nie są jeszcze w pełni znane, ale dla określonych klas systemów można zadać pytanie dotyczące bezpieczeństwa. Na przykład:

Twierdzenie. Istnieje algorytm, który określi, czy systemy monoperacyjne są bezpieczne w odniesieniu do ogólnego prawa r.

Ale te klasy są wrażliwe na dozwolone polecenia:

Twierdzenie. Pytanie bezpieczeństwa dla systemów monotonicznych jest nierozstrzygalne.

Ograniczenie zestawu poleceń do poleceń dwuwarstwowych nie pomaga:

Twierdzenie. Pytanie bezpieczeństwa dla dwuwarstwowych systemów monotonicznych jest nierozstrzygalne. Ale ograniczając je do operacji monokondycyjnych:

Twierdzenie. Istnieje algorytm, który określi, czy monotoniczne systemy monokonserwacyjne są bezpieczne w odniesieniu do ogólnego prawa r. W rzeczywistości dodanie operacji usunięcia pierwotnego nie wpływa na ten wynik (chociaż dowód jest inny):

Twierdzenie. Istnieje algorytm, który określi, czy systemy monotoniczne, które nie używają pierwotnych operacji niszczenia, są bezpieczne względem ogólnego prawa r.

MODEL TYPED ACCESS-CONTROL

Wariant modelu macierzy kontroli dostępu dodaje typ do jednostek. Typowany model macierzy kontroli dostępu, nazywany TAM, wiąże typ z każdą jednostką i odpowiednio modyfikuje reguły manipulacji macierzą. Pojęcie to pozwala grupować jednostki w subtelniejsze kategorie niż tylko przedmiot i przedmiot i umożliwia nieco inną analizę niż sugeruje wynik HRU. W TAM zestaw reguł jest acykliczny, jeśli ani jednostka E, ani żaden z jej potomków nie mogą utworzyć nowej jednostki o tym samym typie co E. Biorąc pod uwagę tę definicję:

Twierdzenie. Istnieje algorytm, który określi, czy acykliczne, monotoniczne modele macierzy typowanej są bezpieczne w odniesieniu do ogólnego prawa r. System acykliczny i monotoniczny jest więc wystarczający do rozstrzygnięcia kwestii bezpieczeństwa. Ale nadal nie wiemy dokładnie, jakie właściwości są konieczne, aby rozstrzygnąć kwestię bezpieczeństwa. Zwracamy się teraz do modeli, które mają bezpośrednie zastosowanie do systemów i środowisk i koncentrują się na bardziej złożonych definicjach „bezpiecznego" i mechanizmów potrzebnych do ich osiągnięcia.

MODELE I KONTROLE.

Modele bezpieczeństwa komputerowego koncentrują się na kontroli: kto ma dostęp do plików i zasobów oraz jakie rodzaje dostępu są dozwolone. Kolejne charakterystyki tych elementów sterujących organizują je dzięki elastyczności użycia i rolom podmiotów kontrolujących dostęp. Są one niezbędne do zrozumienia, jak działają bardziej zaawansowane modele.

OBOWIĄZKOWE I UZNANIOWE MODEL KONTROLI DOSTĘPU

Metody kontroli dostępu są oparte na regułach; to znaczy użytkownicy nie mają nad nimi kontroli. Tylko system lub specjalny użytkownik (na przykład) oficer bezpieczeństwa systemu (SSO) może je zmienić. W ten sposób działa rządowy system klasyfikacji. Ktoś bez zezwolenia nie może czytać TOP SECRET materiału, nawet jeśli osoba, która ma dokument, chce na to zezwolić. Zasada ta jest nazywana obowiązkową, ponieważ musi być przestrzegana bez wyjątku. Przykładami innych obowiązkowych zasad są ogólnie prawa, których należy przestrzegać zgodnie z zapisami, a nie można zwolnić innej z odpowiedzialności za złamanie prawa; lub mechanizm kontroli dostępu oparty na pierścieniu Multics, w którym dostęp do segmentu danych poniżej dolnej granicy przedziału dostępu segmentu jest zabroniony niezależnie od uprawnień dostępu. Ten typ kontroli dostępu jest nazywany obowiązkową kontrolą dostępu, lub MAC. Reguły te opierają decyzję o dostępie na atrybutach podmiotu i obiektu (i ewentualnie innych informacji). Inne metody kontroli dostępu pozwalają właścicielowi jednostki kontrolować dostęp. Na przykład osoba, która prowadzi dziennik, decyduje, kto może go przeczytać. Nie musi nikomu tego pokazywać, a jeśli przyjaciel poprosi o jej przeczytanie, może odmówić. Tutaj właściciel umożliwia dostęp do dziennika według własnego uznania. Ten rodzaj kontroli nazywa się uznaniowym. Uznaniowa kontrola dostępu, czyli DAC, jest najpowszechniejszym rodzajem mechanizmu kontroli dostępu na komputerach. Kontrole mogą być (i często są) łączone. Gdy obowiązkowe i uznaniowe kontrole są łączone w celu egzekwowania jednolitej polityki kontroli dostępu, najpierw stosuje się obowiązkowe kontrole. Jeśli odmawiają dostępu, system odmawia dostępu, a kontrole dyskrecjonalne nigdy nie muszą być wywoływane. Jeśli obowiązkowe przepisy zezwalają na dostęp, konsultowane są kontrole uznaniowe. Jeśli oba zezwalają na dostęp, dostęp jest przyznawany.

MODEL KONTROLI DOSTĘPU KONTROLOWANY PRZEZ INICJATORA I DRM.

Inne rodzaje kontroli dostępu zawierają elementy obowiązkowej i uznaniowej kontroli dostępu. Kontrola dostępu kontrolowana przez inicjatora, lub ORCON, mechanizmów pozwala inicjatorowi określić, kto może uzyskać dostęp do zasobu lub danych. Rozważ dużą rządową agencję badawczą, która opracowuje badanie przewidywanej sprzedaży uchwytów do motyk na następny rok. Rynek

uchwytów motyki jest bardzo niestabilny i jeśli wyniki badania wycieką przedwcześnie, niektórzy sprzedawcy uzyskają ogromną przewagę rynkową. Ale badanie musi zostać rozesłane do agencji regulacyjnych, aby mogły przygotować odpowiednie przepisy, które będą obowiązywać po opublikowaniu badania. Tak więc agencja badawcza musi zachować kontrolę nad badaniem, nawet gdy krąży między innymi grupami. Dokładniej, kontrola dostępu kontrolowana przez twórcę spełnia dwa warunki. Załóżmy, że obiekt o jest oznaczony jako ORCON dla organizacji X. X postanawia zwolnić o podmiotom działającym w imieniu innej organizacji Y. Następnie

1. Podmioty, którym udostępniono kopie o, nie mogą zwolnić o podmiotom działającym w imieniu innych organizacji bez zgody X; i

2. Wszelkie kopie o muszą zawierać te ograniczenia.

Rozważ kontrolę, która implementuje te wymagania. Teoretycznie obowiązkowa kontrola dostępu mogłaby rozwiązać ten problem. W praktyce wymagane reguły muszą przewidywać wszystkie organizacje, którym dane będą udostępniane. Wymóg ten, w połączeniu z koniecznością posiadania oddzielnej reguły dla każdego możliwego zestawu obiektów i organizacji, które mają mieć dostęp do obiektu, czyni obowiązkową kontrolę dostępu, która spełnia wymagania nie do zrealizowania. Ale jeśli kontrola miała charakter uznaniowy, każdy podmiot, który otrzymał kopię badania, mógłby udzielić dostępu do swojej kopii bez zgody autora; więc kontrola dostępu kontrolowana przez twórcę nie jest ani uznaniowa, ani obowiązkowa. Jednak połączenie uznaniowych i obowiązkowych kontroli dostępu może wdrożyć tę kontrolę. Obowiązkowe mechanizmy kontroli dostępu zabraniają właścicielowi zmiany uprawnień dostępu do obiektu o i wymagają, aby każda kopia tego obiektu miała takie same uprawnienia kontroli dostępu, jak w przypadku o. Uznaniowa kontrola dostępu mówi, że twórca może zmienić uprawnienia kontroli dostępu do dowolnej kopii o. Jako przykład wykorzystania tego modelu w bardziej popularnym kontekście, firmy fonograficzne chcą kontrolować wykorzystanie swojej muzyki. Pojęciowo chcą zachować kontrolę nad muzyką po jej sprzedaży, aby uniemożliwić właścicielom rozpowszechnianie nieautoryzowanych kopii wśród znajomych. Tutaj twórcą jest wytwórnia płytowa, a chronionym zasobem jest muzyka. W praktyce kontrole dostępu kontrolowane przez twórców są trudne do wdrożenia technologicznego. Problem polega na tym, że mechanizmy kontroli dostępu zazwyczaj kontrolują dostęp do jednostek, takich jak pliki, urządzenia i inne obiekty. Jednak kontrola dostępu kontrolowana przez twórcę wymaga, aby kontrola dostępu była stosowana do informacji zawartych w byty - znacznie trudniejszy problem, dla którego nie ma jeszcze ogólnie akceptowanego mechanizmu.

OPARTE NA ROLACH MODELE KONTROLI DOSTĘPU I GRUPY

W prawdziwym życiu funkcja pracy często dyktuje uprawnienia dostępu. Księgowy biura ma bezpłatny dostęp do rachunków bankowych firmy, podczas gdy sprzedawcy nie. Jeśli Anne jest zatrudniona jako sprzedawca, nie może uzyskać dostępu do funduszy firmy. Jeśli później zostanie księgową, może uzyskać dostęp do tych funduszy. Dostęp jest więc uwarunkowany nie tożsamością osoby, ale rolą, jaką odgrywa osoba. Ten przykład ilustruje kontrolę dostępu opartą na rolach (RBAC) . Przypisuje zestaw ról, zwane autoryzowanymi rolami podmiotu, dla każdego podmiotu. W każdej chwili s mogą przyjąć co najwyżej jedną rolę, zwaną aktywną rolą s. Wtedy

Aksjomat. Reguła autoryzacji ról mówi, że aktywna rola s musi znajdować się w zestawie autoryzowanych ról s.

Ten aksjomat ogranicza się do przyjmowania ról, które może on przyjąć. Bez tego s mógłby przyjąć dowolną rolę, a więc robić wszystko. Rozszerzając ten pomysł, pozwólmy, aby predykat canexec (s, c) był prawdziwy, gdy podmiot s może wykonać polecenie c.

Aksjomat. Reguła przypisywania ról mówi, że jeśli canexec (s, c) jest prawdziwe dla dowolnych s i dowolnych c, to s musi mieć aktywną rolę.

Mówi to po prostu, że aby wykonać polecenie c, s musi mieć aktywną rolę. Bez takiej roli nie może wykonywać żadnych poleceń. Chcemy również ograniczyć polecenia, które można wykonać; robi to następny aksjomat.

Aksjomat. Reguła autoryzacji transakcji mówi, że jeśli canexec (s, c) jest prawdą, to tylko podmioty o tej samej roli co aktywna rola s mogą również wykonywać transakcja.

Oznacza to, że każda rola ma zestaw poleceń, które może wykonać, a jeśli c nie znajduje się w zestawie poleceń, które może wykonywać aktywna rola s, to s nie może jej wykonać. Jako przykład mocy tego modelu, rozważ dwa typowe problemy: powstrzymywanie ról i podziału obowiązków. Ograniczenie ról oznacza, że podwładny u ogranicza się do wykonywania ograniczonego zestawu poleceń, które może również wykonywać przełożony; przełożony może również wykonywać inne polecenia. Przydziel rolę a przełożonemu i roli b podwładnemu; ponieważ wszystko, co może zrobić podmiot z aktywną rolą b, podmiot z aktywną rolą może to zrobić, mówimy, że rola zawiera rolę b. Następnie możemy powiedzieć, że jeśli a jest autoryzowaną rolą s, a a zawiera b, to b jest również autoryzowaną rolą s. Biorąc to dalej, jeśli podmiot jest upoważniony do przyjęcia roli, która zawiera inne (podrzędne) role, może również przyjąć dowolną z podrzędnych ról. Oddzielenie obowiązku jest wymogiem, aby wiele podmiotów musiało połączyć wysiłki w celu wykonania zadania. Na przykład firma może wymagać od dwóch funkcjonariuszy podpisania czeku za ponad 50 000 USD. Pomysł polega na tym, że jedna osoba może naruszyć bezpieczeństwo, ale dwie osoby rzadziej łączą siły, aby złamać zabezpieczenia. Jednym ze sposobów radzenia sobie z rozdzieleniem obowiązków jest wymaganie, aby dwie różne role wypełniły zadanie i aby role wzajemnie się wykluczały. Dokładniej, niech r będzie rolą, a meauth (r), wzajemnie wyłączny zestaw autoryzacji r, to zestaw ról, których podmiot z uprawnioną rolą r nigdy nie może założyć. Następnie rozdzielenie obowiązków to:

Aksjomat. Zasada rozdzielenia obowiązków mówi, że jeśli rola a znajduje się w zbiorze meauth (b), to żaden podmiot, dla którego a jest uprawnioną rolą, może mieć b jako inną uprawnioną rolę.

Ta reguła jest stosowana do zadania, które wymaga ukończenia dwóch różnych osób. Zadanie jest podzielone na kroki, które mają wykonać dwie osoby. Każdej osobie przypisano oddzielną rolę, a każda rola należy do wzajemnie wykluczającego się zestawu autoryzacji drugiej. Zapobiega to wykonaniu zadania przez którąkolwiek osobę; muszą współpracować, każdy w swojej roli, aby go ukończyć. Role są podobne do grup, ale cele grup i ról są różne. Członkostwo w grupie jest definiowane przez zasadniczo arbitralne reguły, ustalane przez menedżerów systemu. Członkostwo w roli jest zdefiniowane przez funkcję zadania i jest powiązane z określonym zestawem poleceń, które są niezbędne do wykonania tej funkcji zadania. Zatem rola jest rodzajem grupy, ale grupa jest szersza niż rola i nie musi być związana z żadnym konkretnym zestawem

PODSUMOWANIE

Cztery rodzaje kontroli dostępu omówione w tej sekcji mają różne cele. Obowiązkowe, uznaniowe i kontrolowane przez twórcę mechanizmy kontroli dostępu są skoncentrowane na danych, określając dostęp na podstawie charakteru lub atrybutów danych. Kontrola dostępu w rolkach koncentruje się na potrzebach podmiotu. Różnica jest fundamentalna. Zasada najmniejszego przywileju mówi, że badani nie powinni mieć więcej przywilejów niż jest to konieczne do wykonywania ich zadań. Kontrola dostępu oparta na rolach, jeśli jest poprawnie zaimplementowana, robi to poprzez ograniczenie zestawu poleceń, które może wykonać podmiot. Pozostałe trzy elementy sterujące robią to, ustawiając atrybuty danych, aby kontrolować dostęp do danych, a nie ograniczając polecenia. Obowiązkowe kontrole

dostępu mają atrybuty ustawione przez oficera bezpieczeństwa systemu lub inny zaufany proces; uznaniowa kontrola dostępu przez właściciela obiektu; oraz kontrole dostępu kontrolowane przez twórcę, przez twórcę lub twórcę danych. Jak wspomniano, mechanizmy te można łączyć, aby sterowanie było łatwiejsze i bardziej precyzyjne w zastosowaniu. Omawiamy teraz kilka modeli, które to robią.

MODELE KLASYCZNE.

Trzy modele odegrały ważną rolę w rozwoju bezpieczeństwa komputerowego. Model Bell-LaPadula, jeden z najwcześniejszych formalnych modeli bezpieczeństwa komputerowego, wpłynął na rozwój wielu technologii bezpieczeństwa komputerowego i nadal jest szeroko stosowany. Biba, jego analog do integralności, odgrywa teraz ważną rolę w analizie programu. Model Clarka-Wilsona opisuje wiele praktyk handlowych w celu zachowania integralności danych. Analizujemy każdy z tych modeli w tej sekcji.

MODEL BELL + LAPADULA.

Model Bell-LaPadula jest formalizacją słynnego rządowego systemu klasyfikacji wykorzystującego poziomy UNCLASSIFIED, CONFIDENTIAL, SECRET i TOP SECRET. Zaczynamy od wykorzystania tych czterech poziomów, aby wyjaśnić idee leżące u podstaw modelu, a następnie zwiększyć te poziomy, aby przedstawić pełny model. Ponieważ model obejmuje wiele poziomów, jest to przykład wielopoziomowego modelu bezpieczeństwa. Czteropoziomowa wersja modelu zakłada, że poziomy są uporządkowane od najniższego do najwyższego jako UNCLASSIFIED, CONFIDENTIAL, SECRET i TOP SECRET. Obiekty są przypisane poziomy w oparciu o ich wrażliwość. Obiekt na wyższym poziomie jest bardziej czuły niż obiekt na niższym poziomie. Przedmioty są przypisywane poziomy w oparciu o obiekty, do których mają dostęp. Temat zostaje wyczyszczony do poziomu, a ten poziom nazywa się poświadczeniem bezpieczeństwa podmiotu. Obiekt jest klasyfikowany na poziomie, a poziom ten nazywany jest klasyfikacją bezpieczeństwa obiektu. Celem systemu klasyfikacji jest zapobieganie wyciekowi informacji lub spływaniu w dół (np. Podmiot w POUFNYM nie powinien mieć możliwości odczytu informacji sklasyfikowanych w TOP SECRET). Dla wygody piszemy poziom (-y) dla poświadczenia bezpieczeństwa podmiotu i klasyfikacji bezpieczeństwa obiektu (o) foran. Nazwa klasyfikacji nazywa się etykietą. Więc obiekt sklasyfikowany w TOP SECRET ma etykietę TOP SECRET. Załóżmy, że Tom jest wyczyszczony na poziomie SECRET. Trzy dokumenty, zwane papierami, artykułami i książkami, są klasyfikowane odpowiednio jako POUFNE, TAJNE i ŚCIŚLE TAJNE. Ponieważ odprawa Toma jest niższa niż klasyfikacja książki, nie może czytać książki. Ponieważ jego zezwolenie jest równe lub większe od klasyfikacji artykułu i papieru, może je przeczytać. Definicja. Prosta właściwość bezpieczeństwa mówi, że podmiot s może odczytać obiekt o jeśli i tylko wtedy, gdy poziom (o) ≤ poziom (y). Jest to czasami nazywane regułą braku odczytu i jest obowiązkową kontrolą dostępu. Jest to jednak niewystarczające, aby zapobiec przepływowi informacji w dół. Załóżmy, że Donna jest wyczyszczona na poziomie POUFNE. Dzięki prostej własności bezpieczeństwa nie może czytać artykułu, ponieważ poziom (artykuł) = SECRET> POUFNE = poziom (Donna). Ale Tom może przeczytać informacje w artykule i napisać je na papierze. A Donna może przeczytaj Papier. W ten sposób informacja SECRET wyciekła do tematu z POUFNYM zezwoleniem. Aby temu zapobiec, należy uniemożliwić Tomowi pisanie na papierze: Definicja. Właściwość that mówi, że podmiot s może napisać obiekt o, jeśli i tylko wtedy, gdy poziom (poziomy) ≤ poziom (o).

Jest to czasami nazywane regułą bez zapisu, a także obowiązkową kontrolą dostępu. Jest również znany jako właściwość gwiazdy i właściwość ograniczenia. Zgodnie z tą regułą, jako poziom (Tom) = SECRET> poziom (Papier), Tom nie może pisać na papierze. To rozwiązuje problem. Wreszcie, model Bell-LaPadula umożliwia właścicielom obiektów korzystanie z uznaniowej kontroli dostępu: Definicja.

Dyskrecjonalna właściwość bezpieczeństwa mówi, że podmiot s może odczytać obiekt o tylko wtedy, gdy wpis macierzy kontroli dostępu dla s i o zawiera prawo do odczytu. W celu ustalenia, czy Tom może czytać papier, system sprawdza prostą właściwość zabezpieczeń i uznaniowy problem bezpieczeństwa. Jako że oba dotyczą Tom i Paper, Tom może czytać Paper. Podobnie system sprawdza właściwość $*$, aby określić, czy Tom może pisać na papierze. Ponieważ właściwość $*$ nie obowiązuje dla Tom i Paper, Tom nie może pisać na papierze. Należy zauważyć, że uznaniowa właściwość bezpieczeństwa nie musi być sprawdzana, ponieważ odpowiednia obowiązkowa właściwość kontroli dostępu (właściwość $*$) odmawia dostępu. Twierdzenie o podstawowym bezpieczeństwie stwierdza, że jeśli system zaczyna się w bezpiecznym stanie i każda operacja jest zgodna z trzema właściwościami, to system pozostaje bezpieczny: Podstawowe twierdzenie bezpieczeństwa. Niech system secure ma bezpieczny stan początkowy $\sigma 0$. Ponadto niech każde polecenie w tym systemie będzie zgodne z prostą właściwością bezpieczeństwa, właściwością $*$ i uznaniową własnością bezpieczeństwa. Wtedy każdy stan σi, i ≥ 0, jest również bezpieczny. Możemy uogólnić to na dowolną liczbę poziomów. Niech L_0,…, L_n będzie zbiorem poziomów bezpieczeństwa uporządkowanych liniowo (tj. $L_0 <… <L_n$). Obowiązują proste właściwości zabezpieczeń, właściwość $*$ i uznaniowa własność bezpieczeństwa, podobnie jak podstawowe twierdzenie bezpieczeństwa. Dzięki temu możemy mieć o wiele więcej niż cztery opisane poziomy. Załóżmy teraz, że Erin pracuje dla Departamentu Europejskiego agencji rządowej, a Don pracuje dla Departamentu Azji dla tej samej agencji. Erin i Don są oczyszczeni na SECRET. Ale niektóre informacje, które Erin zobaczy, to informacje, których Don nie musi znać i odwrotnie. Wprowadzenie dodatkowych poziomów bezpieczeństwa nie pomoże tutaj, ponieważ wtedy Don mógłby odczytać wszystkie dokumenty, które Erin mogła, lub odwrotnie. Potrzebujemy alternatywnego mechanizmu. Alternatywny mechanizm to rozszerzenie idei „poziomu bezpieczeństwa". Definiujemy

kategoria jako rodzaj informacji. Komora bezpieczeństwa to para (poziom, zestaw kategorii) i odgrywa rolę, którą wcześniej spełniał poziom bezpieczeństwa. Załóżmy na przykład, że kategorią dla Departamentu Europejskiego jest EUR, a kategorią dla Departamentu Azji jest AZJA. Erin zostanie rozliczony w przedziale (SECRET, {EUR}), a Don w przedziale (SECRET, {ASIA}). Dokumenty mają również przedziały bezpieczeństwa. Dokument EurDoc może być sklasyfikowany jako (POUFNY, {EUR}), a papier AsiaDoc może być (TAJNE, {ASIA}). Dokument EurAsiaDoc zawiera informacje zarówno na temat Europy, jak i Azji, a więc znajduje się w przedziale (SECRET, {EUR, ASIA}). Tak jak poprzednio, piszemy poziom (Erin) = (SECRET, {EUR}), poziom (EurDoc) = (POUFNE, {EUR}) i poziom (EurAsiaDoc) = (SECRET, {EUR, ASIA}). Następnie musimy zdefiniować analog jako „większy niż". Jak wspomniano wcześniej, przedziały bezpieczeństwa nie są już uporządkowane liniowo, ponieważ nie można porównywać każdej pary przedziałów. Na przykład przedział Don nie jest „większy" niż Erin, a Erin nie jest „większy" niż Don. Ale klasyfikacja EurAsiaDoc jest wyraźnie „większa" niż u Dona i Erin. Porównujemy przedziały za pomocą relacji dom, ponieważ „dominuje". Definicja. Niech L i L ' będą poziomami bezpieczeństwa i niech C i C ' będą zestawami kategorii. Następnie (L, C) dom (L ', C ') wtedy i tylko wtedy, gdy L ' = L i C ' ⊆ C Relacja dom odgrywa rolę, jaką „większe lub równe" odegrało dla poziomów bezpieczeństwa. Kontynuując nasz przykład, poziom (Erin) = (SECRET, {EUR}), dom (POUFNE, {EUR}) = poziom (EurDoc) i poziom (EurAsiaDoc) = (SECRET, {EUR, ASIA}) dom (SECRET, {EUR}) = poziom (Erin). Przeformułowaliśmy teraz prostą właściwość bezpieczeństwa i właściwość $*$ w zakresie dom: Definicja. Prosta właściwość bezpieczeństwa mówi, że podmiot może odczytać obiekt o jeśli i tylko jeśli poziom (-y) dom poziom (o). Definicja. Właściwość that mówi, że podmiot s może pisać do obiektu o, jeśli tylko poziom (o) dom poziom (y). W naszym przykładzie załóżmy, że dowolna kontrola dostępu jest ustawiona tak, aby każdy podmiot mógł uzyskać dostęp do wszystkich typów. W takim przypadku, jako poziom (Erin) dom poziom (EurDoc), Erin może odczytać EurDoc (przez prostą właściwość bezpieczeństwa), ale nie pisać EurDoc (przez właściwość $*$). I odwrotnie, jako poziom (EurAsiaDoc) domena (Erin), Erin nie może czytać EurAsiaDoc (przez prostą właściwość bezpieczeństwa), ale może

pisać do EurAsiaDoc (przez właściwość ∗). Logiczne pytanie brzmi, jak określić najwyższy przedział bezpieczeństwa, który mogą czytać zarówno Erin, jak i Don, i najniższy, który mogą pisać. Aby to zrobić, musimy przejrzeć niektóre właściwości dom. Najpierw zwróć uwagę na poziom (poziomy) domeny; to znaczy dom jest zwrotny. Relacja jest także antysymetryczna, ponieważ jeśli poziom (y) dom poziom (o) i poziom (o) dom (y) poziom są prawdziwe, to poziom (y) = poziom (o). Jest przechodni, ponieważ jeśli poziom (s1) dom poziom (o) i poziom (o) dom poziom (s2), to poziom (s1) dom poziom (s2). Definiujemy również największe dolne ograniczenie (glb) dwóch przedziałów jako:

Definicja. Niech A = (L, C) i B = (L ', C '). Następnie glb (A, B) = (min (L, L '), C ∩ C '). To odpowiada na pytanie o najwyższy przedział bezpieczeństwa, w którym dwa podmioty s i s mogą czytać obiekt. Jest to glb (poziom (y), poziom (s)). Na przykład Don i Erin mogą czytać obiekty w:

glb (poziom (Don), poziom (Erin)) = (SECRET, χ;). Ma to sens, ponieważ Don nie może odczytać obiektu w żadnym przedziale oprócz tych z zestawem kategorii {ASIA} lub pustym zestawem, a Erin może tylko czytać obiekty w komora z zestawem kategorii {EUR} lub pustym zestawem. Oba są na poziomie TAJNE, więc przedział musi być również na poziomie TAJNE. Analogicznie możemy zdefiniować najmniejszą górną granicę (lub) dwóch przedziałów: Definicja. Niech A = (L, C) i B = (L ', C '). Następnie lub (A, B) = (max (L, L '), C ∪ C '). Możemy teraz określić najniższy przedział bezpieczeństwa, w którym mogą pisać dwie osoby. To jest lub (poziom (y), poziom (s ')). Na przykład Don i Erin mogą pisać do obiektów w:

glb ((poziom (Don), poziom (Erin)) = (SECRET, {EUR, ASIA}).

Ma to sens, ponieważ Don nie może pisać do obiektu w żadnym przedziale z wyjątkiem tych z ASIA w zestawie kategorii, a Erin może pisać tylko do obiektów w przedziale, w którym znajduje się zestaw kategorii. Najmniejszym zestawem kategorii spełniającym oba te wymagania jest {EUR, ASIA}. Oba są na poziomie TAJNE, więc przedział musi być również na poziomie TAJNE. Pięć właściwości (refleksyjne, antysymetryczne, przechodnie, istnienie najmniejszej górnej granicy dla każdej pary elementów i istnienie największej dolnej granicy dla każdej pary elementów) oznacza, że przedziały bezpieczeństwa tworzą strukturę matematyczną zwaną kratą. Ma to użyteczne właściwości teoretyczne i jest na tyle ważne, że modele wykazujące ten typ struktury nazywane są modelami kratowymi. Gdy model jest zaimplementowany w systemie, programiści często wprowadzają pewne modyfikacje. Zdecydowanie najczęstszym jest ograniczenie pisania do bieżącej komory lub do ograniczonego zestawu przedziałów. Zapobiega to zmianie informacji poufnych przez osoby, które nie mogą go odczytać. Strukturę modelu można również wykorzystać do implementacji zabezpieczeń przed złośliwymi programami, które zmieniają pliki, takie jak pliki binarne systemu. Aby temu zapobiec, umieść binaria systemu w przedziale zdominowanym przez przedziały przypisane do użytkowników. Dzięki prostej właściwości zabezpieczeń użytkownicy mogą odczytywać pliki binarne systemu, ale według właściwości ∗ użytkownicy nie mogą ich zapisywać. W związku z tym, jeśli wirus komputerowy infekuje programy lub dokumenty użytkownika, może rozprzestrzeniać się w przedziale użytkownika, ale nie w systemie plików binarnych. Model Bell-LaPadula jest podstawą kilku innych modeli. Badamy jeden z jego wariantów, który modeluje integralność, a nie poufność.

Model polityki ścisłej uczciwości Biby.

Ścisła polityka Biba dotycząca modelu uczciwości, zwykle zwany modelem Biby, jest matematycznym podwójnym modelem Bell-LaPadula. Rozważ kwestię wiarygodności. Gdy wysoce wiarygodny proces odczytuje dane z niezaufanego pliku i działa na podstawie tych danych, proces ten nie jest już godny zaufania - jak to się mówi, „śmieci, wyrzucanie śmieci". Ale jeśli proces odczytuje dane bardziej godne zaufania niż proces, wiarygodność tego procesu nie ulega zmianie. W gruncie rzeczy wiarygodność wyniku jest tak samo wiarygodna, jak najmniej wiarygodny proces i dane. Zdefiniuj zestaw klas integralności w taki sam sposób, w jaki zdefiniowaliśmy przedziały bezpieczeństwa dla modelu Bell-

LaPadula i pozwól, aby i-level (s) był przedziałem integralności s. Następnie powyższy tekst mówi, że „odczytuje” (wiarygodny proces odczytu danych niewiarygodnych) powinien zostać zablokowany, ponieważ zmniejsza wiarygodność procesu. Ale „czyta” jest dozwolone, ponieważ nie wpływa to na wiarygodność procesu. Jest to dokładnie przeciwieństwo prostej właściwości bezpieczeństwa. Definicja. Prosta właściwość integralności mówi, że podmiot s może odczytać obiekt o jeśli i tylko wtedy, gdy i-level (o) dom i-level (s). Ta definicja oddaje pojęcie zezwalania na „odczyt” i niedozwolone „odczytuje”. Podobnie, jeśli wiarygodny proces zapisuje dane w niewiarygodnym pliku, wiarygodność pliku może (lub nie) wzrosnąć. Ale jeśli niewiarygodny proces zapisuje dane w wiarygodnym pliku, spada wiarygodność tego pliku. s „zapisywać” powinno być dozwolone i „zapisywać” zabronione. Definicja. Właściwość ity -integrity mówi, że podmiot s może pisać do obiektu o, jeśli tylko i-level (s) dom i-level (o). Ta właściwość blokuje próby „zapisu”, pozwalając jednocześnie na „zapisywanie”. Trzecia właściwość dotyczy wykonywania podprocesów. Załóżmy, że data procesu chce wykonać czas polecenia jako podproces. Jeśli przedział wiarygodności daty dominuje nad przedziałem czasu, to każda data informacji przechodzi do czasu jest przekazywana do mniej wiarygodnego procesu, a zatem jest dozwolona w ramach właściwości integralności ity. Ale jeśli przedział integralności czasu dominuje nad przedziałem daty, wówczas naruszana jest własność $*$. Stąd:

Definicja. Właściwość integralności wykonania mówi, że podmiot s może wykonać temat s ’ wtedy i tylko wtedy, gdy i-level (s ’) dom i-level (s).

Biorąc pod uwagę te trzy właściwości, można pokazać:

Twierdzenie. Jeśli informacje mogą zostać przeniesione z obiektu o_1 do obiektu o_n, a następnie przez prostą właściwość integralności, właściwość ity-integrity i właściwość integralności wykonania, i-level (o_1) dom i-level (o_n). Innymi słowy, jeśli przestrzegane są wszystkie zasady modelu Biba, integralność informacji nie może zostać uszkodzona, ponieważ informacje nigdy nie mogą płynąć z obiektu mniej wiarygodnego do obiektu bardziej wiarygodnego.

Model ten sugeruje metodę analizy programów w celu zapobiegania naruszeniom bezpieczeństwa. Po uruchomieniu program odczytuje dane z różnych źródeł: samego siebie, systemu, sieci i użytkownika. Niektóre z tych źródeł są godne zaufania, takie jak sam proces i system. Użytkownik i sieć są pod kontrolą zwykłych użytkowników (lub użytkownicy zdalni) i dlatego są mniej godni zaufania. Zastosuj więc model Biba z dwoma przedziałami integralności, (UNTAINTED, Ø) (oznacza to, że zestaw kategorii w przedziale jest pusty) i (TAINTED, Ø), gdzie (UNTAINTED, Ø) dom (TAINTED, Ø). Dla wygody notacji będziemy pisać (UNTAINTED, Ø) jako UNTAINTED i (TAINTED, Ø) jako TAINTED; i dom jako ≥. W ten sposób UNTAINTED ≥ TAINTED. Technika działa zarówno w analizie statycznej, jak i dynamicznej, ale zazwyczaj jest używana do analizy dynamicznej. W tym trybie wszystkie stałe są przypisane etykiecie integralności UNTAINTED. Zmienne są przypisywane etykietom na podstawie przepływów danych w programie. Na przykład w przypisaniu etykieta integralności przypisywanej zmiennej jest ustawiana na etykiecie integralności przypisanego do niej wyrażenia. Gdy zmienne NIEZNANE i TAINTED są mieszane w wyrażeniu, etykieta integralności wyrażenia jest TAINTED. Jeśli zmiennej przypisano wartość z niezaufanego źródła, etykieta integralności zmiennej jest ustawiona na TAINTED. Gdy dane są używane jako (na przykład) parametry wywołań systemowych lub funkcji bibliotecznych, system sprawdza, czy etykieta integralności zmiennej dominuje nad etykietą parametru. Jeśli tak się nie stanie, program podejmuje pewne działania, takie jak przerwanie lub rejestrowanie ostrzeżenia lub rzucenie wyjątku. Ta akcja albo zapobiega exploitowi, albo ostrzega administratora ataku. Załóżmy na przykład, że programista chce zapobiec atakowi formatowania łańcucha. Jest to atak wykorzystujący lukę w funkcji drukowania C printf. Pierwszym argumentem printf jest ciąg formatujący, a zawartość tego ciągu określa, ile innych argumentów printf oczekuje. Poprzez manipulowanie zawartością ciągu formatującego osoba atakująca może nadpisać wartości zmiennych i uszkodzić stos, powodując

nieprawidłowe działanie programu - zazwyczaj na korzyść atakującego. Kluczowym krokiem ataku jest wprowadzenie nieoczekiwanej wartości dla ciągu formatu. Oto fragment kodu z wadą:

if (fgets (buf, sizeof (buf), stdin)! = NULL) printf (buf);

Odczytuje linię znaków z wejścia do tablicy buf i natychmiast drukuje zawartość tablicy. Jeśli dane wejściowe to „xyzzy% n", wówczas jakiś element stosu zostanie nadpisany wartością 5.24. Dlatego pierwszy parametr printf musi zawsze mieć klasę integralności UNTAINTED. W ramach tej techniki analizy, gdy fgets funkcji wejściowej jest wykonywany, zmiennej buf zostanie przypisana etykieta integralności TAINTED, ponieważ dane wejściowe (które są niezaufane) są w niej przechowywane. Następnie przy wywołaniu printf klasa integralności buf jest porównywana z klasą wymaganą dla pierwszego parametru printf. Ten pierwszy jest UCZCIWY; ten ostatni jest NIEDOSTĘPNY. Wymagamy jednak, aby klasa integralności zmiennej (TAINTED) dominowała w parametrze (UNTAINTED), a tutaj TAINTED ≤ UNTAINTED. Stąd analiza znalazła niedozwolony przepływ i działa odpowiednio.

MODEL CLARKA - WILSONA

Lipner zidentyfikował pięć wymagań dotyczących komercyjnych modeli uczciwości:

1. Użytkownicy nie mogą pisać własnych programów do manipulowania zaufanymi danymi. Zamiast tego muszą używać programów upoważnionych do dostępu do tych danych.

2. Programiści opracowują i testują programy na systemach nieprodukcyjnych, w razie potrzeby wykorzystując nieprodukcyjne kopie danych produkcyjnych.

3. Przeniesienie programu z systemów nieprodukcyjnych do systemów produkcyjnych wymaga specjalnego procesu.

4. Ten specjalny proces musi być kontrolowany i kontrolowany.

5. Kierownicy i audytorzy systemu muszą mieć dostęp do dzienników systemu i bieżącego stanu systemu.

Model Biba może zostać utworzony, aby spełnić pierwsze i ostatnie warunki poprzez odpowiednie przypisanie poziomów integralności, ale pozostałe trzy koncentrują się na integralności procesów. Dlatego też, mimo że model Biby działa dobrze w przypadku niektórych problemów z integralnością, nie spełnia tych wymagań dla komercyjnego modelu integralności. Model Clarka-Wilsona został opracowany w celu opisania procesów w wielu firmach komercyjnych. Istnieje kilka specjalistycznych terminów i pojęć potrzebnych do zrozumienia trybu Clarka-Wilsona; najlepiej je przedstawić na przykładzie:

* Rozważmy bank. Jeśli D są depozytami dziennymi, W wypłatami dziennymi, I kwotą pieniędzy na rachunkach bankowych na początku dnia, a F kwotą pieniędzy na rachunkach bankowych na koniec dnia, wartości te muszą spełniać ograniczenie I + D - W = F.

* Nazywa się to ograniczeniem integralności, ponieważ jeśli system (zestaw kont bankowych) go nie spełni, naruszona zostanie integralność banku.

* Jeśli system spełnia swoje ograniczenia integralności, mówi się, że jest w stałym stanie.

* Podczas działania system przechodzi z jednego spójnego stanu do drugiego. Operacje, które to robią, nazywane są dobrze utworzonymi transakcjami. Na przykład, jeśli klient przenosi pieniądze z jednego konta na drugie, transfer jest dobrze zrealizowaną transakcją. Jego działania składowe (wycofanie z

pierwszego konta i wpłata w drugim) pojedynczo nie są dobrze ukształtowanymi transakcjami, ponieważ jeśli tylko jedno się zakończy, system będzie w niespójnym stanie.

* Procedury sprawdzające, czy spełnione są wszystkie ograniczenia integralności, nazywane są procedurami integrityverification (IVP).

* Dane, które muszą spełniać ograniczenia integralności, nazywane są ograniczonymi elementami danych (CDI), a gdy spełniają ograniczenia, mówi się, że są w prawidłowym stanie.

* Wszystkie inne dane nazywane są nieograniczonymi danymi (UDI).

* Oprócz ograniczeń integralności danych, funkcje implementujące dobrze sformatowane transakcje są ograniczone. Muszą być certyfikowane, aby były dobrze uformowane i prawidłowo wdrożone. Taka funkcja nazywana jest procedurą transformacji (TP).

Model zawiera dziewięć zasad, z których pięć odnosi się do certyfikacji danych, a cztery z nich opisują, w jaki sposób wdrożenie modelu musi egzekwować certyfikaty. Pierwsza reguła ujmuje wymóg, aby system był w stanie spójnym:

Reguła certyfikacji 1. IVP musi zapewnić spójność systemu. Relacja certyfikowana wiąże pewien zestaw CDI z TP, który przekształca te CDI z jednego prawidłowego stanu w (prawdopodobnie inny) stan prawidłowy. Druga reguła to ujmuje.

Reguła certyfikacji 2. W przypadku niektórych zestawów powiązanych CDI TP przekształca te CDI z prawidłowego stanu w (prawdopodobnie inny) stan prawidłowy. Pierwszy przepis dotyczący egzekwowania zapewnia, że system śledzi certyfikowaną relację i uniemożliwia TP wykonanie z CDI spoza przypisanego zestawu certyfikatów:

Zasada egzekwowania 1. System musi utrzymywać certyfikowaną relację i zapewnia, że tylko TP certyfikowane do działania na CDI manipulują tym CDI. W typowej firmie zestaw użytkowników, którzy mogą korzystać z TP, jest ograniczony. Na przykład w banku kasjer nie może przenieść milionów dolarów z jednego banku do drugiego; wymaga to urzędnika bankowego. Druga reguła egzekwowania zapewnia, że tylko upoważnieni użytkownicy mogą uruchamiać TP na CDI, definiując relację dozwoloną przez użytkownika, TP i zestaw CDI, do których TP może uzyskać dostęp w imieniu tego użytkownika:

Zasada egzekwowania 2. System musi powiązać użytkownika z każdym TP i zestawem CDI. TP może uzyskać dostęp do tych CDI w imieniu powiązanego użytkownika. Jeśli użytkownik nie jest powiązany z konkretnym TP i zestawem CDI, TP nie może uzyskać dostępu do tych CDI w imieniu tego użytkownika. Oznacza to, że system może poprawnie identyfikować użytkowników. Następna reguła wymusza to:

Zasada egzekwowania 3. System musi uwierzytelnić każdego użytkownika próbującego wykonać TP. Zapewnia to, że tożsamość osoby próbującej wykonać TP jest prawidłowo powiązana z odpowiednią tożsamością użytkownika w komputerze. Forma uwierzytelniania zależy od instancji modelu, ponieważ różne środowiska sugerują różne wymagania uwierzytelniania. Na przykład urzędnik banku może użyć urządzenia biometrycznego i hasła do uwierzytelnienia się na komputerze, który przenosi miliony dolarów; kasjer, którego działania ograniczają się do mniejszych kwot, może tylko potrzebować podaj hasło. Omówiony już rozdział obowiązków jest kluczowym zagadnieniem w wielu operacjach komercyjnych. Model Clarka-Wilsona rejestruje go w następnej regule:

Zasada certyfikacji 3. Dozwolony związek musi spełniać wymogi nałożone przez rozdzielenie cła. Podstawową zasadą uczciwości handlowej jest to, że operacje muszą być kontrolowane. Wymaga to rejestrowania wystarczającej ilości informacji, aby określić, co zrobiła transakcja. Następna reguła ujmuje to wymaganie:

Reguła certyfikacji 4. Wszystkie TP muszą dołączać wystarczającą ilość informacji, aby zrekonstruować operację do dołączonego CDI. Jedynym dodatkiem CDI jest oczywiście dziennik. Do tej pory uważaliśmy wszystkie dane wejściowe do TP za CDI. Niestety jest to niemożliwe. W naszym przykładzie bankowy kasjer wprowadzi informacje o koncie oraz dane dotyczące depozytów i wypłat; ale to nie są CDI; kasjer może coś pomylić. Zanim TP będzie mogła skorzystać z tych informacji, musi zostać sprawdzona, aby upewnić się, że umożliwi to poprawne działanie TP. Ostatnia reguła certyfikacji obejmuje to:

Reguła certyfikacji 5. TP, która pobiera dane UDI jako dane wejściowe, musi wykonać dobrze sformatowaną transakcję lub nie zawierać żadnej transakcji dla żadnej wartości UDI. Zatem albo odrzuca UDI, albo przekształca go w CDI.

Obejmuje to również słabo spreparowane TP; jeśli dane wejściowe mogą wykorzystać luki w zabezpieczeniach TP, aby spowodować, że będzie działać w nieoczekiwany sposób, nie można uzyskać certyfikatu zgodnie z tą zasadą. W modelu leży możliwy konflikt. Zgodnie z powyższymi zasadami jeden użytkownik może poświadczyć, że TP działa na CDI, a następnie wykonuje TP na tym CDI. Problem polega na tym, że złośliwy użytkownik może poświadczyć TP, który nie wykonuje dobrze ukształtowanej transakcji, powodując naruszenie przez system ograniczeń integralności. Oczywiste jest, że zastosowanie zasady rozdziału obowiązków rozwiąże ten problem, a ostatnia reguła w modelu właśnie to robi:

Zasada egzekwowania 4. Jedynie osoba poświadczająca TP może zmienić poświadczoną relację dla tej TP. Ponadto żaden certyfikator TP ani żadnego CDI związanego z tą TP nie może wykonać TP na powiązanym CDI.

Oddziela to możliwość certyfikacji TP od możliwości wykonania TP i możliwość certyfikacji CDI dla danej TP z możliwości wykonania tej TP na tym CDI. Wymusza to wymóg rozdzielenia obowiązków. Teraz powróć do wymagań Lipnera dotyczących komercyjnych modeli uczciwości. TP odpowiadają programom Lipner i CDI do danych produkcyjnych. Aby spełnić wymaganie 1, certyfikatorzy Clark-Wilson muszą być zaufani, a zwykli użytkownicy nie mogą certyfikować ani TP, ani CDI. Następnie zasada egzekwowania 4 i zasada certyfikacji 5 wymuszają ten wymóg. Wymóg 2 jest spełniony, nie poświadczając programów rozwoju; ponieważ nie są one TP, nie można ich uruchomić na danych produkcyjnych. „Proces specjalny" w wymaganiu 3 to TP. Reguła certyfikacji 4 opisuje dziennik; specjalny proces w wymaganiu 3 będący TP, będzie dołączał informacje do dziennika, który może być kontrolowany. Co więcej, TP jest z definicji procesem kontrolowanym, a Zasada Egzekwowania 4 i Zasada Certyfikacji 5 kontrolują jego wykonanie. Przed instalacją instalowany program jest UDI; po zainstalowaniu jest to CDI (i TP). Tak więc spełniony jest wymóg 4. Wreszcie, model Clarka-Wilsona ma dziennik, który rejestruje wszystkie aspekty działania TP, a do którego dostęp mają menedżerowie i audytorzy. Mają także dostęp do stanu systemu, ponieważ mogą uruchomić IVP, aby sprawdzić jego integralność. Zatem wymóg 5 Lipnera jest spełniony. Tak więc model Clarka-Wilsona jest rzeczywiście zadowalającym modelem integralności handlowej. Ten model jest ważny z dwóch powodów. Po pierwsze, rejestruje sposób działania większości firm komercyjnych, w tym stosowanie rozdzielenia obowiązków (coś, czego model Biby nie ujmuje dobrze). Po drugie, rozdziela pojęcia certyfikacji i egzekwowania. Egzekwowanie zazwyczaj można wykonać w ramach tworzenia modelu. Ale model nie może wymusić sposobu certyfikacji; może jedynie wymagać, aby certyfikujący to zrobił. Dotyczy to oczywiście wszystkich modeli, ale model Clarka-Wilsona wyraźnie określa założenia dotyczące certyfikacji.

CHINSKI MODEL ŚCIANY. Czasami nazywany modelem Brewer-Nash, celem modelu ChineseWalljest zapobieganie konfliktom interesów. Czyni to poprzez grupowanie obiektów należących do tej samej

firmy w zestawy danych firmy i zestawy danych firmy w klasy konfliktu interesów. Jeśli dwie firmy (reprezentowane przez powiązane z nimi zestawy danych firmy) znajdują się w tej samej klasie konfliktu interesów, wówczas prawnik lub makler reprezentujący obydwa podmioty będą mieli konflikt interesów. Reguły modelu zapewniają, że podmiot może odczytać tylko jeden zestaw danych firmy w każdej klasie konfliktu interesów. Ogólnie rzecz biorąc, obiekty to dokumenty lub zasoby zawierające informacje, które firma chce (lub musi) zachować w tajemnicy. Istnieje jednak wyjątek. Firmy publikują dane publicznie w formie raportów rocznych; informacje te są starannie oczyszczane, aby usunąć wszystkie poufne treści. Aby odzwierciedlić praktykę biznesową, model musi umożliwiać wszystkim podmiotom zobaczenie tych danych. Model definiuje zatem klasę konfliktu interesów zwaną klasą odkażoną, która ma jeden zestaw danych firmowych zawierający tylko obiekty zawierające oczyszczone dane.

Rozważ teraz temat czytający obiekt. Jeśli podmiot nigdy nie przeczytał żadnego obiektu w klasie konfliktu interesów obiektu, odczytanie obiektu nie powoduje konfliktu interesów. Jeśli podmiot przeczytał obiekt w tym samym zestawie danych firmy, jedyne informacje, które podmiot widział w tej klasie konfliktu interesów, pochodzą od tej samej firmy, co obiekt, który próbuje odczytać, co jest dozwolone. Jeśli jednak podmiot przeczytał obiekt należący do tej samej klasy konfliktu interesów, ale inny zestaw danych firmy, to czy nowe żądanie odczytu zostało udzielone, podmiot przeczytałby informacje od dwóch różnych firm, w przypadku których występuje konflikt interesów - dokładnie to, co model próbuje zapobiec. To jest niedozwolone. Następna reguła podsumowuje to:

Definicja. Prosta własność CW mówi, że podmiot może odczytać obiekt o, jeśli tylko:

1. s nie przeczytał żadnego innego przedmiotu w klasie konfliktu interesów; lub

2. Jedyne przeczytane obiekty w klasie konfliktu interesów znajdują się w zestawie danych firmy.

Aby zobaczyć, dlaczego to działa, załóżmy, że wszystkie banki są w tej samej klasie konfliktu interesów. Makler giełdowy reprezentuje Big Bank. Podchodzi do reprezentowania The Bigger Bank. Gdyby się zgodził, potrzebowałaby dostępu do informacji The Bigger Bank, a konkretnie informacji obiektu w zestawie danych firmy The Bigger Bank. Ale to oznaczałoby, że mogłaby czytać obiekty z dwóch zestawów danych firmowych w tej samej klasie konfliktu interesów, coś, czego nie może zabezpieczyć przez własność bezpieczeństwa CW. Ważny jest element czasowy modelu; nawet jeśli zrezygnował z reprezentacji Big Banku, nie może reprezentować Big Banku, ponieważ warunek 2 prostej zabezpieczeń CW uwzględnia wszystkie obiekty, które wcześniej przeczytała. Ma to sens, ponieważ miał dostęp do Big Banku i mogła w sposób niezamierzony naruszyć interesy poprzedniego pracodawcy reprezentując The Bigger Bank. Właściwość bezpieczeństwa CW-simple niejawnie mówi, że s może odczytać dowolny oczyszczony obiekt. Aby to zobaczyć, zauważ, że jeśli s nigdy nie przeczytał oczyszczonego obiektu, warunek 1 jest zachowany. Jeśli ma odczytać oczyszczony obiekt, a następnie warunek 2, ponieważ wszystkie oczyszczone obiekty znajdują się w tym samym zestawie danych firmowych. Pisanie stanowi kolejny problem. Przypuśćmy, że Barbara reprezentuje Big Bank, a Percival pracuje dla Big Banku. Obaj reprezentują również The Biggest Toy Company, która - nie będąc instytucją finansową - znajduje się w innej klasie konfliktu interesów niż jakikolwiek bank. W związku z tym nie ma konfliktu interesów w reprezentacji banku i firmy zabawkarskiej Barbary lub Percivala. Ale istnieje ścieżka, wzdłuż której informacje mogą płynąć z Barbary do Percival i odwrotnie, co umożliwia wystąpienie konfliktu interesów. Barbara może odczytać informacje z obiektu w zestawie danych firmy Big Bank i zapisać go w obiekcie w zestawie danych firmy The Biggest Toy Company. Percival może odczytać informacje z obiektu w zestawie danych firmy The Biggest Toy Company, tym samym skutecznie dając mu dostęp do informacji The Big Bank - co stanowi konflikt interesów. To, że potrzebuje pomocy Barbary, nie umniejsza problemu. Cel modelu wymaga zapobieżenia tej konspiracji. Następna reguła to:

Definicja. Właściwość CW - ∗ - mówi, że podmiot może pisać do obiektu o, jeśli i tylko wtedy, gdy spełnione są oba następujące warunki:

1. Prosta ochrona CW pozwala s czytać o; i

2. Wszystkie niesanitarne obiekty, które można odczytać, znajdują się w tym samym zestawie danych firmy co 0.

Teraz Barbara może czytać obiekty zarówno w zestawie danych firmy The Big Bank, jak i w zestawie danych The Biggest Toy Company. Ale gdy próbuje napisać do zestawu danych The Biggest Toy Company, właściwość CW - ∗ - uniemożliwia jej to, ponieważ warunek 2 nie jest spełniony (ponieważ może odczytać obiekt w zestawie danych firmy The Big Bank). To również dotyczy oczyszczonych obiektów. Załóżmy, że Skyler reprezentuje największą firmę z branży zabawek i żadną inną firmę. Potrafi także czytać informacje z klasy odkażonej. Kiedy próbuje napisać do obiektu w zestawie danych firmy The Biggest Toy Company, spełnia oba warunki właściwości CW-simple security (ponieważ czytał tylko obiekty w tym zestawie danych firmy) oraz wszystkie niezasanalizowane obiekty, które może odczytać są w tym samym zestawie danych firmy, co obiekt, który może odczytać. Zatem oba warunki właściwości CW - ∗ - są spełnione, więc Skyler może zapisać obiekt. Warunki CW - ∗ - własność są bardzo restrykcyjne; w rzeczywistości podmiot może pisać do obiektu tylko wtedy, gdy ma dostęp do zestawu danych firmy zawierającego ten obiekt i nie ma innego zestawu danych firmy, z wyjątkiem zestawu danych firmy w klasie oczyszczonej. Ale bez tego ograniczenia możliwe są konflikty interesów.

PODSUMOWANIE.

Cztery modele omówione w tej sekcji odegrały kluczową rolę w rozwoju naszego rozumienia bezpieczeństwa komputerowego. Chociaż nie jest to pierwszy model poufności, model Bell-LaPadula opisuje szeroko stosowany schemat bezpieczeństwa. Model Biba uchwycił pojęcia „zaufania" i „wiarygodności" intuicyjny sposób, a ostatnie postępy w analizie programów pod kątem luk w zabezpieczeniach zastosowały ten model z doskonałym skutkiem. Model Clarka-Wilsona przeniósł pojęcie komercyjnych modeli integralności z modeli wielopoziomowych na modele, które badają integralność procesu, jak również integralność danych. Model chińskiego muru zbadał konflikt interesów, obszar, który często powstaje, gdy wykonuje się usługi poufne dla wielu firm lub ma dostęp do poufnych informacji od wielu firm. Modele te są uważane za klasyczne, ponieważ ich struktura i pomysły leżą u podstaw zasad i struktur wielu innych modeli.

INNE MODELE

Niektóre modele badają specyficzne środowiska. Model bezpieczeństwa systemów informacji klinicznej uwzględnia ochronę dokumentacji zdrowotnej, podkreślając odpowiedzialność, jak również poufność i integralność. Traducement opisuje proces rejestracji nieruchomości, który wymaga ścisłej definicji uczciwości i odpowiedzialności z zachowaniem poufności w niewielkim lub żadnym stopniu. Inne modele uogólniają klasyczne modele. Najbardziej znane są modele bezpieczeństwo nieinterferencyjne i bezpieczeństwo dedukcyjności. Oba są wielopoziomowymi modelami bezpieczeństwa z dwoma poziomami, WYSOKIM i NISKIM. Model nieinterferencyjny30 definiuje bezpieczeństwo jako zdolność obiektu HIGH do zakłócania tego, co widzi obiekt LOW. Na przykład, jeśli podmiot WYSOKI może uniemożliwić NISKIM podmiotom uzyskanie zasobu w określonym czasie, podmiot WYSOKI może przesłać informacje do obiektu NISKIEGO. Zasadniczo, interferencja jest formą zapisu i należy jej zapobiec, tak jak model Bell-LaPadula zapobiega temu, aby WYSOKI obiekt zapisywał do NISKIEGO obiektu. Model dedukcyjności sprawdza, czy NISKI podmiot może wywnioskować cokolwiek o działaniach WYSOKIEGO podmiotu, badając tylko wyjścia LOW. Oba te modele są przydatne w analizie bezpieczeństwa systemów i mechanizmów wykrywania włamań, a

doprowadziły do pracy, która wykazała, że połączenie dwóch bezpiecznych systemów obliczeniowych może stworzyć niezabezpieczony system. Dalsze prace koncentrują się na ustaleniu warunków, w których połączenie dwóch bezpiecznych systemów tworzy bezpieczny system.

WNIOSEK.

Skuteczność modelowania matematycznego zależy od aplikacji tych modeli. Zazwyczaj modele przechwytują szczegóły specyficzne dla systemu i opisują ograniczenia zapewniające bezpieczeństwo systemu lub informacje w systemie. Jeśli model nie przechwytuje poprawnie szczegółów całego systemu, wyniki mogą nie być wyczerpujące, a analiza może przeoczyć sposoby, w jakie bezpieczeństwo może być zagrożone. To jest ważna kwestia. Na przykład model Bell-LaPadula zawiera wyobrażenie o tym, co system musi zrobić, aby zapobiec usunięciu tematu dla informacji o wycieku TOP SECRET do podmiotu, dla którego określono POUFNE. Ale jeśli to system wymusza model, temat TOP SECRET może nadal spełniać temat POUFNY i przekazać mu drukowaną wersję informacji TOP SECRET. To jest poza systemem, więc nie został przechwycony przez model. Ale jeśli model obejmuje również procedury, wówczas procedura jest niezbędna, aby zapobiec temu „zapisywaniu". W takim przypadku wadą byłaby implementacja procedury, która nie zapobiegła przekazywaniu informacji – innymi słowy nieprawidłowa instancja modelu, dokładnie to, o czym mówi komentarz Dorothy Denning we wstępie do tej sekcji. Modele opisane w tej sekcji obejmują fundamentalny (model macierzy kontroli dostępu) do zastosowanego (Bell-LaPadula, Biba, Clark-Wilson i ChineseWall). Wszystkie odgrywają rolę w pogłębianiu naszej wiedzy na temat bezpieczeństwa i sposobu jego egzekwowania. Obszar modelowania matematycznego jest obszarem bogatym i ważnym. Stanowi podstawę do wykazania, że projektowanie systemów jest bezpieczne dla określonych definicji bezpieczeństwa. Bez tych modeli nasze zrozumienie sposobu zabezpieczenia systemów byłoby ograniczone.