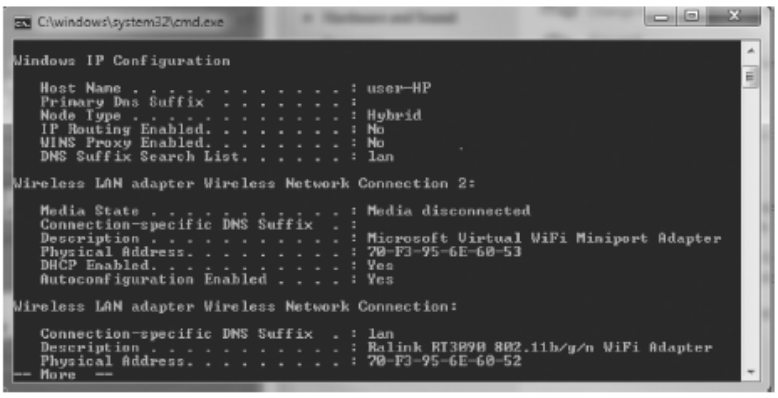
Luki w systemach informatycznych

Cyberprzestrzeń: od terra incognita do terra nullius.

Wprowadzenie

Luka w zabezpieczeniach dowolnego systemu jest wynikiem celowego lub niezamierzonego pominięcia lub nieumyślnego błędu projektowego, który bezpośrednio lub pośrednio prowadzi do kompromisu w zakresie dostępności, integralności lub poufności systemu. W ramach zapewniania informacji luki mogą ukrywać się na każdym poziomie bezpieczeństwa, czy to bezpieczeństwa dostępu do informacji, bezpieczeństwa komputera i pamięci masowej, bezpieczeństwa komunikacji czy bezpieczeństwa operacyjnego i fizycznego. W przypadku systemów informatycznych głównymi komponentami są ludzie, sprzęt i oprogramowanie. Z tego względu należy poszukiwać luk w każdym z tych trzech obszarów.

Ponad pół wieku temu projektanci, inżynierowie i naukowcy z powodzeniem skwantyfikowali pojęcie niezawodności w zakresie projektowania i konserwacji sprzętu i oprogramowania w mniejszym stopniu. Obecnie podejmowane są wysiłki w celu oszacowania abstrakcyjnej koncepcji podatności na zagrożenia, która dotyczy bezpieczeństwa systemów informatycznych. Celem jest wyrażenie postrzeganego poziomu bezpieczeństwa w sposób mierzalny, ustandaryzowany i zrozumiały oraz poprawa „… mierzalności bezpieczeństwa poprzez wyliczenie podstawowych danych bezpieczeństwa, zapewnienie znormalizowanych języków jako środków do dokładnego przekazywania informacji i zachęcanie do udostępnianie informacji użytkownikom poprzez tworzenie repozytoriów ”. Luki w zabezpieczeniach można ukryć w danych, kodzie, a najczęściej w procesach, które przypadkowo umożliwiają nieautoryzowany dostęp. Włamania mogą jednak wystąpić nie tylko w Internecie, ale także w intranetach, w których najczęściej bezpieczeństwo nie jest tak silne. Bezpieczeństwo można wzmocnić dzięki inteligentnym mechanizmom uwierzytelniania stosowanym na obu końcach - po stronie użytkownika, jak i po stronie serwera. Po stronie użytkownika uwierzytelnianie można znacznie wzmocnić dzięki wprowadzeniu dodatkowych mechanizmów, takich jak hasła jednorazowe (OTP), zapewnianych przez kanały wewnątrz- lub dodatkowe. Takimi kanałami mogą być dane biometryczne, kwestionariusze lub dodatkowe przejrzyste parametry związane z numerami identyfikacyjnymi urządzenia użytkownika, takie jak numer seryjny producenta, Media Access Control (MAC) lub International Mobile Equipment Identity (IMEI). MAC, zwany także adresem fizycznym, to 48-bitowa liczba, wyrażona jako 12 cyfr szesnastkowych, która jednoznacznie identyfikuje interfejs sieciowy komputera. Obwód interfejsu sieciowego może być wkładaną kartą sieciową lub może być osadzony w płycie głównej komputera. Rysunek pokazuje, jak można zidentyfikować adres MAC komputera osobistego.



IMEI w podobny sposób jednoznacznie identyfikuje urządzenia korzystające z telefonii komórkowej i jest liczbą, zwykle o długości od 13 do 15 cyfr. Dostawcy telefonii komórkowej przypisują numer telefonu urządzeniom łączącym je z ich identyfikatorem IMEI. Rycina 1.3 ilustruje dostępne numery IMEI wewnątrz telefonów komórkowych.

Oprócz dostępnych numerów MAC i IMEI do uwierzytelnienia można również używać numerów seryjnych urządzenia i parametrów sieci, takich jak adresy intranetowe i internetowe. Powyższe dotyczy uwierzytelniania klienta w stosunku do serwera. Po stronie serwera użycie certyfikatów, ograniczeń IP i enkapsulacji danych może znacznie poprawić uwierzytelnianie i bezpieczeństwo. Podczas transportu dane mogą być chronione kodami skrótu, takimi jak cykliczny kod redundancji (CRC), oraz szyfrowaniem klucza prywatnego / klucza publicznego , mechanizmy Luki w systemach informatycznych mogą wynikać z bardzo różnych przyczyn, od penetracji zapory ogniowej i ataków koni trojańskich po decentralizację i statyczny przydział zasobów. Najczęściej luki są wprowadzane podczas aktualizacji lub dostosowywania systemów do nowych środowisk operacyjnych.

Przyczyny podatności

W kontekście cyberbezpieczeństwa podatność jest brakiem, który może spowodować obniżenie wydajności lub awarię systemu. Taki niedobór może występować w samej aplikacji lub w komputerze hosta, w sieci, a nawet w brakach szkoleniowych użytkownika.

Dlatego aplikacje muszą być stale sprawdzane. Parametry aplikacji, oprócz samego kodu oprogramowania, obejmują interfejsy z systemem hostingu, a także interfejsy z użytkownikami. Dlatego kod należy zastąpić zaktualizowanym, należy wziąć pod uwagę nowe cechy systemu hostingu i poprawić umiejętności użytkownika. Poprawki dostarczone przez programistę należy natychmiast zastosować. Ponadto aplikacje zaprojektowane dla określonej wersji systemu operacyjnego niekoniecznie muszą działać z inną wersją wcześniejszą lub późniejszą. Chociaż sama aplikacja może być bezpieczna, jej zdalna dostępność może mieć słabą kontrolę, taką jak hasła lub inne mechanizmy uwierzytelniania, tworząc w ten sposób lukę w systemie. Często sieć organizacji jest wypełniona aplikacjami, które nieumyślnie i przypadkowo umożliwiają dostęp krzyżowy. Oznacza to, że legalne

wejście do jednej aplikacji umożliwia wejście backdoora do innej. W takim przypadku konieczna jest segmentacja sieci w celu wyeliminowania takiej podatności. Na przykład sieć operacji finansowych i punkty dostępu powinny być fizycznie oddzielone od dostępu użytkowników do informacji o produktach. Należy zdać sobie sprawę, że ochrona, czy to dla kraju, domu czy bazy danych, ma wiele parametrów, z których każdy ma swoją cenę. Tak więc każda tarcza ochronna musi przejść analizę opłacalności, aby zoptymalizować alokację zasobów organizacji. Luki można tworzyć z wnętrza organizacji. Nazywamy to zagrożeniem dla osób z wewnątrz. Takie zagrożenie niekoniecznie musi być złośliwe. Przeważnie takie podatności wynikają z dwóch następujących głównych kategorii:

• Brak szkolenia w zakresie cyberbezpieczeństwa dla personelu

• Niedojrzały osąd, szczególnie w zakresie cyberbezpieczeństwa środków

Wewnętrzne szkolenie w zakresie bezpieczeństwa cybernetycznego dla wszystkich osób uzyskujących dostęp do sieci jest podobne do uzyskiwania prawa jazdy, z tą różnicą, że szkolenie w zakresie bezpieczeństwa cybernetycznego wymaga aktualizacji w miarę ewolucji zagrożeń cybernetycznych.

Pomiar podatności

US National Institute of Standards and Technology (NIST) opracował protokół znormalizowanej klasyfikacji i oceny zawartości zabezpieczeń w systemie oprogramowania w celu „ujednolicenia sposobu identyfikacji i katalogowania podatności na zagrożenia i informacji o konfiguracji". Protokół nosi nazwę Security Content Automation Protocol (S'CAP, wymawia się „es-cap") i składa się z następujących sześciu składników.

1. Typowe luki w zabezpieczeniach i narażenia (CVE). Jest to depozyt znanych zarejestrowanych luk w zabezpieczeniach informacji, w których każde wystąpienie ma swój własny, niepowtarzalny numer identyfikacyjny. Początkowo takie zdarzenie jest definiowane jako potencjalna luka, a jeśli zostanie zaakceptowane, staje się wpisem zarejestrowanym na liście MITER CVE i ostatecznie zarejestrowanym w National Vulnerability Database (NVD). Pod koniec lata 2010 r. NVD zawierało 43 163 luki w zabezpieczeniach CVE, rosnąc w tempie 11 luk dziennie. W tej bazie danych można znaleźć słabe punkty bezpieczeństwa wielu programów, w tym dobrze znanych systemów operacyjnych i przeglądarek internetowych.

2. Common Configuration Enumerator (CCE). Jest to podobny depozyt, ale zawiera luki w zabezpieczeniach i niespójności interfejsu występujące w konfiguracjach systemu. Takie informacje mogą pomóc w zapewnieniu zgodności z przepisami, w określeniu właściwej interoperacyjności, a także w kontrolach audytu. Dostarczone informacje są w formie narracyjnej identyfikującej istniejące problemy i zwykle zalecającej rozwiązania.

3. Common Platform Enumerator (CPE). Ten komponent protokołu dotyczy prawidłowej nazewnictwa oprogramowania i oferuje hierarchiczną strukturę. W ten sposób oprogramowanie jest zdefiniowane jawnie, co znacznie ułatwia zarządzanie zapasami oprogramowania

4. Common System Vulnerability Scoring System (CVSS) Jest to algorytm, który bierze pod uwagę parametry związane z opracowaniem i użyciem przedmiotowego oprogramowania i zapewnia wynik wyrażający poziom obliczonego bezpieczeństwa. Dostarczony algorytm jest dostępny do użytku bez żadnych kosztów i cieszy się powszechnym stosowaniem przez projektantów systemów i analityków bezpieczeństwa wykonujących analizy ryzyka i planowanie systemu. Kalkulator CVSS dostępny online implementuje opracowany algorytm

5. Format opisu rozszerzalnej listy kontrolnej konfiguracji (XCCDF). Jest to szablon XML, który ułatwia przygotowanie znormalizowanych dokumentów zawierających wytyczne bezpieczeństwa, które przedstawiają luki w zabezpieczeniach lub obawy związane z bezpieczeństwem całego oprogramowania lub konkretnych konfiguracji lub zastosowań oprogramowania adresowanego, w znormalizowanym „ zawartość konfiguracyjna za pomocą zautomatyzowanych narzędzi bezpieczeństwa ".

6. Otwarty język luk w zabezpieczeniach i język oceny (OVAL). Służy to jako wspólny wątek „w całym spektrum narzędzi i usług związanych z bezpieczeństwem informacji ... (i)… standaryzuje trzy główne etapy procesu oceny", a mianowicie reprezentację informacji o systemie, wyrażenie konkretnego stany maszyny, a na koniec raportowanie oceny, wszystko w języku wspólnym dla społeczności systemów bezpieczeństwa. Przedsiębiorstwa używają OVAL do szerokiej gamy kluczowych funkcji, w tym oceny podatności, zarządzania konfiguracją, zarządzania poprawkami, zasad zgodności, dokumentacja porównawcza i automatyzacja zawartości bezpieczeństwa. „Protokół Security Content Automation Protocol (S'CAP) to synteza interoperacyjne specyfikacje pochodzące z pomysłów społeczności. " Poza S'CAP inne inicjatywy zapewniają normalizację w dodatkowych obszarach. Należą do nich: Common Weaknesses Enumerator (CWE), Common Malware Enumerator (CME), Common Weaknesses Scoring System (CWSS), Malware Attack Enumeration and Characterization (MAEC), oraz Common Attack Enumeration and Classification (CAPEC)). Łącznie powyższe standardy składają się na bardzo potężną infrastrukturę do oceny podatności systemów informatycznych i raportowania. W związku z tym podatności, słabości i złośliwe oprogramowanie można opisać w dokładny, znormalizowany, ilościowy i wyraźny sposób. Korzystając z technologii S'CAP, systemy informacyjne można klasyfikować pod względem poziomu bezpieczeństwa w sposób, który jest standardowy i ostatecznie akceptowany w całej branży.

Unikanie luk w zabezpieczeniach dzięki bezpiecznemu kodowaniu

W kulturze tworzenia oprogramowania, jeszcze nie tak dawno temu, celem projektowania było tworzenie efektywnych programów z minimalną liczbą linii kodu lub wykonywanie w minimalnym czasie lub kombinacja obu. Odzwierciedlając wysoki koszt pamięci i niską prędkość procesorów, cennymi towarami były przestrzeń pamięci i szybkość wykonywania. Tolerancja błędów dla integralności danych często przychodziła na myśl projektantom, ale nigdy dodatkowego kodu dla bezpieczeństwa, ponieważ wszystkie systemy były autonomiczne i fizycznie odizolowane od siebie. W dużych projektach oprogramowania luki można zminimalizować, dzieląc kod i dane na sekcje rezydentne i przejściowe, zgodnie z zasadami projektowania systemu operacyjnego. Oznacza to, że gdy wywoływana jest aplikacja, procedury i dostęp do danych są wprowadzane tak, jak tego wymaga natychmiastowa potrzeba, zamiast gromadzenia całego spisu aplikacji. W ten sposób, jeśli dojdzie do ingerencji złośliwego oprogramowania, szkody zostaną uwzględnione w pobranej (załadowanej pamięci) części aplikacji. Dzisiaj, ogólnie mówiąc, ani przestrzeń pamięci, ani szybkość przetwarzania nie wydają się już ograniczeniami programowymi. Ta era minęła nieodwracalnie, a jej miejsce zajął świat szybkich połączeń wzajemnych. Internet oferuje tak wysoki poziom użyteczności, że każdy możliwy do podłączenia sprzęt, od telefonów komórkowych po superkomputery, chce zostać podłączony. Podany poziom tego zaawansowanego narzędzia i wygody skutkuje zwiększoną produktywnością, która często przesłania towarzyszące ryzyko bezpieczeństwa. Eksperci otwarcie mówią: „Internet jest wrogim środowiskiem, dlatego musisz ... [być w stanie] wytrzymać atak [s, aby przetrwać]", i jest często określany jako Wild Wild Web, WWW. W związku z tym kod, który będzie używany w sieci, musi zostać zaprojektowany z myślą o możliwości wrogich ataków, dokładnie w taki sam sposób, jak budynek musi być zaprojektowany i zbudowany tak, aby był odporny na trzęsienia ziemi. Kryterium tego, co stanowi dobry kod, zmieniło się teraz z minimalnego kodu na minimalnie

podatny na atak. Istnieje popularne greckie przysłowie, które mówi: „Lepiej jest spędzać czas na zabezpieczaniu osła na drzewie niż na poszukiwaniu zagubionego osła”. Dotyczy to bezpośrednio projektowania oprogramowania z dostępem do Internetu. Lepiej jest spędzać czas na projektowaniu bezpiecznego oprogramowania, niż stawiać czoła konsekwencjom włamań - kosztowny rozwój łatek, zła reklama itp. Innymi słowy, dzisiaj pojęcia jakości oprogramowania i bezpieczeństwa oprogramowania są ze sobą powiązane. Dlatego jeśli specyfikacja oprogramowania wymaga ochrony, musi istnieć odpowiedni mechanizm bezpieczeństwa. Coraz częściej obserwuje się, że aplikacje internetowe mają własne zapory ogniowe zamiast polegać wyłącznie na systemach hostingu. Koszt usunięcia podatności jest zwykle bardzo wysoki, jeśli weźmie się pod uwagę powstające szkody niematerialne. Podczas naprawy zasoby będą musiały zostać odebrane innym zadaniom, aby zająć się tą sprawą w bardzo pilny sposób. Pod względem pieniężnym koszt usunięcia luki w zabezpieczeniach wynosi od pięciu do siedmiu cyfr w dolarach, w zależności od głębokości projektu tej luki. Procedura niezbędna do usunięcia luki zazwyczaj wykonuje następujące kroki:

1. Lokalizacja źródła luki

2. Zaprojektowanie łatki, która wzmocni kod i wyeliminuje luki

3. Zastosowanie i testowanie łatki

4. Potwierdzenie, że nie ma żadnych skutków ubocznych

5. Opracowanie dokumentacji łaty

6. Przygotowanie planu dystrybucji łatek do wszystkich klientów, których dotyczy problem

7. Instalacja poprawki

8. Potwierdzenie skuteczności łatki; kampania public relations w celu zrównoważenia wcześniejszej negatywnej reklamy

O ile źródło luki nie zostanie jednoznacznie zidentyfikowane, każda łatka może obejmować problem bez eliminacji źródła luki.

Najczęstsze luki w zabezpieczeniach spowodowane niepewnym kodem to

• Przepełnienie bufora. Działania w programie, zwłaszcza dane wejściowe, przydzielają pewną ilość skończonej przestrzeni na nowe, utworzone dane do przechowywania. Nadmierna ilość danych zapełnia bufor, a program ulegnie awarii, chyba że program zastosuje środki zapobiegawcze.

• Przepełnienie arytmetyczne. Zwykle ma to miejsce, gdy następuje akumulacja i przekroczona zostaje maksymalna liczba akumulatorów. Następna dodawana ilość przewraca się nad akumulatorem, tworząc fałszywy wynik, chyba że ponownie zostaną zastosowane środki zapobiegawcze.

• Atak sformatowanym łańcuchem. Niepewny kod pozwala, aby dane dostarczone przez użytkownika były traktowane jako polecenie, gdy mogą to być tylko dane. Jeśli jest to dozwolone, osoby atakujące mogą zainstalować własny kod wykonywalny.

• Wstrzyknięcie polecenia. Brak weryfikacji danych wejściowych pozwoli na przyjęcie i wykonanie niedozwolonych poleceń.

• Cross-site scripting (XSS). Jest to wstrzyknięcie złośliwego kodu do oprogramowania klienckiego, zwykle w celu ominięcia kontroli dostępu,

• Zastrzyki SQL. To jest kod SQL, który wprowadza w błąd aplikację do wykonania.

• Niezabezpieczone bezpośrednie odniesienie do obiektu. Dotyczy to wyszukiwania stron internetowych. Zamiast nazwy pliku HTML odwołuje się do niego bezpośrednia definicja obiektu, którą strona ma w bazie danych. Ujawnia to tożsamość bazy danych, a osoba atakująca może użyć jej do pobrania innych plików.

• Niebezpieczne przechowywanie. Jest to przechowywanie krytycznych plików bez szyfrowania lub ochrony przed odczytem / zapisem. Do tej kategorii należą słabe przykłady steganografii.

• Słaba kryptografia. Jest to szyfrowanie plików, które można łatwo odczytać.

• Warunki wyścigu. Dzieje się tak, gdy przyznawany jest jeden zasób drugi użytkownik przed zakończeniem pierwszego.

Bezpieczeństwo oprogramowania musi pochodzić ze specyfikacji produktu. Nie jest to krok w rozwoju oprogramowania, ale jest to rozwiązanie, które należy przeplatać z funkcjonalnym projektem i pisaniem kodu. Oznacza to, że sposób myślenia projektanta oprogramowania musi mieć dwie równoległe ścieżki - funkcjonalność i bezpieczeństwo - przy czym bezpieczeństwo jest mechanizmem, który ujawnia, a następnie blokuje włamania. Taki mechanizm bezpieczeństwa powinien być solidny i dobrze zaprojektowany i nigdy nie powinien być zastępowany przez jakiś schemat zaciemnienia danych, który próbuje przechytrzyć atakującego.

Oprócz myślenia o potencjalnym napastniku, projektant musi również pomyśleć o wygodzie użytkownika i zapewnić mechanizm bezpieczeństwa, który nie przeszkadza w normalnych operacjach, będąc dla użytkownika jak najbardziej przejrzysty.

Błędy mogą być dobre

Bez względu na to, błędy są zawsze postrzegane negatywnie i są identyfikowane jako chwile niepowodzenia. To jest złe podejście. Kiedy Thomasowi Edisonowi powiedziano, że sto razy zawiódł przed udoskonaleniem lampy żarowej, odpowiedział, że skoro nauczył się czegoś nowego z każdego ze stu razy, proces jego sukcesu po prostu zrobił sto kroków. Oczywiście, popełnianie tego samego błędu raz po raz nie jest oznaką mądrości. Chodzi o to, aby nie obawiać się popełniania błędów, o ile popełniane są one w dobrej wierze, nie prowadzą do katastrof i, miejmy nadzieję, dodają trochę nowej wiedzy. Błędem byłoby rozszerzenie aplikacji lub użytkownika o więcej uprawnień bezpieczeństwa niż jest to konieczne do wykonania przypisanego zadania. Podczas instalowania aplikacji należy wymienić wszystkie potrzebne zasoby z wymaganymi uprawnieniami, takimi jak odczyt, zapis, tworzenie, usuwanie. W ten sposób, jeśli złośliwe oprogramowanie zainstaluje się w aplikacji, potencjalne szkody będą minimalne, a złośliwe oprogramowanie nie będzie mogło wędrować wewnątrz systemu. Podobnie, gdy komputer jest pojedynczym użytkownikiem, niekoniecznie powinien działać w trybie administratora, ale w trybie użytkownika. W ten sposób, jeśli złośliwe oprogramowanie przejdzie w trybie użytkownika, nie rozprzestrzeni się do obszarów uprzywilejowanych w trybie administratora. Dzięki posiadaniu uprzywilejowanych warstw - użytkownika, superużytkownika, administratora, superadministratora - złośliwe oprogramowanie jest w pewnym stopniu zawarte.

Klasyfikacja zagrożeń

Zasadniczo istnieją trzy rodzaje zagrożeń, mianowicie nielegalna zmiana danych, nielegalny dostęp do danych i nielegalna blokada dostępu do danych

Badacze bezpieczeństwa informacji w firmie Microsoft ogólnie klasyfikowali wszystkie zagrożenia w sześciu kategoriach, których akronim wygodnie składa się na słowo STRIDE (fałszowanie, modyfikowanie, odrzucanie, ujawnianie informacji, odmowa usługi, podniesienie uprawnień).

Proces modelowania zagrożeń

W systemach informatycznych zagrożenia - potencjalne ataki, które mogą wykorzystać luki w systemie - pochodzą z różnych źródeł i mają różny poziom grawitacji. Istnieje potrzeba ustrukturyzowanego podejścia do definiowania zagrożeń i radzenia sobie z nimi. Poniżej znajduje się sekwencja kroków, które mogą służyć jako mechanizm do tego celu.

1. Zdefiniuj, co stanowi przedmiotowy system informacyjny. To znaczy określ granice funkcjonalne i geograficzne tego, co nazywamy naszym systemem. Poza jakim punktem nie jest to nasza odpowiedzialność?

2. Teraz, gdy skwantyfikowaliśmy nasz system, próbujemy go zidentyfikować co uważamy za zagrożenia oparte na wewnętrznych lub zewnętrznych podatnościach. Wymienione wcześniej kategorie STRIDE mogą służyć jako punkt wyjścia do klasyfikacji zagrożeń.

3. Rozpoznaj, które z zagrożeń, w kontekście operacji, stanowią absolutne niebezpieczeństwo, które może prowadzić do katastrofalnego wpływu na system. Zdefiniuj tryby, w których takie zagrożenia mogą stać się skutecznymi atakami.

4. Opracuj opcje obrony dla każdego rozpoznanego zagrożenia i uszereguj rozważane mechanizmy obrony w oparciu o efektywność i zapotrzebowanie na zasoby.

5. Wybierz optymalne podejścia do eliminacji zagrożeń, skuteczności równoważenia, prawdopodobieństwa wystąpienia, ciężkości wystąpienia i kosztów opracowania rozwiązania.

Jest to iteracyjny proces wymagający udziału wszystkich, którzy są związani z bezpieczną wydajnością systemu, a mianowicie analityków, projektantów, programistów, użytkowników, trenerów, sprzedaży i oceniających. Proces ten należy poddawać przeglądowi w określonych odstępach czasu lub gdy pojawią się nowe zagrożenia.

Bezpieczeństwo zaczyna się w domu

Oczywiście dom to kreatywne fazy cyklu rozwoju oprogramowania (SDLC), zwłaszcza etapy projektowania i kodowania programów. SDLC jest często opisywane jako składające się z siedmiu podstawowych etapów przejściowych. Mianowicie,

1. Od abstrakcyjnych potrzeb do wymagań formalnych - analiza

2. Od wymagań formalnych po ogólny projekt - projektowanie

3. Od ogólnego projektu do kodu programu

4. Od wszystkich powyższych do opracowania dokumentacji

5. Od wszystkich powyższych do testowania systemu

6. Od testowania systemu do wykorzystania systemu (lub do poprzedniego kroku)

7. Od wykorzystania systemu do ulepszeń (krok 1)

W fazie analizy definiowane są wymagania bezpieczeństwa, w fazie projektowania rozwijane są mechanizmy bezpieczeństwa. Później podczas testów luki zostaną odkryte i wyeliminowane. Eliminacja podatności zwykle przyjmuje jedną z dwóch form. Jednym z nich jest usunięcie przyczyn luki, a drugim usunięcie skutków luki. W zależności od panujących uzasadnień zwykle wdrażane jest jedno z dwóch rozwiązań. Jest to obecnie standardowa praktyka i wymóg prawny na wszystkich głównych budowach, aby inżynierowie bezpieczeństwa nadzorowali wszystkie działania na rzecz bezpieczeństwa

pracowników. Podobnie podczas opracowywania systemów informatycznych specjaliści ds. Bezpieczeństwa oprogramowania muszą nadzorować proces rozwoju od a do z i wykonywać następujące czynności:

• Zidentyfikuj typowe błędy programowania, które prowadzą do podatności.

• Ustanowić standardowe praktyki bezpiecznego kodowania.

• Edukuj programistów (/ projektantów).

Ponadto mogą obejmować inicjatywy rozwoju oprogramowania wewnątrz organizacji

• Certyfikacja programistów w zakresie bezpiecznego kodowania

• Certyfikacja oprogramowania w zakresie bezpiecznego kodowania

• Wykorzystanie narzędzi do analizy oprogramowania

Narzędzia programowe są bardzo ważne w rozwoju bezpiecznych i dobrze udokumentowany kod i można go podzielić na następujące cztery kategorie:

• Pokrycie kodu - Śledzi kod i lokalizacje danych, które zostały utworzone, odczytane lub zmodyfikowane.

• Śledzenie instrukcji - rejestruje wyniki każdej instrukcji, udostępniając ją do późniejszej analizy krok po kroku.

• Analiza pamięci - śledzi wykorzystanie miejsca w pamięci, szukając możliwych naruszeń.

• Analiza wydajności - w oparciu o kryteria użytkownika oprogramowanie jest analizowane i dostrajane w celu optymalizacji wydajności.

W ten sposób aplikacje systemu informatycznego będą zarówno bezpieczne, jak i niezawodne. Software Engineering Institute na Carnegie Mellon University jest liderem w opracowywaniu standardów, narzędzi i szkoleń mających na celu projektowanie niezawodnych i bezpiecznych systemów informatycznych. Głównym problemem związanym z tworzeniem oprogramowania jest ekonomia cyklu życia i logistyka, w których bardzo często inwestycja przedpremierowa nie pokrywa się z przychodami po wydaniu. Często czas od wydania pomysłu do produktu jest minimalizowany, co często skutkuje wydaniem niezabezpieczonego oprogramowania. Na ratunek szybkiemu rozwojowi oprogramowania przybyły języki zorientowane obiektowo i ustrukturyzowane koncepcje programowania leżące u podstaw projektowania nowoczesnych systemów informatycznych. Jednak bezpieczne oprogramowanie od samego początku nadal stanowi nową zasadę głoszącą, że bezpieczeństwo musi stanowić integralną część funkcjonalności

Bezpieczeństwo w aplikacjach

Luki w zabezpieczeniach są również dostarczane z niezabezpieczonymi aplikacjami. Microsoft Word jest typowym przykładem. Jego wadą jest to, że po zażądaniu zapisu poprzednio usunięty tekst, choć już nie pojawia się w końcowym dokumencie, nadal pozostaje częścią pliku. Po otwarciu pliku Word w Notatniku usunięty tekst jest wyraźnie widoczny podczas przeglądania .

Dokument Word nie jest częścią końcowego widocznego tekstu. Ta luka narusza prywatność użytkownika i sugeruje poufność, nie oferując żadnej dodatkowej funkcjonalności. W takim przypadku jedynym rozwiązaniem jest wybranie wszystkich, skopiowanie i wklejenie w zupełnie nowym dokumencie pod koniec przygotowywania dokumentu. Jednak przeciętni użytkownicy nie

podejrzewają istnienia luk w zabezpieczeniach, szczególnie w przypadku produktów znanych marek, ani nie dysponują umiejętnościami technicznymi do wykrywania i usuwania wad projektowych oprogramowania. Obowiązkiem pracowników ds. bezpieczeństwa informacji w organizacji jest posiadanie wiedzy o istnieniu wad w oprogramowaniu, które zatwierdzają do użytku, oraz odpowiednie doradzanie użytkownikom. Departament Bezpieczeństwa Wewnętrznego Stanów Zjednoczonych (US-DHS) utrzymuje obszerny zespół gotowości na wypadek awarii komputera (US-CERT), który ostrzega opinię publiczną o istnieniu luk w oprogramowaniu, które jest powszechnie używane i które prowadzi online bazę danych notatek o zagrożeniach. Luka w zabezpieczeniach.

Uwaga, na przykład VU # 446012, identyfikuje i opisuje „podatność na uszkodzenie pamięci wskaźnika Microsoft Word". Notatka wskazuje na własny biuletyn zabezpieczeń firmy Microsoft, który stwierdza: „Gdy użytkownik otworzy specjalnie spreparowany dokument Word za pomocą zniekształconego wskaźnika obiektu, może on uszkodzić pamięć systemową w taki sposób, że osoba atakująca może wykonać dowolny kod [złośliwego oprogramowania]". Notatka kontynuuje, że „Dokumenty Office mogą zawierać obiekty osadzone" (które mogą zawierać złośliwe oprogramowanie). Dlatego zaleca się, aby przed użyciem aplikacji odwiedzić tę bazę danych i dowiedzieć się o obecności interesujących luk w zabezpieczeniach. Sortując wyżej wspomnianą bazę danych według poziomu istotności, można rozpoznać, że na górze pojawiają się luki spowodowane przepełnieniem bufora. W związku z tym konieczne jest, aby bufory miały dynamiczny rozmiar, a także samooczyszczanie przestarzałych danych. Najczęstszą bramą przed atakami złośliwego oprogramowania jest przeglądarka internetowa. Z jednej strony musi być otwarty i wydajny, aby ułatwić wyświetlanie danych; z drugiej strony musi być skonfigurowany do ochrony komputera hosta. Eksperci ds. Bezpieczeństwa uważają, że „Wiele przeglądarek internetowych jest skonfigurowanych w celu zapewnienia zwiększonej funkcjonalności kosztem obniżenia bezpieczeństwo." Doskonały przewodnik po zabezpieczaniu przeglądarki internetowej jest dostępny online, z którego mogą korzystać zarówno nowicjusze, jak i eksperci.

Przedstawiamy środki zaradcze

Przeciwdziałanie cyberbezpieczeństwu to działanie, którego celem jest zablokowanie lub zminimalizowanie wpływu ataku cybernetycznego na nasz nasz zasób podłączony do sieci. Takim działaniem może być wykorzystanie określonego procesu, a określonego urządzenia, określonej technologii lub określonego systemu. Jako składnik aktywów uwzględniamy każdą naszą jednostkę podłączoną do Internetu, taką jak komputer, smartfon, urządzenie lub serwer. Podczas gdy podatności nie można całkowicie wyeliminować, można zastosować środki zaradcze, które zminimalizują znaczenie zagrożenia. Obrona przed zagrożeniami zewnętrznymi może przybierać różne formy, takie jak:

• Sprawdzanie adresów URL prób dostępu do systemu online pod kątem listy wstępnie zatwierdzonych.

• Minimalizowanie ryzyka włamania poprzez zmniejszenie uprawnień dostępu wszędzie tam, gdzie nie są absolutnie konieczne. Lub przedłuż je i wycofaj, gdy zajdzie taka potrzeba.

• Minimalizowanie dostępności wrażliwych danych w Internecie, zmniejszając w ten sposób narażenie na ewentualne włamanie.

• Posiadanie haseł odpornych na ataki „słownikowe" poprzez dołączanie liter języków obcych.

• Korzystanie z uwierzytelniania wieloskładnikowego, na przykład otrzymywanie dodatkowych danych parametry dostępu przez telefonię komórkową.

• Projektowanie pod kątem dużego natężenia ruchu, tak aby próby „zalania" portów nie powiodły się.

• Rozwijanie umiejętności rozpoznawania rozproszonego ataku typu „odmowa usługi”, DDoS, ataków za pomocą dynamicznych wskaźników, które stale obserwują wykorzystanie zasobów.

• Korzystanie z zapory ogniowej, oprogramowania antywirusowego oraz aktualizacja oprogramowania (aktualizacji i łatek) urządzeń podłączonych do Internetu.

• Rejestruj i zgłaszaj próby włamań i podejrzane żądania sieciowe.

• Odłącz lub dezaktywuj zasoby dostępne w Internecie, które nie są w tej chwili używane.

• Przeprowadzaj częste (codzienne lub cotygodniowe) audyty. Bądź świadomy rodzaju aplikacji znajdujących się w twoim systemie.

Międzynarodowa świadomość

Podobnie jak US-CERT, inne kraje zajmujące się bezpieczeństwem cyberprzestrzeni utworzyły podobne agencje rządowe w tym celu.

ĆWICZENIA

1. Otwórz program MS Word i utwórz plik zawierający tylko słowa „Dzień dobry”. Usuń wszystko i napisz „Dobranoc”. Zapisz plik pod nazwą Good.doc. Otwórz plik za pomocą Notatnika. Zobacz dolną część kodu. Zobaczysz usunięte słowa „Dzień dobry”. Jakie wyciągasz wnioski?

2. W dwustronicowym raporcie wyjaśnij podatność na przepełnienie bufora. Przywołaj pięć odniesień - komunikaty prasowe - w których zidentyfikowano to jako przyczynę odmowy usługi.

3. Zidentyfikuj preferowane działanie domowych systemów alarmowych i opracuj wymagania formalne (ilościowe).

4. Wybierz dwie przeglądarki internetowe i wykonaj badanie porównawcze ich funkcji bezpieczeństwa.

5. Zbadaj użycie terminów administrator i super administrator.

6. Zbadaj użycie terminów użytkownik i super użytkownik.