Luki w zabezpieczeniach sieciowych

Dokonamy systematycznego przeglądu podstawowych komponentów oprogramowania tworzących aplikacje webowe, ze szczególnym naciskiem na handel elektroniczny, i zawiera przegląd zagrożeń dla każdego z tych komponentów. Celem tego rozdziału jest wskazanie, że każdy system będzie wiązał się z zagrożeniami dla bezpieczeństwa i prywatności, które należy systematycznie analizować i ostatecznie rozwiązać

ŁAMANIE SYSTEMÓW E-HANDLU.

Aby system był bardziej bezpieczny, może być wskazane jego uszkodzenie. Znalezienie słabych punktów w systemie jest konieczne w celu jego wzmocnienia, ale złamanie systemu e-commerce wymaga innego sposobu myślenia niż programistów, którzy go opracowali. Zamiast myśleć o opracowaniu w ramach specyfikacji, przestępca lub haker spogląda poza specyfikacje. Hakerzy uważają, że reguły istnieją tylko po to, aby je złamać, i zawsze używają systemu w nieoczekiwany sposób. Robiąc to, zwykle podążają ścieżką najmniejszego oporu. Obszary uważane za zapewniające najwyższe bezpieczeństwo lub najbardziej odporne na ataki hakerskie zostaną prawdopodobnie zignorowane. Na przykład, jeśli system używa Secure Sockets Layer (SSL) do szyfrowania sesji Web między klientami Web a serwerem Web, haker nie będzie próbował przerwać strumienia szyfrowania, ale zamiast tego poszuka łatwiejszego sposobu na uzyskanie danych po ich odszyfrowaniu i przechowywane w czystym miejscu. Hakerzy idą tam, gdzie są pieniądze - czasem dosłownie, a czasem nie. Zazwyczaj próbują włamać się na stronę tylko wtedy, gdy za ich wysiłek jest jakaś nagroda. Czasami hakerzy motywowani są pieniędzmi, ale równie często motywacją jest przynęta sławy, rozgłosu lub akceptacji ze strony grupy rówieśniczej. Poziom ochrony powinien być współmierny do wartości chronionego zasobu. Na przykład witryna internetowa, która publikuje menu lokalnych restauracji, nie może być postrzegana jako cel odmowy usługi lub jakiejkolwiek inny atak innego typu. Taka strona internetowa po prostu nie jest tak atrakcyjnym celem jak witryna internetowa banku, gdzie haker może z pewnością zyskać rozgłos, a nawet zyski finansowe. Podobnie większość ludzi nie zawraca sobie głowy szyfrowaniem wiadomości e-mail z powodu braku świadomości bezpieczeństwa i ponieważ większość potencjalnych szpiegów nie jest zainteresowana zwykłą osobistą wiadomością e-mail. Wrażliwe wiadomości e-mail od znanej organizacji powinny być jednak szyfrowane, aby chronić cenny kapitał intelektualny. Należy pamiętać, że system handlu elektronicznego jest łańcuchem tak silnym, jak jego najsłabsze ogniwo. Hakerzy naturalnie próbują zaatakować ten punkt najmniejszego oporu. To wyjaśnia, dlaczego strona może celowo skonfigurować honeypot (system ofiarny) z atrakcyjnie wrażliwymi usługami w celu śledzenia i monitorowania potencjalnych hakerów. W handlu elektronicznym lub dowolnym systemie z obecnością w sieci kryptografia jest często postrzegana jako zapewniająca najwyższy poziom ochrony; dlatego hakerzy zazwyczaj atakują usługi po stronie hosta i treści po stronie klienta. niezwykle ważne jest zapewnienie silnego bezpieczeństwa hosta, zarówno wewnątrz, jak i na zewnątrz obwodu. Jednym niefortunnym skutkiem ubocznym granic korporacyjnych jest to, że administratorzy systemu często przeoczają bezpieczeństwo oparte na hoście, co może całkowicie narazić hosta. W rezultacie hakerzy, którzy przechodzą przez urządzenia i zasady obwodowe (np. Zapory ogniowe, routery, a nawet recepcjonistki), mogą wykorzystać wewnętrzne relacje zaufania, aby narazić wiele zasobów, w tym stacje robocze, serwery i systemy telefoniczne. Ostrożny administrator będzie traktował jednakowo zarówno przy wejściu do systemów, jak i w tych systemach.

STUDIUM PRZYPADKU ZŁAMANIA E-BIZNESU.

Zastanów się nad aplikacją e-biznesową do inwestowania online i tym, jak haker może zająć się demontażem bezpieczeństwa w celu uzyskania złośliwych lub finansowych korzyści. Inwestowanie online jest bardzo popularne z kilku powodów. Zamiast czekać na kwartalne wyciągi pocztowe lub

zajmować się menu telefonu, klienci mogą szybko sprawdzić status, saldo i bieżącą wartość inwestycji, odwiedzając strony internetowe swoich menedżerów portfela. Jeśli chcą kupować i sprzedawać akcje na żądanie, mogą założyć internetowe konta maklerskie. Aby zobaczyć, w jaki sposób można złamać tę aplikację, warto przyjrzeć się przykładowej architekturze sieci, która implementuje aplikację online. System składa się z komputerów klienckich użytkowników końcowych, Internetu, routerów, zapory ogniowej, front-endowych serwerów WWW i poczty e-mail, serwerów aplikacji, baz danych i stacji orkst. Istnieje wiele sposobów na złamanie tej aplikacji online przez hakera. Pokażemy jeden możliwy scenariusz. W kroku 1 ataku haker korzysta z Internetu i przeglądarki internetowej do niewłaściwego użycia jednego ze skryptów Common Gateway Interface (CGI), które wywołują aplikację na serwerze. Skrypt CGI może być programistycznym skryptem CGI przypadkowo pozostawionym na serwerze przed uruchomieniem, domyślnym skryptem CGI z dystrybucji serwera aplikacji lub skryptem, który implementuje wadliwą logikę w aplikacji inwestycyjnej online. Wykorzystanie skryptów CGI poprzez skrypty między witrynami lub inne popularne metody wykorzystujące lukę pozwalają hakerom uzyskać dostęp do powłoki powłoki na serwerach WWW. Luka nie musi znajdować się w skrypcie CGI. Serwery aplikacji mogą być implementowane w Javie, C, C ++, Perlu lub Pythonie w różnych ramach serwera aplikacji. Trudność nie polega na tym, w jakim języku opracowywana jest logika aplikacji biznesowych; ważniejsze są luki wprowadzone przez złożoną logikę w tej warstwie oprogramowania pośredniego. Jednym z kluczowych problemów w rozwoju oprogramowania pośredniego aplikacji jest słabe sprawdzanie poprawności danych wejściowych; to znaczy, programiści nie narzucają limitów na akceptowalny wkład. Haker może wykorzystać brak sprawdzania poprawności danych wejściowych, aby podać serwerowi aplikacji nieoczekiwane dane wejściowe użyte w poleceniach systemowych. Ta technika może uzyskać uprawnienia do powłoki na komputerze. Chociaż niewłaściwe użycie serwera aplikacji jest częstym sposobem włamania się do systemów, istnieje wiele innych sposobów uzyskania początkowego dostępu w kroku 1 ataku. Na przykład na serwerach internetowych i pocztowych może działać dowolna z kilku usług sieciowych, takich jak FTP i BIND / DNS, które mogą być źle skonfigurowane, a tym samym „zatrute". Oprogramowanie WWW i same serwery poczty mogą być narażone na atak. Wiele popularnych komercyjnych serwerów WWW i poczty elektronicznej jest narażonych na ataki związane z przepełnieniem bufora, które często pozwalają na pełne uprawnienia root systemu na hoście. Po tym, jak atakujący uzyskają uprawnienia systemowe na hoście wewnętrznym, mogą wykorzystać wewnętrzne zaufanie, często utkane między komputerami za pomocą zasad sieciowych, w celu uzyskania dostępu do innych systemów w sieci. Strategia ta jest dokładnie tym, co atakujący stosuje w kroku 2 ataku zilustrowanym na Rys. 21.2. Gdy atakujący uzyskają dostęp do różnych systemów plików na serwerze aplikacji, mogą przeglądać kod źródłowy skryptów CGI lub innego oprogramowania pośredniego w celu wykrycia numerów kont klientów, hasła, a nawet hasła administratora bazy danych umożliwiające dostęp do baz danych zaplecza. Stamtąd mogą pobierać ważne i poufne informacje o kliencie przechowywane w bazie danych. W kroku 3 atakujący wykorzystuje niezauważone wewnętrzne przywileje do sadzenia backdoorów w systemie. Pakiet oprogramowania, powszechnie znany jako rootkit i dostępny dla hakerów, pozwala im nie tylko niezauważalnie dostać się do systemu, ale także usunąć ślady w dziennikach kontroli. W pokazanym przykładzie haker instaluje nieuczciwy program do zdalnej administracji znany jako Back Orifice, który zapewnia możliwość zdalnego administrowania hostem i siecią z tymi samymi uprawnieniami i mocą jak autentyczny administrator systemu.

W tym momencie ataku haker przejął całkowitą kontrolę nad systemami i wszelkimi procesami związanymi z Internetem lub procesami handlowymi, z wieloma opcjami, w tym:

* Kradzież informacji o kliencie lub firmie w celu uzyskania korzyści finansowych

* Defacing stron internetowych dla rozgłosu lub opublikowania agendy

* Pracując w ukryciu, aby odkryć zastrzeżone informacje biznesowe i inny poufny kapitał intelektualny (szpiegostwo)

* Szantażowanie firmy grożąc jej dyskredytacją

* Zniszczenie aplikacji w celu uzyskania jakichkolwiek innych korzyści osobistych

Krok 4 ataku ilustruje ostatni przypadek, w którym atakujący zasila osobiste konto gotówkowe. Haker musi działać wystarczająco szybko, aby wycofać te środki, zanim tradycyjne kontrole wewnętrzne wykryją rozbieżność. Było wiele nieczytelnych stron internetowych agencji rządowych i innych ważnych witryn oraz wiele zgłoszonych przypadków innych przypadków. Oczywiście niektóre firmy niechętnie publikują informacje o zdarzeniach, które mogą obniżyć zaufanie klientów, jednak w wielu przypadkach przepisy wymagają ujawnienia naruszenia. Niestety haker musi skompromitować system jako całość tylko jeden wadliwy, niepoprawiony komputer lub przeoczona luka. Chociaż niezależność obrony (stosowanie wielu form bezpieczeństwa, takich jak zapory ogniowe na obwodzie i wykrywanie włamań w sieci) jest popularną strategią, często wiele warstw obrony spada jak domek z kart, gdy wykorzystuje się jeden dołek. Na przykład atakujący, który zyskuje możliwość rootowania, może wyłączyć wszystkie inne środki bezpieczeństwa. Problem znany jest jako atak asymetryczny, ponieważ obrona przed takim atakiem jest znacznie trudniejsza i bardziej kosztowna niż jego uruchomienie. Chociaż zaangażowany haker może być w stanie poświęcić tyle czasu, ile potrzeba na włamanie się do systemu, firma nie może wydać tyle pieniędzy, ile potrzeba, aby obronić się przed wszystkimi możliwościami. Liczba błędów, które można wykorzystać, jest oszałamiająca, biorąc pod uwagę wszystkie różne platformy i urządzenia, które tworzą obecną infrastrukturę technologii informatycznych (IT). Problemem jest również fakt, że haker potrafi względnie anonimowo korzystać z komputera i modemu za 500 USD do przeprowadzania ataków. Co gorsza, hakerzy mogą pracować z dowolnej liczby kiosków internetowych dostępnych na lotniskach, w centrach handlowych, kawiarniach, a nawet w pralniach. W miarę, jak hakerzy stają się coraz bardziej wyrafinowani i gdy stają się dostępne łatwiejsze w użyciu skrypty, ataki będą uruchamiane z urządzeń mobilnych, które mogą wędrować do różnych stref geograficznych i z nich wychodzić, a następnie zostać odrzucone – czyniąc śledzenie atakującego niemożliwym.

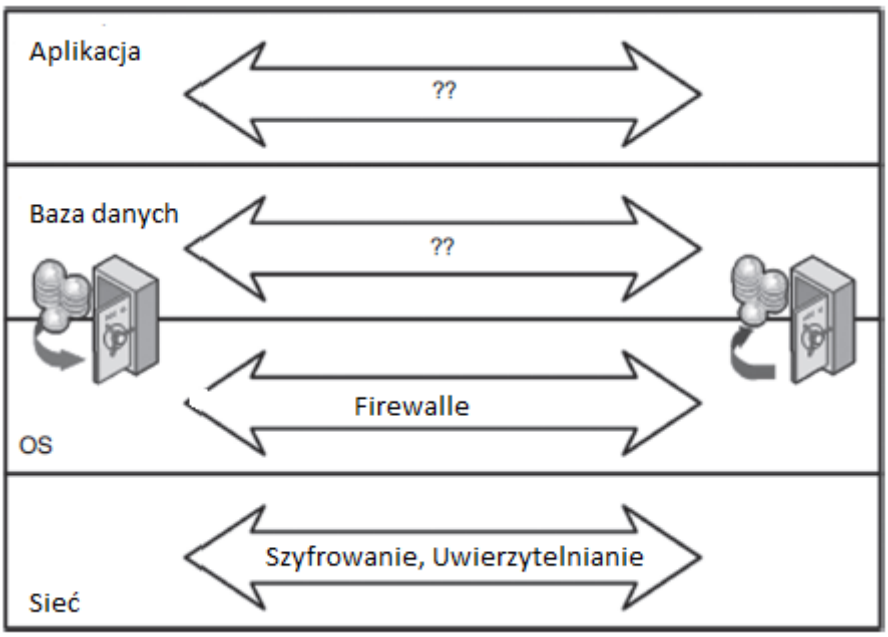
BEZPIECZEŃSTWO SYSTEMU APLIKACJI WEBOWEJ.

Pomimo dość ponurego obrazu namalowanego tutaj organizacje mogą skutecznie zarządzać ryzykiem hakerów. Podobnie jak w wielu innych domenach bezpieczeństwa, postawa bezpieczeństwa lub postawa przyjęta przez organizację ma kluczowe znaczenie dla odstraszania i udaremniania hakerów. Aby zastosować analogię do świata fizycznego, rozważ włamywaczy, którzy zamierzają włamać się do domów w ładnej okolicy. Gdy włamywacze wykrywają potencjalne cele, zauważą niektóre domy z alarmami przeciwwłamaniowymi - wraz z widocznymi znakami systemów alarmowych - a niektóre bez nich. Najprawdopodobniej włamywacze ominą domy z alarmami antywłamaniowymi i ruszą na inne, słabiej chronione cele. Tak więc pozycja bezpieczeństwa przyjęta przez właściciela odgrywa ważną rolę. Każda organizacja musi najpierw ustalić pożądany poziom bezpieczeństwa, zgodnie z dokumentacją w polityce bezpieczeństwa. Administratorzy systemu używają zasad bezpieczeństwa do konfigurowania systemów, routerów, zapór i rozwiązań dostępu zdalnego. Bez wyraźnej polityki bezpieczeństwa nie ma sposobu, aby określić stopień bezpieczeństwa organizacji, jak skonfigurować jej systemy, a nawet jeśli wystąpiło nieoczywiste naruszenie bezpieczeństwa. Po opracowaniu polityki bezpieczeństwa należy ocenić faktyczne wdrożenie zabezpieczeń. Oznacza to, że system musi zostać przetestowany i oceniony, aby określić, jak dobrze spełnia swoją politykę bezpieczeństwa. Zwykle istnieje różnica między pożądaną postawą a rzeczywistą postawą. Różnica polega na luce w bezpieczeństwie między miejscem, w którym organizacja chciałaby być (zabezpieczona przed określonymi zagrożeniami), a tym,

gdzie faktycznie jest. Proces opracowywania polityki bezpieczeństwa i oceny systemów organizacji pod kątem tej polityki pozwoli zidentyfikować nie tylko luki między rzeczywistym stanem bezpieczeństwa a pożądaną postawą, ale także słabości samej polityki bezpieczeństwa. Ważne jest, aby niezależna partia, najlepiej zewnętrzna, oceniała stopień bezpieczeństwa organizacji. Strona trzecia może rzetelnie ocenić, czy system organizacji przestrzega polityki bezpieczeństwa. Jeśli grupa, która opracowuje politykę bezpieczeństwa lub konfigurację systemu, jest również odpowiedzialna za jej ocenę, ocena może być stronnicza, a potencjalne luki w zabezpieczeniach mogą zostać przeoczone. Może się to zdarzyć nie z powodu złej woli lub nieuczciwości, ale raczej z powodu trudności w obiektywnym podejściu do własnej pracy.

OCHRONA APLIKACJI INTERNETOWYCH.

Bezpieczeństwo aplikacji internetowych można rozumieć z różnych widoków. Po pierwsze, rozważ widok e-biznesu.



Schemat pokazuje dwa e-biznesy komunikujące się przez Internet, być może przeprowadzające transakcje między przedsiębiorstwami. W tym widoku najniższą warstwą e-biznesu jest warstwa sieciowa. W warstwie sieciowej istnieją obawy dotyczące niezawodności, integralności i poufności danych przesyłanych przez kanał komunikacyjny. Ta warstwa jest szczególnie ważna, ponieważ Internet jest publiczną siecią komutowaną, co oznacza, że dowolna liczba stron trzecich może mieć dostęp do danych, które przemierzają węzły Internetu w drodze od źródła danych do miejsca docelowego. Ponadto protokół internetowy (IP) jest protokołem bezpołączeniowym, co oznacza, że nie ma dedykowanego obwodu między źródłem a miejscem docelowym. W rezultacie pakiety wysyłane podczas danej sesji mogą podążać różnymi trasami do miejsca docelowego, w zależności od natężenia ruchu i algorytmów routingu. Ponieważ IP jest zawodnym protokołem datagramowym (tzn. IP używa niezależnych pakietów przekazywanych od węzła do węzła, ale nie ma gwarancji pomyślnej transmisji), warstwa sieciowa zawiera niezawodną warstwę transmisji zorientowaną na połączenie, taką jak protokół kontroli transmisji (TCP), który zapewnia, że upuszczone lub utracone pakiety są retransmitowane i że bit może być odwrócony podczas transmisji (np. przez sieci bezprzewodowe) są korygowane. Chociaż protokół TCP / IP zapewnia bardziej niezawodne dostarczanie pakietów internetowych, protokół IP w wersji 4 nie zapewnia bezpiecznych usług połączeń. Zazwyczaj oznacza to, że nie ma gwarancji poufności, identyfikacji, a nawet dostarczenia pakietów wysyłanych z jednego

hosta internetowego do drugiego. Ponieważ pakiety często przechodzą przez kilka węzłów internetowych od źródła do miejsca docelowego, treści pakietów mogą być przechwytywane przez strony trzecie, kopiowane, zastępowane, a nawet niszczone. Jest to ryzyko, że większość ludzi powołuje się na ryzyko związane z handlem elektronicznym to potępi; przeoczyli bardziej istotne ryzyko związane z handlem elektronicznym związanym z bezpieczeństwem i prywatnością po stronie serwera i klienta. IPv6 może przynieść wiele ulepszeń bezpieczeństwa w warstwie sieci, ale nie został szeroko wdrożony w Stanach Zjednoczonych. Na szczęście, nawet w IPv4, mamy dobre rozwiązania problemu poufności danych. Techniki kryptograficzne mogą zapewnić silne gwarancje poufności danych, uwierzytelnienia stron oraz integralności danych przesyłanych podczas transmisji. Ponadto podpisy cyfrowe mogą być używane do podpisywania otrzymanej poczty w aplikacji „potwierdzenie odbioru", która zapewnia gwarancje dostarczenia wiadomości e-mail. Tak więc, jak pokazano na powyższym rysunku, możemy korzystać z usług szyfrowania w celu ochrony danych przesyłanych przez sieć. System operacyjny (OS) lub platforma hostująca aplikacje Web działa w warstwie sieciowej. W modelu warstwowym usługi jednej warstwy korzystają z usług dolnej warstwy i zapewniają usługi dla wyższych warstw. Warstwa sieci jest często uważana za rdzeń systemu operacyjnego; jednak z punktu widzenia usług warstwowych oprogramowanie systemu operacyjnego działa na warstwie sieci. Systemy operacyjne są powszechnie znane z wad oprogramowania, które wpływają na bezpieczeństwo systemu. Systemy operacyjne są wrażliwe, ponieważ komercyjne systemy operacyjne są obecnie niezwykle złożone; na przykład system operacyjny Windows Vista prawdopodobnie zawiera ponad 50 milionów linii kodu źródłowego. Nie można wyłapać wszystkich błędów projektowych i programistycznych, które mogą mieć konsekwencje bezpieczeństwa na tak złożonej platformie. Mimo że systemy operacyjne UNIX są używane przez większą część 30 lat, nowe wady narzędzi systemu operacyjnego są cotygodniowe we wszystkich różnych wariantach platform UNIX. Luki w zabezpieczeniach platformy są z natury krytyczne. Oznacza to, że jeśli sam system operacyjny jest podatny na wykorzystanie, dziury w platformie mogą zagrozić bezpieczeństwu aplikacji. System operacyjny jest zawsze podstawą, na której budowane są aplikacje, więc pęknięcia w fundamencie powodują słabe bezpieczeństwo w warstwie aplikacji. Jak pokazuje rysunek, zapory ogniowe zapewniają ochronę przed niektórymi wadami systemu operacyjnego. Jedną z kluczowych ról zapór ogniowych jest ich zdolność do zamykania usług oferowanych dla logicznych adresów domen. Za pomocą adresów domen internetowych administrator zapory może podzielić adresy internetowe na zaufane i niezaufane zakresy domen. Na przykład dowolny adres internetowy spoza domeny firmy można uznać za niezaufany. W rezultacie wszystkie usługi systemu operacyjnego, takie jak zdalne logowanie, mogą zostać zamknięte dla wszystkich osób spoza domeny firmy. Domeny mogą być podzielone na partycje, tak aby niektóre poddomeny były zaufane w zakresie dostępu do niektórych komputerów, a inne nie.

Kluczową zaletą zapór ogniowych jest zatem możliwość ograniczenia dostępu do platformy za pośrednictwem oferowanych usług (tj. Określonych funkcji, które przekazują dane przez zaporę). W rezultacie zapory ogniowe mogą łatwo ukryć wady systemu operacyjnego przed niezaufanymi podmiotami. Mimo to zapory są podatne na ataki oparte na danych lub kodzie za pośrednictwem oferowanych usług. Są to czasami luki w zabezpieczeniach „dozwolona ścieżka". Na przykład atak za pośrednictwem SMTP (poczta) lub HTTP (sieć) nie zostanie zatrzymany przez zaporę, jeśli zapora jest skonfigurowana do przepuszczania poczty e-mail i usług internetowych, co jest konieczne w przypadku handlu elektronicznego. Zapory ogniowe nie powstrzymają również exploitów systemu operacyjnego od osób wewnętrznych lub zaufanych podmiotów, którym przyznano dostęp do platformy. Należy jednak pamiętać, że odpowiednio skonfigurowane zapory ogniowe mogą ograniczyć narażenie na znaczną liczbę luk w zabezpieczeniach platformy, po prostu odmawiając niezaufanym osobom z zewnątrz dostępu do narzędzi i usług platformy. Rysunek ilustruje rozsądną ochronę przed atakami

sieciowymi i platformowymi, ale nie przed atakami aplikacji i baz danych. Warstwa bazy danych jest pokazana osobno na schemacie ze względu na znaczenie jej roli w e-commerce; jednak ataki na bazę danych zwykle można uznać za rodzaj ataku na aplikację. Ataki oparte na aplikacjach stanowią krytyczny problem, który zyskuje coraz większą uwagę w ostatnich latach. Nie ma prostego rozwiązania, ale rośnie poczucie, że bezpieczeństwo zapewniane przez zaporę ogniową, stronę internetową z obsługą SSL oraz szyfrowanie, wraz z cyfrowymi certyfikatami i podpisami, nie wystarczą. Środki te są konieczne, ale niewystarczające. Aplikacje to przede wszystkim biznes online. Standardy bezpieczeństwa danych branżowych kart płatniczych (PCI DSS) informują każdą organizację przetwarzającą transakcje kartą kredytową, że od czerwca 2008 r. Wymagana jest ochrona przed lukami w zabezpieczeniach aplikacji Web. PCI DSS określa: „sprawdzenie całego niestandardowego kodu aplikacji pod kątem typowej luki w zabezpieczeniach organizacji specjalizującej się w bezpieczeństwie aplikacji ”lub„ instalowanie zapory ogniowej warstwy aplikacji przed aplikacjami internetowymi ”. Aplikacje online są coraz bardziej wyrafinowane, a oprogramowanie wdrażające logikę aplikacji stało się bardzo złożone, wymagając paradygmatów obiektowych i obiektowych, takich jak Enterprise Java Beans, CORBA i usługi DCOM / COM. Łącznie są one znane jako serwery aplikacji. Kluczową kwestią jest jednak to, że ponieważ logika aplikacji jest niestandardowa i złożona, często występuje mnóstwo błędów w implementacji lub logice, które mogą być i często są wykorzystywane przez hakerów. Bezpieczeństwo aplikacji nie może być mylone z oświadczeniami marketingowymi. Bezpieczna aplikacja online jest odporna na ataki. Nie jest to po prostu uwierzytelnianie użytkowników końcowych, szyfrowanie danych transakcji, zapewnianie niezaprzeczalności transakcji i gwarancja usług. Są to wszystkie ważne kwestie dotyczące właściwości transakcji, a nie właściwości oprogramowania. Pozostała część tej części szczegółowo opisuje problem z oprogramowaniem.

ELEMENTY I WRAŻLIWOŚCI W SYSTEMACH E-HANDLU.

Za pomocą Internetu klienci sieci Web (działający na komputerach osobistych lub urządzeniach przenośnych) łączą się z serwerem front-end Web, warstwą oprogramowania pośredniego logiki aplikacji biznesowych, zapleczem bazy danych, system ERP, oprogramowanie do zarządzania łańcuchem dostaw, a nawet niektóre starsze systemy, które są teraz dostępne w Internecie.

Ryzyko po stronie klienta.

Większość handlu elektronicznego odbywa się przy użyciu standardowych przeglądarek internetowych i klientów pocztowych. Coraz częściej handel elektroniczny odbywa się na przenośnych urządzeniach mobilnych, takich jak PDA i telefony komórkowe. Zagrożenia bezpieczeństwa po stronie klienta wynikają głównie ze złośliwego kodu mobilnego, takiego jak skrypty sieciowe, formanty ActiveX i wrogie aplety Java. Innym poważnym ryzykiem w oprogramowaniu po stronie klienta jest utrata prywatności. Każdy komputer wraz z powiązanym oprogramowaniem odbiera i przesyła wiele osobistych informacji identyfikujących (PII). Na przykład przeglądarki mogą przekazywać informacje o komputerach (nazwa, adres IP, typ przeglądarki, wersja, nazwa firmy), a czasem także o samych użytkownikach, szczególnie jeśli włączono funkcje automatycznego wypełniania formularzy. Przeglądarki służą również do śledzenia ruchów w sieci. Na przykład każda witryna odwiedzana przez przeglądarkę zazwyczaj otrzymuje zapis poprzedniej witryny, z której wszedł użytkownik. Reklamy banerowe na stronach internetowych śledzą także, które strony były odwiedzane w celu utworzenia profilu korzystania z sieci, a pliki cookie mogą być umieszczane w systemach użytkowników końcowych w celu śledzenia, a czasami ponownego tworzenia danych wejściowych użytkownika. Klasa bardziej podstępnych programów, zwanych programami szpiegującymi, może wysyłać informacje o użyciu komputera do określonych witryn, często bez wiedzy i zgody użytkownika. Jednym z kluczowych zagrożeń związanych z oprogramowaniem po stronie klienta jest to, że użytkownicy po prostu nie wiedzą, co programy ujawniają o sobie lub komu. Prostą zasadą jest to, że program nie powinien

wysyłać żadnych danych, o których użytkownik nie wie. Audyt programów na dowolnej maszynie prawdopodobnie ujawniłby, że sporo z nich łamie tę zasadę. Chociaż większość programów szpiegujących jest napisana w celu dostarczania danych marketingowych producentom oprogramowania lub przekierowywania użytkownika do wcześniej określonych stron, wielu bardziej skutecznie manipuluje użytkownikami i atakuje ich, szpiegując aktywność użytkowników, a nawet instalując dodatkowy złośliwy kod. Na przykład SpectorSoft sprzedaje narzędzie szpiegujące znane jako eBlaster osobom, które podejrzewają, że ich małżonkowie angażują się w nielegalne sprawy internetowe. Ten rodzaj programu szpiegującego szpieguje aktywność użytkownika, od naciśnięć klawiszy po zrzuty ekranu, i wysyła dziennik aktywności przez sieć do określonej strony logowania. Oprogramowanie szpiegujące można zidentyfikować za pomocą programów diagnostycznych, takich jak Ad-Aware firmy Lavasoft (www.lavasoft.com), ale ze względu na dynamiczny charakter technologii spyware określenie wszystkich możliwości infekcji może być trudne.

Ostatnim ryzykiem po stronie klienta, na które firmy powinny szczególnie zwracać uwagę, jest ryzyko złośliwych plików wykonywalnych działających na ich stacjach roboczych. Komputer stacjonarny jest jak szalka Petriego dla oprogramowania: ciągle się zmienia i rośnie wraz z nowymi plikami wykonywalnymi oprogramowania - niektóre z nich mają nietypowy charakter. Złośliwe oprogramowanie lub złośliwe oprogramowanie, jak jest obecnie znane, znajduje wiele sposobów infekowania maszyn. Na przykład jednym z powszechnych sposobów rozpowszechniania złośliwego oprogramowania jest załączanie wiadomości e-mail. Innym jest maskarada jako legalne oprogramowanie na stronie internetowej dostępnej do pobrania. Użytkownicy często przesyłają i pobierają oprogramowanie do i z wewnętrznych udziałów sieciowych. Ponadto „pendrive" USB, a nawet staromodne dyskietki są nadal realnym sposobem na złośliwe oprogramowanie. „Zestaw do zdalnej administracji" Back Orifice jest dobrze znanym przykładem złośliwego oprogramowania, które po zainstalowaniu umożliwia każdemu - w tym hakerowi - zdalne administrowanie i kontrolowanie maszyny. Niektóre produkty złośliwego oprogramowania, działającego na komputerze wewnętrznym może zagrozić całej sieci, ponieważ zdalny haker może kontrolować zaufany zasób, prawdopodobnie z prawami administracyjnymi przedsiębiorstwa. Wyciek danych i kradzież oferowane przez początkową infekcję spyware mogą powodować incydenty, które mogą podważyć integralność, autentyczność, dostępność lub kontrolę całej infrastruktury danych. Dlatego ważne jest, aby korporacje dokładnie filtrowały i ściśle monitorowały oprogramowanie aplikacji, które jest pobierane i uruchamiane (z wiedzą użytkownika lub bez) na wszystkich systemach, od obwodu e-commerce po poszczególne hosty wewnętrzne.

Zagrożenia protokołu sieciowego.

Zagrożenia sieciowe wynikają przede wszystkim z wysyłania poufnych danych przez Internet - publiczną sieć z przełączaniem pakietów. Wiele dobrych protokołów dotyczy ryzyka związanego z wysyłaniem poufnych danych przez Internet. Kilka lat temu lista zawierała:

* SET

* SSL

* S / HTTP

* S / MIME

* CyberCash

Chociaż niektóre z tych protokołów są nadal dostępne w takiej czy innej formie, przemysł ogólnie zaakceptował SSL jako protokół wyboru dla bezpiecznych transakcji w przeglądarce internetowej.

Celem najbardziej bezpiecznych protokołów sieciowych jest nałożenie warstw zabezpieczeń na warstwy sieciowe TCP / IP. Chociaż protokół TCP / IP zapewnia niezawodne i niezawodne dostarczanie datagramów przez Internet, nie zapewnia poufności, uwierzytelniania ani silnych usług integralności wiadomości. Są to właściwości zapewniane przez bezpieczne protokoły. Niektórzy idą nawet dalej. Na przykład protokoły zgodne z SET pozostawiają zaszyfrowany numer karty kredytowej nawet na stronie sprzedawcy. Ponieważ sprzedawca nie musi znać numeru karty kredytowej konsumenta, ukrywając ten numer przed sprzedawcą, można wyeliminować znaczną część oszustw związanych z kartami kredytowymi. Zamiast odszyfrowywać numer karty kredytowej w witrynie akceptanta, jest on przekazywany w formie zaszyfrowanej od akceptanta do banku wydającego kredyty. Tam jest odszyfrowywany, a konto handlowca jest uznawane za kwotę zakupu. Szczegóły protokołu SET, SSL i innych protokołów handlu elektronicznego są opisane w artykule Bezpieczeństwo handlu elektronicznego: Słabe linki, Najlepsza ochrona, a także w innych książkach. W zależności od potrzeb aplikacji biznesowej online istnieje zapotrzebowanie na mniej lub więcej właściwości bezpieczeństwa zapewnianych przez bezpieczne protokoły. W przypadku większości przeglądarek internetowych bezpieczny protokół nie jest konieczny; wystarczy protokół standardWeb, HTTP. Jednak gdy klienci przesyłają poufne lub osobiste informacje do witryny, preferowany jest bezpieczny protokół, który szyfruje dane. De facto bezpiecznym standardem protokołu jest SSL, teraz zaimplementowany w każdej standardowej przeglądarce internetowej. SSL nie tylko negocjuje tajny klucz sesji między Stroną a klientem w celu zaszyfrowania danych, ale także uwierzytelnia Stronę. Witryna musi mieć ważny certyfikat zatwierdzony przez urząd certyfikacji, któremu użytkownicy domyślnie ufają. Korzystając z listy zaufanych urzędów certyfikacji prowadzonych w przeglądarce klienta, można zapobiec dostępowi do innych niezaufanych witryn. Po ustanowieniu połączenia użytkownik może sprawdzić, czy strona internetowa jest rzeczywiście zgodna z zamierzeniem, sprawdzając certyfikat witryny. Użytkownicy rzadko robią to w praktyce, ale certyfikat jest dostępny i należy go szerzej wykorzystywać. Te bezpieczne właściwości - czyli szyfrowane sesje i uwierzytelnianie strony hosta - służą do większości aplikacji handlu online. Jednak niektóre aplikacje mogą wymagać jeszcze silniejszych usług bezpieczeństwa. Na przykład aplikacje bankowe i inwestycyjne często przesyłają wysoce poufne informacje, z możliwością ogromnych strat finansowych w wyniku nieumyślnego błędu lub celowego oszustwa. Tego rodzaju transakcje wymagają nie tylko poufności danych, ale także uwierzytelnienia klienta. Instytucja finansowa nigdy nie może udzielać dostępu do informacji o koncie ani zezwalać na transakcje bez uwierzytelnienia użytkownika. Typowe schematy identyfikacji w Internecie obejmują proste uwierzytelnianie nazwy użytkownika i hasła. O wiele bezpieczniejszym rozwiązaniem jest wymaganie silnego uwierzytelnienia klienta przy użyciu certyfikatów klienta. SSL obsługuje certyfikaty klienta, chociaż witryny rzadko korzystają z tej możliwości, ponieważ wymaga to od klientów uzyskania certyfikatu od urzędu certyfikacji. W przyszłości protokoły handlu elektronicznego będą musiały być coraz bardziej wyrafinowane, jeśli mają spełniać rygorystyczne wymogi bezpieczeństwa i prywatności nowych aplikacji internetowych. Na przykład, gdy migracyjne, medyczne, patentowe i inne ważne bazy danych migrują do Internetu, protokoły opracowane w celu uzyskania do nich dostępu muszą uwzględniać potrzeby w zakresie bezpieczeństwa i prywatności właścicieli i opiekunów baz danych, a także klienta lub osoby żądającej. Dzisiaj postęp w genomice dostarcza wielu informacji na temat prawdopodobieństwa rozwoju śmiertelnej choroby na podstawie sekwencji genetycznej DNA. Ta wiedza rodzi pytania moralne i etyczne dotyczące tego, ile informacji o potencjalnych chorobach powinna być ujawniona lekarzom i pacjentom, a także podnosi spektrum takich informacji dostających się w niepowołane ręce, gdy są one dostępne w Internecie. Rozważ przypadek, w którym lekarz odpytuje internetową bazę danych chorób genetycznych o sekwencję DNA pacjenta. Aplikacja online próbuje dopasować sekwencję DNA do chorób, które mogą się rozwinąć w przyszłości. Gdyby baza danych była prowadzona przez podmiot komercyjny, taki jak ubezpieczyciel, pacjent prawie na pewno nie chciałby, aby firma wiedziała o jakiejkolwiek chorobie, która mogłaby zostać zwrócona w wyniku

zapytania, ponieważ informacje te mogłyby zostać wykorzystane do odmowy ubezpieczenia i zatrudnienie. Podobnie opiekun bazy danych prawdopodobnie nie chciałby wiedzieć o zapytaniu lub jego wyniku, ponieważ taka wiedza może narazić ją na ryzyko w przypadku wycieku informacji. Ponadto opiekun bazy danych chciałby, aby reszta bazy danych pozostała niedostępna, z wyjątkiem określonych wyników zwróconych do zatwierdzonego zapytania. Uniemożliwienie dostępu do jakichkolwiek innych informacji w bazie danych pomogłoby chronić interesy handlowe firmy, ponieważ wówczas bazy danych nie można łatwo powielić. Nie można też zadawać pytań bez mechanizmu śledzenia kosztów. Aby wesprzeć ten podwójny model bezpiecznego i prywatnego dostępu do informacji, należy opracować i wprowadzić na rynek protokoły handlu elektronicznego, które nie tylko szyfrują przesyłane dane, ale także uwzględniają całkowite potrzeby w zakresie bezpieczeństwa i prywatności obu stron transakcji. Kolejny obszar aplikacji handlu elektronicznego, który będzie wymagał lepszych protokołów bezpieczeństwa i prywatności, obejmuje aplikacje akceptujące płatności elektroniczne lub elektroniczne. Obecnie większość schematów płatności online korzysta z kart kredytowych lub debetowych, przy czym płatności są dokonywane z konta czekowego kupującego w banku. Płatności są dokonywane za pomocą weryfikacji środków online lub płatności partiami w trybie off-line na koniec dnia. Tworzonych jest wiele aplikacji, zwłaszcza dotyczących płatności w wysokości kilku dolarów, a nawet groszy, które nie są w stanie pokryć kosztów transakcji bankowej. Wiele komercyjnych usług i produktów, takich jak automaty vendingowe i parkomaty, są aktywowane monetami i nie wymagają od klientów posiadania konta ani linii kredytowej. Chociaż podejmowane są wysiłki w celu przekształcenia takich usług w urządzenia skomputeryzowane, z możliwością mikropłatności, upłynie wiele lat, zanim zostanie to faktycznie zrealizowane. W każdym razie zawsze znajdą się klienci, którzy chcą zapłacić gotówką lub jej ekwiwalentem elektronicznym, aby transakcja nie mogła być śledzona przez osobę trzecią, takich jak bank lub wierzyciel. Nowsze aplikacje online do mikropłatności mogą obejmować pobieranie opłat za pobieranie muzyki, dane, prognozy pogody, notowania giełdowe, artykuły z czasopism i strony z książek. Wiele z tych aplikacji jest obecnie udostępnianych bezpłatnie i wspieranych przez banery reklamowe, ale troska o zyski motywuje większość stron internetowych do poszukiwania dodatkowych źródeł dochodów. Niezależnie od zastosowania, potrzebne są alternatywy gotówkowe w stosunku do obecnego systemu płatności na rachunku bieżącym. Kluczowe obawy związane z bezpieczeństwem i prywatnością polegają na zapewnieniu, że e-gotówka jest właściwie wydawana i rozliczana oraz że zachowana jest anonimowość. Z myślą o tych celach opracowano kilka protokołów, ale żaden nie osiągnął komercyjnego sukcesu ani nie został przyjęty przez społeczność dostawców. Ponieważ mobilny handel elektroniczny zaczyna prowadzić bardziej tradycyjnie transakcje gotówkowe (np. parkomaty, automaty i kasy biletowe), dostawcy usług bezprzewodowych mogą przyjąć te nowe cyfrowe protokoły kasowe. Niezależnie od protokołu sieciowego używanego w aplikacji e-commerce, kluczową kwestią są osoby atakujące, które będą próbowały przełamać najłatwiejszą przeszkodę w dążeniu do uzyskania uprawnień systemowych i nieautoryzowanego dostępu do danych. postrzegani jako silni, atakujący będą szukać alternatyw, które omijają bezpieczeństwo sieci. Na przykład tego rodzaju standardowe ataki z zestawu narzędzi hakera omijają zabezpieczenia zapewniane przez większość protokołów e-commerce:

* Ataki typu man-in-the-middle. Przechwytywanie transmisji w tranzycie w celu podsłuchu lub fałszowania

* Ataki DNS. Zmiana rekordów w światowym systemie nazw domen w celu przekierowania połączeń do niewłaściwych adresów

* War dialing. Zautomatyzowane testowanie każdego numeru telefonu w bloku liczb

* Wykorzystywanie luk w zabezpieczeniach oprogramowania w usługach sieciowych. Takie jak serwery FTP, Bind, SMTP i HTTP

* Dostęp wewnętrzny. Niewłaściwe korzystanie z autoryzowanego dostępu przez osobę z zewnątrz (pracownika, kontrahenta)

* Wykorzystanie zaufanych hostów. Atak z innego systemu, który ma uprzywilejowaną relację z systemem docelowym

* Ataki kryptograficzne Brute-force. Zautomatyzowane testowanie wszystkich możliwych kluczy odszyfrowywania w celu odszyfrowania tekstu zaszyfrowanego

Podsumowując, ważne jest nie tylko wybranie odpowiedniego protokołu sieciowego dla każdej aplikacji online, ale także unikanie fałszywego poczucia bezpieczeństwa, które mogłoby wynikać z używania „bezpiecznego" protokołu sieciowego. Dobra inżynieria bezpieczeństwa weźmie pod uwagę luki w innych elementach systemu, które są bardziej atrakcyjnym celem dla zdeterminowanych hakerów.

Logika aplikacji biznesowych.

Logika aplikacji biznesowych stanowi jeden z kluczowych obszarów podatności na zagrożenia w systemach e-commerce. Logika programu koduje to, co oferuje firma internetowa pod względem produktów, usług i wygody klientów. Definiuje także wygląd i działanie Witryny oraz zapewnia wszystkie interaktywne funkcje, takie jak dynamiczne strony internetowe, spersonalizowane strony internetowe i możliwości transakcji online. Ponieważ każda aplikacja jest wyjątkowa, oprogramowanie, które implementuje logikę, musi być w wielu przypadkach opracowane na zamówienie dla każdej konkretnej witryny. Natomiast większość innych składników oprogramowania Witryny to komercyjne oprogramowanie gotowe (COTS). Na przykład serwer sieci Web, wewnętrzne bazy danych i oprogramowanie logistyczne dla łańcucha dostaw są często kupowane od dostawców z półki. Dzięki oprogramowaniu COTS użytkownik końcowy nie ma kontroli nad kodem i dlatego nie jest odpowiedzialny za kodowanie poprawek błędów. Po wykryciu błędów oprogramowania w oprogramowaniu COTS sprzedawca oprogramowania zazwyczaj wydaje łatkę lub wprowadza poprawkę błędu w następnej wersji. Dostawcy oprogramowania mogą naprawiać wykryte błędy, ale polegają na witrynach klientów, którzy faktycznie zastosują poprawki lub aktualizacje oprogramowania. W praktyce zdarza się to rzadziej niż jest to pożądane i jest istotnym powodem, dla którego wiele systemów internetowych jest podatnych na zagrożenia.10 Najważniejszym zadaniem w zabezpieczaniu systemów oprogramowania COTS jest upewnienie się, że: (1) są odpowiednio skonfigurowane do bezpiecznej instalacji zgodnie z polityką bezpieczeństwa witryny (2) oprogramowanie jest odpowiednio aktualizowane do bieżącej wersji i poziomu łaty, a (3) administratorzy są świadomi ryzyka związanego z błędami w oprogramowaniu. Ponieważ wiele aplikacji biznesowych jest opracowywanych na zamówienie, zarówno przez wewnętrzny personel, jak i przez outsourcing do programisty e-biznesu, programy stanowią kluczowe ryzyko z kilku powodów. Dynamiczny, interaktywny charakter e-biznesu w połączeniu z coraz bardziej wyrafinowanymi usługami internetowymi wymaga znacznego rozwoju w celu zakodowania logiki. W rezultacie aplikacje są zwykle bardzo złożonymi programami, które mogą zawierać wady i są podatne na ataki przeprowadzane na stronach internetowych. W praktyce błędy w projektowaniu i wdrażaniu logiki aplikacji biznesowych często zagrażają bezpieczeństwu e-biznesu. Tradycyjnie środkowa warstwa oprogramowania jest implementowana na serwerach Web przy użyciu CGI, a ostatnio PHP: Hypertext Processor (PHP). Skrypty CGI i PHP to programy, które działają na komputerze serwera Web jako oddzielne procesy od oprogramowania serwera Web. Serwer sieci Web wywołuje te programy ogólnego przeznaczenia w odpowiedzi na żądania użytkowników. Główną funkcją skryptu CGI / PHP jest przetwarzanie danych wejściowych użytkownika i wykonywanie niektórych usług, takich jak pobieranie danych lub dynamiczne tworzenie strony internetowej dla użytkownika końcowego.

Ponieważ skrypty CGI i PHP przetwarzają niezaufane dane wejściowe użytkownika, związane z nimi zagrożenia bezpieczeństwa i inne formy oprogramowania warstwy pośredniej są niezwykle wysokie. Wiele ataków na systemy oparte na sieci Web jest przeprowadzanych przez wyzyskiwane skrypty CGI / PHP. I chociaż skrypty CGI i PHP można pisać w dowolnym języku programowania ogólnego, najczęściej są pisane w językach Perl, C, Tcl i Python. Ostatnio oprogramowanie oparte na komponentach (CBS) zdobywa popularność zarówno w aplikacjach e-commerce, jak i standardowych aplikacjach internetowych. Celem CBS jest rozwój, zakup i ponowne wykorzystanie sprawdzonego oprogramowania w celu szybkiego, łatwego i wysokiej jakości wdrożenia logiki aplikacji. Dwoma bardziej popularnymi strukturami komponentów dla aplikacji e-commerce są Enterprise JavaBeans (EJB) korzystające z XML i Java 2 Enterprise Edition (J2EE), które obsługują Javę opartą na komponentach. Inne modele komponentów obejmują architekturę Common Object Request Broker Architecture (OMBA) grupy Object Management Group (OMG) oraz Common Object Model (COM) firmy Microsoft i COM COM (DCOM). Te struktury komponentów to klej, który umożliwia komponentom oprogramowania korzystanie ze standardowych usług infrastrukturalnych, a jednocześnie ukrywanie szczegółów implementacji przy użyciu dobrze zdefiniowanych interfejsów. Logika aplikacji biznesowych, gdy jest zakodowana w systemach CBS, zwykle działa na serwerach aplikacji z określonymi modelami komponentów, takimi jak EJB, CORBA, COM i DCOM. CBS zapewnia również interfejs do usług zaplecza, takich jak zarządzanie bazą danych, planowanie zasobów przedsiębiorstwa (ERP) i starsze systemy oprogramowania. Oprócz obsługi tradycyjnych funkcji CGI oczekuje się, że oprogramowanie oparte na komponentach umożliwi rozproszone aplikacje między przedsiębiorstwami przez Internet. Paradygmat oprogramowania opartego na komponentach obsługuje również dobrą inżynierię oprogramowania, jak opisano poniżej. Ujednolicony język modelowania (UML) ułatwia analizę obiektową i projektowanie w ramach opartych na komponentach. Ponadto wraz ze wzrostem rynku oprogramowania opartego na komponentach pojawia się wiele standardowych komponentów aplikacji biznesowych któ®e będzie można kupić z półki. Chociaż zalety oprogramowania opartego na komponentach są liczne, stanowią one zagrożenie bezpieczeństwa podobne do tych, jakie dają skrypty CGI. Umożliwia oprogramowanie oparte na komponentach programowania w językach programowania ogólnego przeznaczenia, takich jak Java, C i C ++, które można wykonywać ze wszystkimi prawami i uprawnieniami procesów serwera. Podobnie jak CGI, przetwarzają niezaufane dane wejściowe użytkownika, a ponieważ oprogramowanie oparte na komponentach może być używane do tworzenia wyrafinowanych aplikacji na dużą skalę, prawdopodobieństwo błędów może być nawet większe niż w przypadku prostych skryptów CGI. Niezależnie od implementacji - CGI lub serwerów aplikacji - zagrożenia bezpieczeństwa oprogramowania po stronie serwera są ogromne, dlatego też oprogramowanie po stronie serwera musi być projektowane i wdrażane ostrożnie. Główne zagrożenia w warstwie oprogramowania pośredniego witryn e-commerce to:

Błędna konfiguracja CGI

Domyślne i programistyczne skrypty CGI pozostawione na serwerze produkcyjnym

Niewłaściwe użycie CGI

Subversion aplikacji

Błędna logika

Błędy programowania

Luki w zabezpieczeniach skryptu CGI.

Skrypty CGI są często celem atakujących, ponieważ często są źle skonfigurowane i podatne na niewłaściwe użycie. Projektując skrypty CGI, rozsądne jest oczekiwanie nieoczekiwanego, szczególnie złośliwego ataku. Chociaż projektant stron ma kontrolę nad zawartością skryptów CGI, nie ma kontroli nad tym, co użytkownicy końcowi będą do nich wysyłać. Często pomijane są również podatności skryptów CGI, które istnieją na serwerze w ramach dystrybucji, ale które nie są faktycznie używane w aplikacji. Niektóre skrypty CGI, zawarte w dystrybucji serwera WWW, mają dobrze znane wady, które można wykorzystać w celu uzyskania nieautoryzowanego dostępu do serwera. Nawet jeśli domyślne skrypty CGI nie są używane jako część stron serwera WWW, każdy może uzyskać do nich dostęp po prostu znając nazwy skryptów. Jednym z najczęstszych, ale łatwych do uniknięcia zagrożeń bezpieczeństwa, jest błędna konfiguracja oprogramowania, zwłaszcza skryptów CGI. Jedną z funkcji obsługiwanych przez wiele serwerów Web jest zdolność osób w całej organizacji do pisania skryptów CGI i wykonywania ich z własnych katalogów. Ta funkcja, choć przydatna do porządkowania osobistych stron internetowych, może również powodować zagrożenia bezpieczeństwa systemu. W aplikacjach internetowych serwer sieci Web powinien być skonfigurowany tak, aby uniemożliwić wykonywanie skryptów CGI w dowolnym miejscu poza jednym katalogiem CGI pod kontrolą administratora systemu. Tryb CGI z aliasami skryptów dla serwerów WWW zapewnia, że skrypty CGI będą uruchamiane tylko z jawnie nazwanego katalogu w pliku konfiguracyjnym serwera. Ponadto ścieżka skryptu CGI nie jest nazwana w Uniform Resource Locator (URL) do CGI. Serwer raczej „aliasuje" jawną ścieżkę do skryptu CGI do wybranej nazwy, takiej jak cgi-bin. W ten sposób uruchomienie serwera w trybie CGI z aliasami skryptów zapobiega wykonywaniu nieuczciwych skryptów CGI, a także ukrywa jawną ścieżkę do skryptów CGI. Katalogi skryptów CGI również powinny być poprawnie skonfigurowane przy użyciu kontroli dostępu do systemu operacyjnego. Na przykład, jeśli skrypty CGI są napisane w skompilowanym języku, takim jak C, źródła skryptów powinny zostać wykluczone z katalogu głównego serwera WWW, aby nie można było uzyskać do nich dostępu przez Internet. Powinny być one dostępne tylko dla administratora systemu lub grupy opracowującej treści WWW i niedostępne dla wszystkich jeszcze w organizacji. Jeśli źródła skryptów wpadną w ręce złośliwych sprawców, kod źródłowy można sprawdzić pod kątem wad, co jeszcze bardziej ułatwia pracę sprawcy. Dostęp do katalogu plików wykonywalnych CGI, często nazywanego cgi-bin, również powinien być odpowiednio kontrolowany. Tylko serwer WWW i administrator potrzebują dostępu do tego katalogu. Liberalne uprawnienia dostępu do katalogu plików wykonywalnych CGI dają złośliwym zamiarom możliwość umieszczania własnych skryptów w witrynie. Większość skryptów CGI jest napisanych w językach skryptowych, takich jak Perl, JavaScript i Python. Języki skryptowe są przydatne do szybkiego prototypowania systemów, ale także pozwalają programiście bardzo łatwo napisać niebezpieczny kod. Na przykład łatwo jest konstruować polecenia systemowe na podstawie danych wprowadzonych przez użytkownika, co jest potencjalnie niebezpieczną sytuacją. Napisanie tej samej funkcjonalności systemu wymaga kilku linii kodu programowania i znajomości bibliotek systemowych. Łatwa dostępność języków skryptowych sprawia, że są one atrakcyjne, ale także zagraża aplikacjom o kluczowym znaczeniu dla bezpieczeństwa. Ważne jest również, aby zakazać dostępu do tłumaczy z serwera WWW. Na przykład administratorzy systemu mogą ulec pokusie włączenia interpretera Perla do katalogów skryptów CGI; jednak takie działanie zapewnia bezpośredni dostęp do Internetu w celu interaktywnego wykonywania poleceń Perla – niezwykle niebezpieczna konfiguracja. Wreszcie, administratorzy powinni wziąć pod uwagę każdy program CGI na serwerze pod względem jego celu, pochodzenia i modyfikacji. Usuń skrypty CGI, które nie spełniają funkcji biznesowej, i podejrzliwie przeglądaj skrypty CGI, które są dystrybuowane z systemami operacyjnymi i serwerami WWW, pobrane z Internetu lub zakupione komercyjnie. Kroki te wyeliminują większość potencjalnie niebezpiecznych skryptów CGI. Po

ustanowieniu stabilnego zestawu programów CGI wykonaj cyfrowy skrót plików wykonywalnych programu (np. Przy użyciu MD5 lub SHA-1), aby umożliwić przyszłe sprawdzanie integralności.

Subwersja aplikacji.

Ataki polegające na wywróceniu aplikacji nie są często omawiane w odniesieniu do e-biznesu, ale stanowią poważne zagrożenie dla większości aplikacji internetowych. Subwersja aplikacji jest formą niewłaściwego użycia programu. W przeciwieństwie do ataków przepełnienia bufora, ataki subwersji aplikacji wykorzystują logikę programu bez naruszenia integralności programu, w celu podniesienia uprawnień użytkownika i uzyskania nieautoryzowanego dostępu do danych. To właśnie złożoność programu docelowego daje atakującemu możliwość uzyskania nieautoryzowanego dostępu. Ataki subwersji aplikacji wykorzystują programy w sposób, którego projektanci i programiści nie przewidzieli. Zazwyczaj ataki te nie są wykonywane skryptami, ale są opracowywane na podstawie interaktywnego wykorzystania online i późniejszych nadużyć.

Atak subwersji aplikacji będzie próbował odkryć sposoby zwierania ścieżek w przepływie pracy. Na przykład może istnieć ukryta ścieżka, która pozwala użytkownikowi uzyskać dostęp do informacji o koncie bez uwierzytelnienia na tym koncie. Wiele takich ataków działa przy założeniu, że dostęp do poufnych informacji nie jest odpowiednio uwierzytelniony. Kolejny typowy atak wysyła zniekształcone dane wejściowe do programu. Wiele stron WWW często używa formularzy do sterowania aplikacją, podczas gdy dane wprowadzane do formularza są sprawdzane za pomocą skryptów po stronie klienta. Osoba atakująca może skorzystać z faktu, że wielu programistów aplikacji internetowych zakłada, że klient będzie prawidłowo korzystał z formularza i że skrypty sprawdzą wszystkie dane wejściowe wysyłane do witryny. Atakujący może zbadać strumień danych wysłany przez formularz, a następnie, zamiast używać formularza, wysłać zmodyfikowany strumień danych za pośrednictwem adresu URL w przeglądarce lub włączyć go do niestandardowego polecenia wybranego przez atakującego. Osoba atakująca może uzyskać dostęp do aplikacji, umieszczając polecenia systemowe w strumieniu wejściowym. Jeśli strumień wejściowy jest następnie wykorzystywany przez aplikację online w wywołaniu system (), użytkownik końcowy może wymusić wykonanie poleceń systemowych w imieniu atakującego. Niektórzy twórcy aplikacji polegają w dużej mierze na ukrytych polach w dokumencie HTML. Ukryte pola pozwalają twórcy stron internetowych na umieszczenie informacji na stronie, która nie jest wyświetlana, chociaż użytkownik końcowy może zobaczyć dane ukrytego pola po prostu przeglądając źródło HTML. Błędem, który popełniają twórcy aplikacji, jest po pierwsze przekonanie, że użytkownik końcowy nie widzi ukrytych pól, a po drugie, poleganie na integralności danych ukrytych pól przy podejmowaniu decyzji online. Dokonali tego niektórzy kupcy online , błąd polegający na podaniu informacji o cenach produktów w ukrytych polach i wykorzystaniu tych cen do ustalenia kosztu transakcji online. Użytkownik końcowy może po prostu zmienić ceny w ukrytych polach i odesłać niższe ceny do akceptantów w celu zakupu ze zniżką. Innym niewłaściwym wykorzystaniem ukrytych pól jest przekierowanie danych wyjściowych aplikacji. Na przykład niektóre strony internetowe zawierają informacje o ścieżce systemu plików w ukrytych polach na swoich stronach internetowych. Informacje te są wykorzystywane przez skrypt po stronie serwera do określania, gdzie należy czytać lub pisać

Informacja o transakcji. Atakujący, po prostu zmieniając ukryte pole, mogą nadpisywać pliki lub czytać pliki, do których nie powinni mieć dostępu. W niektórych przypadkach może być możliwe zapisanie programu wpisanego w polu formularza, który będzie później używany jako sposób uruchamiania uprzywilejowanej powłoki w systemie zdalnym. Podsumowując, rygorystyczna kontrola jakości oprogramowania jest niezbędna podczas projektowania i rozwoju wszystkich aplikacji internetowych i e-biznesowych, w tym interfejsu użytkownika

Strony internetowe, oprogramowanie pośrednie aplikacji i systemy operacyjne. Gdy oprogramowanie zostanie uznane za odporne na niewłaściwe użycie aplikacji i ataki subwersyjne, administrator systemu musi wykonać inne czynności, aby zapewnić bezpieczeństwo oprogramowania pośredniego dla e-biznesu:

* Wszystkie niepotrzebne skrypty lub programy serwera aplikacji muszą zostać usunięte z serwera produkcyjnego.

* Kod źródłowy oprogramowania pośredniego aplikacji musi być ostrożnie chroniony przed pobraniem lub nieautoryzowanym dostępem.

* Prawidłowa konfiguracja CGI i oprogramowania pośredniego jest niezbędna, aby zapewnić dostęp do wykonywalnego oprogramowania tylko do właściwego oprogramowania pośredniego, z najniższym praktycznym poziomem uprawnień. Należy sprawdzić poprawność danych wejściowych do oprogramowania pośredniego aplikacji, aby upewnić się, że akceptowane są tylko poprawnie sformułowane dane wejściowe.

* Testowanie kodu aplikacji przy użyciu programów takich jak WebInspect firmy SPI Dynamics (www.spidynamics.com), które są specjalnie zaprojektowane do wykrywania wad w aplikacjach internetowych.

Wykorzystuje serwer WWW.

Bezpieczeństwo serwera WWW zostało opisane i szczegółowo omówione, w tym między innymi w E-Commerce Security: Słabe linki, Najlepsza ochrona i Książka źródłowa bezpieczeństwa internetowego. W tym miejscu podkreślamy niektóre z typowych exploitów serwerów sieciowych wykorzystywanych przeciwko wielu e-firmom.

Konfiguracja.

Kluczem do bezpieczeństwa serwera WWW jest jego konfiguracja. Podobnie jak inne złożone elementy oprogramowania, serwery WWW są wysoce konfigurowalne, aby sprostać potrzebom dowolnej witryny. Domyślnie większość dostawców oprogramowania konfiguruje aplikację pod kątem maksymalnej funkcjonalności, ale minimalnego bezpieczeństwa. Dlatego domyślnie, gdy serwer jest uruchamiany po raz pierwszy, prawdopodobnie będzie bardziej liberalny, niż by tego wymagała polityka bezpieczeństwa danej firmy. Podstawowym założeniem konfigurowalności są różnice występujące między witrynami. Prawidłowo skonfigurowany serwer WWW jest wynikiem zasad, które określają, jaki dostęp jest dozwolony, do jakich osób, dla każdego zasobu. Ta zasada z kolei służy do konfigurowania routerów, zapór i wszystkich serwerów publicznych, takich jak serwer WWW. Skonfigurowanie serwera WWW, choć jest konieczne, w żadnym wypadku nie wystarcza do zabezpieczenia systemu.

Kodowanie HTML i obejmowanie po stronie serwera.

Po skonfigurowaniu serwera WWW tak bezpiecznie, jak to możliwe, ważne jest, aby same strony internetowe nie otwierały luk w zabezpieczeniach. Wielu programistów stron internetowych ma pewne typowe pułapki, które mogą zagrozić bezpieczeństwu witryny. W poprzedniej sekcji wspomniano o problemie polegania na ukrytych polach w HTML dla bezpieczeństwa lub danych krytycznych dla biznesu. Użytkownicy mogą nadużywać danych ukrytych pól w celu obalenia aplikacji. HTML oferuje inne potencjalne luki w zabezpieczeniach. Najczęściej krytykowany jest serwer po stronie (SSI). Opcja konfiguracji SSI, jeśli jest włączona, umożliwia osadzenie dyrektyw w HTML, które serwer

wykona. Na przykład, jeśli następna instrukcja została osadzona w dokumencie HTML, poleciłby serwerowi wyświetlenie zawartości pliku haseł systemowych:

<! - # exec / bin / cat / etc / passwd ->

Z pewnością strony internetowe nie powinny być pisane przy użyciu SSI bez ważnego powodu. Chociaż dostęp do kodu HTML jest zwykle pod kontrolą witryny, istnieje wiele sposobów, w jakie osoba atakująca może uzyskać SSI na stronie HTML. Po pierwsze, osoba atakująca mogła znaleźć inną drogę do systemu (np. Poprzez exploit skryptu CGI), ale może chcieć zapewnić łatwiejsze backdoor lub redundantne backdoor na wypadek znalezienia i zamknięcia luki w skrypcie CGI. Po wykorzystaniu skryptu CGI osoba atakująca może wdrożyć dyrektywę SSI na jednej ze stron HTML witryny. Innym sposobem uzyskania dostępu przez atakującego jest wygenerowanie przez serwer strony internetowej z wybranym SSI wbudowanym w HTML. Jak atakujący może to zrobić? Jedno podejście wykorzystuje serwer, który generuje dynamiczny HTML w zależności od danych, preferencji lub historii użytkownika końcowego. Jeśli serwer internetowy wykorzysta niektóre dane atakującego do wygenerowania strony HTML, może on być w stanie wstawić dyrektywę SSI do HTML. Podsumowując, lepszym rozwiązaniem niż ciągłe monitorowanie stron HTML jest po prostu wyłączenie SSI. W takim przypadku, nawet jeśli SSI byłyby osadzone, serwer nie wykonałby ich. Jest to opcja konfiguracyjna i podobnie jak wszystkie pliki konfiguracyjne systemu, plik konfiguracyjny serwera WWW powinien być chroniony zarówno przez ochronę uprawnień do plików, jak i sprawdzanie integralności plików, aby zapewnić, że nie można go sfałszować. Chociaż SSI są często wyróżniane, bardziej powszechne ryzyko wynika z przechowywania dokumentów lub plików w publicznie dostępnej części serwera WWW. Dostępna część serwera WWW nosi nazwę katalogu głównego dokumentu. Ten katalog główny określa część systemu plików, którą serwer Web może odczytać i wyświetlić klientowi Web na żądanie. Katalog główny może być nadzbiorem stron internetowych, które są faktycznie wyświetlane, gdy użytkownik kliknie stronę internetową. W katalogu głównym dokumentu mogą znajdować się inne dokumenty, które nie są powiązane ze stronami internetowymi ani ze stron internetowych. Nie oznacza to jednak, że nie są one dostępne. Podanie poprawnego adresu dla dowolnego dokumentu spowoduje wyświetlenie lub pobranie dokumentu do dowolnego klienta WWW. Na tym polega problem. Warto zauważyć, że treści HTML nie można już znaleźć tylko w plikach obsługiwanych przez Internet. Na lokalnym dysku twardym aplikacje Microsoft Windows często instalują dokumenty Pomocy składające się z plików CHM („chum") i plików binarnych. Osoba atakująca może wykorzystać tę lukę, co prowadzi do podniesienia uprawnień do strefy „Mój komputer" w systemie Windows XP; niestety lokalne pliki są często pomijane i uznawane za „zaufane" podczas testów penetracyjnych. Uznając te zagrożenia, Microsoft Corp. zaczął używać języka Microsoft Assistance Markup Language w systemie operacyjnym Windows Vista.

Prywatne dokumenty w katalogach publicznych.

Prywatne dokumenty przypadkowo umieszczone w publicznym katalogu mogą doprowadzić do naruszenia poufności informacji i utraty prywatności. Na przykład, jeśli plik bazy danych z numerami kart kredytowych (powiedzmy cardnumbers.mdb) był przechowywany w katalogu głównym serwera WWW dla fikcyjnej firmy, mojafirma.com, adres URL wpisany w przeglądarce internetowej mógłby pobrać plik: www.mycompany.com/cardnumbers.mdb

Ryzyko to jest jeszcze większe, jeśli włączone jest przeglądanie katalogów, inna konfigurowalna funkcja. Przeglądanie katalogów umożliwia użytkownikowi końcowemu przeglądanie zawartości systemu plików na danym poziomie katalogu, jeśli strona internetowa nie istnieje o tej samej nazwie. Przeglądanie katalogów jest naprawdę sposobem na eksplorację systemu plików innej witryny za pomocą przeglądarki internetowej. Użytkownicy mogą wyświetlić tę funkcję, jeśli cofną się o jeden poziom z danej strony internetowej, usuwając prawą sekcję adresu URL strony i przeglądając jej katalog

wyższego poziomu (np. W http: //abcom/d/e.htm, można usunąć „e.htm”, próbując w ten sposób przeglądać http: // abcom / d /). Atakujący mogą dowiedzieć się wielu cennych informacji podczas przeglądania zawartości katalogu, w tym prywatne pliki. Ponadto sama przeglądarka umożliwia kliknięcie nazwy pliku w strukturze katalogów, co powoduje pobranie pliku. Ponownie, przeglądanie katalogów, jeśli jest włączone, stanowi otwartą lukę i niestety łatwy sposób na pobranie prywatnych lub poufnych informacji

Pliki cookie i inne zagrożenia po stronie klienta.

Inną potencjalną podatnością na zagrożenia dla e-biznesu jest stosowanie plików cookie. Ponieważ HTTP jest protokołem bezstanowym, każda odwiedzana nowa strona WWW nie ma pamięci o ostatniej stronie odwiedzonej przez tego użytkownika. Pliki cookie służą do „utrzymywania stanu” między różnymi stronami odwiedzanymi w danej sesji. Pliki cookie mogą sprawiać, że transakcja e-biznesowa wydaje się płynna, sekwencyjna i skoordynowana. Większość ludzi, omawiając ryzyko związane z plikami cookie, koncentruje się na zagrożeniach prywatności po stronie klienta. Chociaż z pewnością istnieją, pliki cookie stanowią również ryzyko dla firm, które je zatrudniają. Jeśli informacje zawarte w plikach cookie są zaufane, podobnie jak treść w ukrytych polach jest zaufana, e-biznes może być podatny na exploity związane z plikami cookie zwane zatruwaniem plików cookie. Niektóre strony internetowe używają plików cookie do przenoszenia informacji uwierzytelniających dla danego użytkownika, który przegląda jego strony. Gdy użytkownicy się uwierzytelnią, ich token uwierzytelnienia może być przenoszony przez pliki cookie z jednej strony internetowej na wszystkie kolejne strony w tej witrynie. Używanie plików cookie jest dość słabą formą uwierzytelniania. Plik cookie może zostać łatwo skradziony przez osobę szpiegującą lokalną sieć użytkownika lub Internet, a następnie, po uzyskaniu informacji, w celu uzyskania dostępu do osobistych stron użytkownika w Witrynie. W tym celu należy zastosować bezpieczne protokoły, takie jak SSL aby ograniczyć to ryzyko. Pliki cookie są również wykorzystywane do innych celów, które mogą wprowadzić nowe luki w transakcjach o znaczeniu krytycznym. Ponieważ pliki cookie są kontrolowane przez użytkowników końcowych, można je zmieniać w dowolny sposób wybrany przez użytkownika. Jeśli pliki cookie mają na celu instruowanie serwera WWW, gdzie należy zapisać plik specyficzny dla klienta, zmieniając dane cookie, użytkownik końcowy może zastąpić inne pliki klienta lub nawet zastąpić krytyczne pliki systemowe. Podobnie, jeśli pliki cookie są używane do przenoszenia informacji o zamówieniu, co jest powszechne w elektronicznych koszykach, zmiana zawartości plików cookie może zaszkodzić transakcji. Może to spowodować nieautoryzowaną głęboką zniżkę dla klienta. Ten przykład i inne luki w skryptach między witrynami są dość powszechne w dzisiejszej sieci. Exploity przeglądarki, wstrzykiwanie nagłówka HTTP, fałszowanie żądań w różnych witrynach (CSRF) oraz liczne rodzaje phishingu to przykłady ataków klienta / użytkownika WWW. Bez względu na zastosowaną technologię ważne jest sprawdzenie funkcji i funkcji serwerów WWW z krytycznego punktu widzenia bezpieczeństwa. Zależność od konkretnej technologii w transakcjach krytycznych wymaga, aby technologia była godna zaufania, aby nie zapewniała wrażliwych punktów ataku.

Osadzone identyfikatory w adresach URL.

Czasami wiadomości rejestracyjne lub anulujące subskrypcję zawierają spersonalizowane adresy URL; na przykład „Aby anulować subskrypcję, kliknij http://www.some-company.com/unusbscribe/? = 12345 ”Problem polega na tym, że kod identyfikacyjny (12345 w przykładzie) można zastąpić dowolnym innym numerem - a niektóre z wygenerowanych numerów anulują subskrypcję innych subskrybentów. Inne niefortunne użycie takiej techniki występuje w wiadomościach e-mail dotyczących uczestnictwa w seminariach internetowych; czasami zakodowany adres URL w wiadomości e-mail przechodzi bezpośrednio na stronę rejestracji z wypełnionymi informacjami - takimi jak dokładna nazwa, adres e-mail i informacje o pracodawcy zaczerpnięte z bazy danych poprzednich

kontaktów. Ogólnie rzecz biorąc, złą praktyką jest podawanie tak krótkich identyfikatorów, że użytkownik lub prosty program lub plik wsadowy mogą zostać użyte do usunięcia prawidłowych rekordów w bazie danych lub do zebrania poufnych danych ze strony internetowej. Na przykład, prosty plik HTML w Adobe Acrobat może być użyty do zmuszenia programu do dostępu do każdego adresu URL w pliku; jeśli złoczyńca utworzył, powiedzmy, 20 000 unikalnych identyfikatorów i wszystkie lub większość z nich jest powiązana z prawdziwymi ludźmi, złoczyńca może z łatwością wpłynąć na konta lub zebrać informacje wyświetlane na stronach specyficznych dla klienta. Aby uniknąć takich problemów, utwórz przestrzeń adresową o znacznie większej możliwej wartości identyfikatora niż odwzorowanie rzeczywistych kont 1: 1 na identyfikatory; np. w przypadku listy obejmującej 100 000 osób wartości wygenerowane przez użycie przy użyciu ciągu 20 pozycji z 94 wartościami na pozycję dostępnymi wielkimi, małymi literami, cyframi i znakami specjalnymi, co daje $94^{20} = 10^{39}$ kluczy, znacznie przekraczających wszystko, co może zrobić amator .

Bezpieczeństwo bazy danych.

Bazy danych tradycyjnie stanowiły ważną własność intelektualną firm informatycznych. W rezultacie prawie zawsze były własnością firmy i były niedostępne dla publicznego dostępu. Jednak w nowym modelu działalności wiele z tych zastrzeżonych baz danych jest udostępnianych w Internecie, często bez uważnego rozważenia związanego z nimi ryzyka. Przeglądarka internetowa staje się interfejsem zapytań do bazy danych, często dla nieznanych i niezaufanych podmiotów. Chociaż w ciągu ostatnich dwóch dziesięcioleci przeprowadzono wiele badań dotyczących bezpieczeństwa baz danych, sektor komercyjny przyjął tylko dwie kluczowe zasady: uwierzytelnianie i autoryzowanie zleceniodawców do niektórych obiektów. Dostęp do baz danych jest zatem kontrolowany poprzez prawidłowe uwierzytelnienie poświadczeń zleceniodawcy, a następnie sprawdzenie, do których obiektów uprawniony jest dostęp. Każda internetowa aplikacja bazy danych musi rygorystycznie wykonywać te dwie funkcje, aby chronić najcenniejsze aktywa e-biznesu, a także prywatność klientów. Chociaż wielu dostawców twierdzi, że bezpieczny dostęp do kanału z serwera sieci Web do bazy danych, istnieje wiele pułapek. Po pierwsze, podstawowym powodem, dla którego bazy danych są podatne na atak, jest to, że interfejsy sieciowe są często dołączane do czegoś, co kiedyś mogło być zamkniętym i zastrzeżonym interfejsem, bez troski o ogólne bezpieczeństwo. Po drugie, niezabezpieczone programy pośrednie, takie jak skrypty CGI lub serwery aplikacji, zwykle pośredniczą w dostępie z serwera sieci Web do bazy danych. Serwery sieciowe mogą zapewniać uwierzytelnianie klienta od prostej nazwy użytkownika i hasła do silnego uwierzytelnienia certyfikowanego. Potrzeby biznesu i wielkość dostępnej społeczności będą decydować o tym, które rozwiązanie jest możliwe. Pomimo oczywistych korzyści bezpieczeństwa większość użytkowników nie przechowuje zaszyfrowanych informacji w swoich bazach danych, głównie ze względu na wydajność. Szyfrowanie i deszyfrowanie informacji w locie podczas operacji wyszukiwania, pobierania i przechowywania może być zbyt wolne dla transakcji w czasie rzeczywistym. Również nawet szyfrowanie danych w jednostce pamięci nie zapewni pełnej ochrony, ponieważ aplikacja online musi mieć możliwość odczytu i zapisu do bazy danych w postaci czystego tekstu. Ataki oparte na aplikacjach nadal byłyby w stanie uzyskać dostęp do danych, gdy były one w formacie niezaszyfrowanego tekstu. Inna kluczowa luka w internetowych bazach danych wynika również z ataków opartych na aplikacjach. Jak opisano wcześniej, ataki wykorzystujące luki w zabezpieczeniach aplikacji biznesowych często zapewniają nieograniczony dostęp do bazy danych. Ataki wykorzystujące lukę przepełnienia bufora w programie serwera aplikacji zwykle będą mogły uzyskać dostęp do powłoki poleceń na zdalnym serwerze. Stamtąd atakujący zwykle może znaleźć kod źródłowy programów serwera aplikacji, takich jak skrypty Perla, a nawet kod C, które są używane do uzyskania dostępu do bazy danych. Ponieważ programy te potrzebują dostępu do bazy danych, muszą również znać hasła używane do uzyskiwania dostępu do różnych partycji danych. Jeśli programiści nierozsądnie zakodowali hasła, wystarczy po prostu przejrzeć kod źródłowy, aby je odkryć. Posiadając hasła,

atakujący może użyć programu serwera aplikacji za pośrednictwem interfejsu internetowego, aby uzyskać nieautoryzowany dostęp do bazy danych. Bardziej bezpośrednio, za pomocą hasła możliwe jest zapytanie do bazy danych z poziomu powłoki poleceń, używając poleceń SQL lub poleceń z wybranego języka bazy danych. Wreszcie, podobnie jak inne złożone programy prowadzące e-biznes, bazy danych muszą być bezpiecznie skonfigurowane. Podstawowe kroki, które musi wykonać administrator bazy danych (DBA), obejmują:

* Wymuszanie uwierzytelnienia klienta WWW w bazie danych

* Wymuszanie autoryzacji klienta WWW w celu uzyskania dostępu do rekordów bazy danych

* Eliminowanie domyślnych kont bazy danych i platform bazy danych

* Zapewnienie odczytu haseł z zaszyfrowanych plików, które nie są przechowywane w kodzie programu

* Zmiana łatwych do odgadnięcia haseł

* Konfigurowanie i utrzymywanie wewnętrznej kontroli dostępu

* Kontrolowanie plików dziennika pod kątem podejrzanej aktywności

Utrzymanie konta może być kluczową luką w zarządzaniu bazą danych. Często dostawca oprogramowania bazy danych tworzy konto DBA z łatwo odgadywalnym hasłem. Co gorsza, DBA mogą używać domyślnego konta i hasła dystrybuowanego podczas instalacji bazy danych. Pozwala to osobie atakującej, która ma wiedzę na temat domyślnych haseł, uzyskać dostęp do wszystkich części bazy danych, przyjmując tożsamość administratora bazy danych.

Bezpieczeństwo platformy.

Jeden z obszarów, o którym wspomniano wcześniej, dotyczy platform hostujących komponenty e-biznesu. Platforma lub system operacyjny stanowi podstawę e-biznesu, ale jest potencjalnie słabym ogniwem w zakresie bezpieczeństwa. Jeśli w fundamencie występują pęknięcia, niewiele może zrobić nawet silne oprogramowanie aplikacyjne, aby zabezpieczyć firmę. Dlatego konieczne jest, aby administratorzy systemu odpowiednio poprawiali luki w zabezpieczeniach platformy i utrzymywali bezpieczeństwo samej platformy. Jak wspomniano wcześniej, zapory ogniowe mogą przejść długą drogę w kierunku blokowania dostępu do luk w zabezpieczeniach platformy przez nieautoryzowane osoby z zewnątrz. Jednak autoryzowani, ale nieuprzywilejowani użytkownicy zapory mogą wykorzystać znane luki w zabezpieczeniach platformy, aby uzyskać uprawnienia administratora na maszynie o znaczeniu krytycznym dla firmy. Osoby postronne, które są w stanie uzyskać uprawnienia użytkownika na platformie za pomocą dowolnych środków, mogą również przenikać dziury w platformie w przypadku poważnych naruszeń bezpieczeństwa. Niektóre kluczowe kroki niezbędne do utrzymania bezpieczeństwa platformy obejmują:

* Eliminacja domyślnych kont zazwyczaj instalowanych w systemie operacyjnym

* Zakaz łatwych do odgadnięcia haseł

* Wymuszanie wygaśnięcia hasła

* Dezaktywowanie niepotrzebnych usług, które mogą być domyślnie uruchomione

* Regularne stosowanie poprawek bezpieczeństwa w systemie operacyjnym

* Aktualizacja systemu operacyjnego do najnowszej wersji

* Zapewnienie, że uprawnienia dostępu do plików są odpowiednio egzekwowane, aby zapobiec niepotrzebnemu dostępowi do krytycznych plików

* Włączanie rejestrowania audytu z monitorowaniem włamań

* Regularne sprawdzanie integralności plików systemowych

Niektórzy administratorzy uważają, że wdrożenie zapory ogniowej jest akceptowalnym zamiennikiem bezpiecznego konfigurowania ich platform. Co więcej, z powodu mnóstwa różnych platform działających w ich przedsiębiorstwach, wielu administratorów systemu rezygnuje z instalowania najnowszych poprawek systemu operacyjnego i bezpiecznego konfigurowania swoich platform. Błędnie zakładają, że zapory ogniowe ochronią je przed wszystkimi zagrożeniami, nawet bez konserwacji. Niestety, rozluźnienie bezpieczeństwa hosta może ułatwić pracę hakerów. Skacząc z maszyny na maszynę, zwykle mogą znaleźć cenne informacje, których szukają, lub jeśli to jest ich cel, mogą siać maksymalne obrażenia. Coraz ważniejszą platformą do rozważań na temat bezpieczeństwa witryny są smartfony. Do końca 2012 r. Na świecie było 2,1 miliarda mobilnych użytkowników Internetu na koniec 2012 r. Według szacunków ITU (czerwiec 2012 r.) Na świecie było 2,1 miliarda aktywnych abonamentów mobilnych. To 29,5 procent globalnej populacji.

* W ciągu ostatnich trzech lat subskrypcje mobilnego Internetu szerokopasmowego rosły o 40 procent rocznie.

* Liczba subskrypcji mobilnego Internetu szerokopasmowego jest większa niż liczba stałych abonamentów szerokopasmowych 3: 1.

* W krajach rozwiniętych użytkownicy mobilnego Internetu szerokopasmowego często mają również dostęp do stacjonarnego połączenia szerokopasmowego, ale w krajach rozwijających się mobilny Internet szerokopasmowy jest często jedyną dostępną metodą dostępu dla ludzi.

* Ericsson (listopad 2012 r.) Prognozuje, że globalne subskrypcje mobilnego Internetu szerokopasmowego osiągną 1,5 mld na koniec 2012 r. I 6,5 mld w 2018 r. telefon komórkowy pozostanie dominującym urządzeniem mobilnego dostępu szerokopasmowego.

Tablety mają również własne wersje przeglądarek. Projektanci stron internetowych coraz częściej muszą radzić sobie ze znacznymi różnicami między platformami uzyskującymi dostęp do ich kodu. Craig Smith zamieścił następujące punkty w artykule ze stycznia 2013 r. Zatytułowanym „Optymalizacja e-commerce na tablety i smartfony”:

* Poprawa nawigacji i użyteczności: Użytkownicy powinni mieć możliwość łatwego i prawidłowego wybierania opcji; unikaj rozwijanych menu i przycisków, które są blisko siebie.

* Wykorzystanie responsywnego projektu: Określ, jak zaprezentować stronę internetową w zależności od tego, jakie urządzenie ma do niej dostęp.

* Określenie celu dostępu: Rozróżnij typy zapytań, które są najczęstsze i różne na różnych platformach. Zoptymalizuj projekt pod kątem najpopularniejszych typów dla każdej platformy.

Kolejne pytanie dotyczy tego, czy witryny handlu elektronicznego powinny zależeć od dostępu do sieci, czy też opracowywać aplikacje (aplikacje) specjalnie dla systemów operacyjnych na smartfony i tablety. Główną zaletą aplikacji jest to, że można je zaprogramować w taki sposób, aby uniknąć luk w zabezpieczeniach typowych dla urządzeń mobilnych w pierwszych latach powszechnego wdrażania.

PODSUMOWANIE.

W tej sekcji omówiono najsłabsze linki , aplikacje internetowe, w tym klientów sieci Web, protokoły sieciowe, zewnętrzne serwery sieciowe, wewnętrzne bazy danych, serwery aplikacji i platformy, na których działają. Bezpieczne protokoły sieciowe są konieczne, ale z pewnością niewystarczające do zabezpieczenia handlu elektronicznego. Opisane tu podatności są w dużej mierze oparte na wadach oprogramowania istniejących w warstwie aplikacji transakcji e-commerce. Być może najczęstszą podatnością w systemach e-commerce jest błędna konfiguracja oprogramowania. Ponieważ odpowiedzialność za konfigurację oprogramowania spoczywa na użytkowniku, należy wdrożyć i egzekwować politykę bezpieczeństwa. Po skonfigurowaniu systemu ważne jest, aby poddać go weryfikacji i testom stron trzecich. Audyt strony trzeciej może zapewnić, że skonfigurowany system, w tym routery, zapory ogniowe, serwery i bazy danych, spełnia specyfikacje polityki bezpieczeństwa. System powinien być również okresowo testowany pod kątem znanych, powszechnych ataków, a także nowych zagrożeń, gdy się pojawią. Jak każdy system oprogramowania, komercyjne gotowe oprogramowanie ma wady, z których wiele ma kluczowe znaczenie dla bezpieczeństwa. Konieczne jest, aby administratorzy systemu byli na bieżąco ze stronami internetowymi dostawców, witrynami hakerów, grupami dyskusyjnymi oraz wszędzie tam, gdzie uwalniane są zagrożenia i łatki do ich oprogramowania. Zarówno środowiska bezpieczeństwa, jak i hakerów nieustannie pracują nad znalezieniem wad w COTS, które producenci oprogramowania i firmy zajmujące się bezpieczeństwem zazwyczaj szybko naprawiają. Konsumenci oprogramowania - ci, którzy kupują i używają oprogramowania - muszą wykonać swoją część, stosując wszystkie odpowiednie łaty w swoim wrażliwym oprogramowaniu. W przypadku niestandardowego oprogramowania, które obejmuje frontonowe strony internetowe, serwery aplikacji, skrypty CGI i treści mobilne, takie jak ActiveX lub Java, programiści muszą zrobić wszystko, co możliwe, aby zapewnić, że ich oprogramowanie nie jest podatne na ataki i że nie narusza w sprawie prywatności użytkowników.