HACKOWANIE SERWERÓW WWW

Internet zrewolucjonizował handel i komunikację. Dostarczanie i kupowanie towarów i usług online jest określane jako e-commerce. Lockdowny spowodowane pandemią COVID-19 przyspieszyły e-commerce, ponieważ firmy i konsumenci przeszli na cyfryzację. Udział e-commerce w globalnym handlu detalicznym wzrósł z 14% w 2019 r. do około 17% w 2020 r. Komunikacja biznesowa również ucierpiała z powodu COVID-19, ponieważ liczba pracy zdalnej i konferencji zdalnych wzrosła drastycznie. Zarówno e-commerce, jak i praca zdalna wymagają serwerów WWW. Wraz ze wzrostem liczby serwerów WWW i ich wykorzystania, wzrosło również hakowanie tych serwerów WWW. Rozwiązania e-commerce i pracy zdalnej wykorzystują również aplikacje WWW i usługi WWW. Tworzenie aplikacji WWW wiąże się z wykorzystaniem platform programistycznych, takich jak Microsoft Active Server Pages (ASP i ASP.NET) oraz Java Server Pages (JSP). Zwykle aplikacja WWW jest obsługiwana przez serwer WWW, który działa na systemie operacyjnym ogólnego przeznaczenia lub osadzonym. Każdy komponent (aplikacja, serwer i system operacyjny) ma własny zestaw luk, ale łączenie tych komponentów zwiększa ryzyko naruszenia bezpieczeństwa aplikacji internetowej, co wpłynie na ogólne bezpieczeństwo sieci. Doświadczeni hakerzy mogą często wykorzystać niewielką lukę w jednej funkcji, takiej jak aplikacja poczty internetowej, i użyć jej jako trampoliny do przeprowadzenia dodatkowych ataków na system operacyjny. Wraz ze wzrostem liczby dostępnych platform i witryn e-commerce, luki w zabezpieczeniach rozprzestrzeniły się. Statystycznie większość cyberataków koncentruje się na serwerach internetowych, aplikacjach internetowych i ich infrastrukturze baz danych. Ten moduł daje przegląd aplikacji internetowych.

ZROZUMIENIE APLIKACJI SIECIOWYCH

Jak się dowiedziałeś, napisanie programu bez błędów jest trudne. Im większy program, tym więcej błędów lub defektów jest możliwych, a niektóre defekty tworzą luki w zabezpieczeniach. Im więcej osób ma dostęp do programu, tym większe ryzyko luk w zabezpieczeniach. W poniższych sekcjach opisano komponenty aplikacji sieciowych i platformy do tworzenia aplikacji sieciowych

Komponenty aplikacji internetowych

HTML jest nadal podstawą większości aplikacji internetowych i jest powszechnie używany do tworzenia statycznych stron internetowych. HTML5 to najnowsza wersja w rodzinie HTML. Statyczne strony internetowe wyświetlają te same informacje niezależnie od pory dnia lub użytkownika, który uzyskuje dostęp do strony. Dynamiczne strony internetowe mogą zmieniać wyświetlane informacje w zależności od zmiennych, takich jak bieżąca godzina i data, nazwa użytkownika i historia zakupów (informacje zbierane za pomocą plików cookie lub błędów internetowych). Aby strony internetowe były dynamiczne, ich kod musi składać się z czegoś więcej niż tylko podstawowych znaczników HTML, takich jak <body> i <h1>. Te strony internetowe wymagają specjalnych komponentów do wyświetlania informacji, które zmieniają się w zależności od danych wprowadzonych przez użytkownika lub informacji z serwera zaplecza. Aby to zrobić, dynamiczne strony internetowe wykorzystują różnorodne narzędzia, w tym element <form>, Asynchronous JavaScript and XML (AJAX), Common Gateway Interface (CGI), Active Server Pages (ASP.NET), Java Server Pages (JSP), Hypertext Preprocessor (PHP), ColdFusion (CF), JavaScript (JS) i ciągi łączników baz danych, takie jak Open Database Connector (ODBC). Te komponenty są omówione w poniższych sekcjach.

Formularze internetowe

Element <form> jest używany w dokumencie HTML, aby umożliwić klientom przesyłanie informacji do serwera internetowego. Prawdopodobnie wypełniłeś formularz podczas zakupu produktu online lub rejestrowania się na newsletter e-mailowy, na przykład. Niektóre formularze mogą być dość długie i

wymagać podania wielu informacji, a niektóre mają tylko kilka pól wejściowych, takich jak nazwa użytkownika i hasło. Serwer internetowy używa aplikacji internetowej do przetwarzania informacji z formularza. Poniższy kod HTML pokazuje składnię prostego formularza, a Rysunek pokazuje stronę internetową utworzoną za pomocą tego kodu.

<html>

<body>

<form>

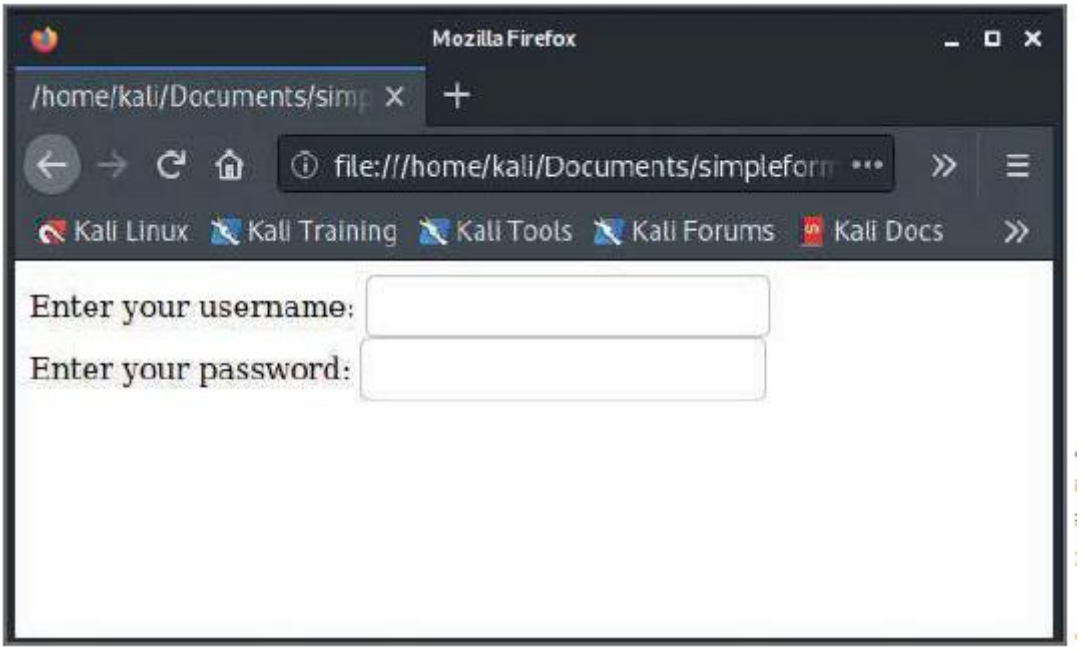
Enter your username:

<input type="text" name="username">

Enter your password:

<input name="password" type="password">

</form> </body></html>



Common Gateway Interface

Innym standardem, który obsługuje przenoszenie danych z serwera WWW do przeglądarki WWW, jest Common Gateway Interface (CGI), który umożliwia projektantom stron WWW tworzenie dynamicznych aplikacji internetowych HTML. Wiele dynamicznych stron WWW jest tworzonych przy użyciu CGI i języków skryptowych. CGI to interfejs, który określa sposób, w jaki serwer WWW przekazuje dane do przeglądarki WWW. Opiera się na Perlu lub innym języku skryptowym lub programowania, aby tworzyć dynamiczne strony WWW, co różni się od stron Active Server Pages (omówionych w dalszej części). Główną rolą CGI jest przekazywanie danych między serwerem WWW a przeglądarką WWW. W rzeczywistości termin „brama” opisuje ten ruch danych między serwerem WWW a przeglądarką WWW. Programy CGI można pisać w wielu językach programowania i skryptowych, takich jak C/C++, Perl, powłoki UNIX, Visual Basic i Java. Języki programowania, takie jak

C i C++, wymagają skompilowania programu przed jego uruchomieniem. Jeśli CGI jest implementowane przy użyciu języka skryptowego, kompilacja nie jest konieczna. Poniższy program CGI wyświetla „Hello Security Testers!” w przeglądarce użytkownika. Ten program hello.pl jest napisany w Perlu i zostanie umieszczony w katalogu cgi-bin na serwerze internetowym:

```
#!/usr/bin/perl

print "Content-type: text/html\n\n";

print "Hello Security Testers!";
```

Aby sprawdzić, czy program CGI działa, zapisz program w katalogu cgi-bin swojego serwera internetowego, a następnie wprowadź adres URL w przeglądarce internetowej, używając formatu http://www.myweb.com/cgi-bin/hello.pl. (Zamień nazwę swojego serwera i prawidłową ścieżkę do katalogu cgi-bin.)

Struktury i biblioteki innych firm

Spring, JSF, AngularJS, Yeoman, Sass i Vaadin to tylko kilka z setek struktur zaprojektowanych w celu ułatwienia programowania. Struktury są zazwyczaj wywoływane w określonym celu. Na przykład siłą struktury Spring jest łączenie komponentów, podczas gdy struktura Sass pomaga stylizować witrynę w celu zwiększenia komfortu użytkownika. Korzystanie z bibliotek oszczędza czas programistom i oznacza, że potrzeba mniej dokumentacji do złożonych procedur niestandardowego kodu. Wraz ze wzrostem popularności bibliotek innych firm, dbanie o ich aktualność i bezpieczeństwo staje się coraz ważniejsze.

Active Server Pages

Active Server Pages (ASP i ASP.NET) to dwie inne technologie, których programiści mogą używać do wyświetlania użytkownikom dokumentów HTML w locie. Oryginalny ASP został wprowadzony przez Microsoft w 1998 roku jako język skryptowy interpretowany po stronie serwera i jest często nazywany klasycznym ASP. Klasyczny ASP jest uważany za martwy język i nie jest już używany przez główny nurt społeczności programistów. Klasyczny ASP został zastąpiony przez ASP.NET, który używa skompilowanego języka po stronie serwera (takiego jak C#) i .NET Framework. ASP i ASP.NET służą do tworzenia dynamicznych stron internetowych. Oznacza to, że gdy użytkownik żąda strony internetowej, jest ona tworzona w tym momencie. ASP umożliwił programistom tworzenie dynamicznych, interaktywnych stron internetowych przy użyciu języków skryptowych, takich jak JScript (wersja JavaScript firmy Microsoft) lub VBScript. Programiści mogą również używać ASP. NET do tworzenia dynamicznych, interaktywnych stron internetowych, ale jak wspomniano wcześniej, jest kompilowany i powszechnie pisany w języku C#. Nie wszystkie serwery internetowe obsługują ASP lub ASP.NET, więc jeśli chcesz tworzyć strony internetowe z jednym z nich, używany serwer musi obsługiwać tę technologię. Internet Information Services (IIS) 4.0 i nowsze obsługują ASP, a IIS 5.0 i nowsze obsługują ASP.NET. Pamiętaj, że serwer internetowy, a nie przeglądarka internetowa, musi obsługiwać ASP. W ćwiczeniach pracujesz z IIS, aby lepiej zrozumieć aplikację internetową

Instalowanie usług Internet Information Services

Czas trwania: 30 minut

Cel: Zainstaluj IIS na komputerze z systemem Windows.

Opis: Aby hostować witrynę internetową, musisz zainstalować IIS na komputerze z systemem Windows. Chociaż IIS jest wdrażany na serwerze w środowisku produkcyjnym, przedprodukcyjne

tworzenie i testowanie stron internetowych można wykonywać na stacjach roboczych. IIS 10 jest dostępny w systemie Windows 10. Ponieważ IIS nie jest instalowany domyślnie, w tej aktywności instalujesz go i używasz przeglądarki internetowej, aby sprawdzić, czy został zainstalowany poprawnie. Ta aktywność zakłada, że nigdy nie instalowałeś IIS na używanym komputerze i instalujesz na komputerze z systemem Windows 10.

1. Otwórz Panel sterowania, a następnie kliknij Programy.

2. W sekcji Programy i funkcje kliknij Włącz lub wyłącz funkcje systemu Windows, aby otworzyć okno dialogowe Funkcje systemu Windows. Jeśli otworzy się okno komunikatu Kontrola konta użytkownika (UAC), kliknij Kontynuuj. Kliknij pole wyboru Usługi Internet Information Services, a następnie kliknij symbol plusa, aby rozwinąć opcje IIS. Kliknij pole wyboru Usługi World Wide Web. Upewnij się, że pola Funkcje tworzenia aplikacji, Wspólne funkcje HTTP i Zabezpieczenia są zaznaczone w obszarze Usługi World Wide Web. W obszarze Funkcje tworzenia aplikacji upewnij się, że wszystkie opcje ASP i ASP.NET są zaznaczone. Kliknij Narzędzia zarządzania siecią i upewnij się, że Konsola zarządzania usługami IIS oraz Skrypty i narzędzia usług IIS są zaznaczone w obszarze Narzędzia zarządzania siecią. Po zakończeniu okno powinno wyglądać tak, jak pokazano na rysunku 10-2. Nie odznaczaj żadnych opcji, które są już zaznaczone.

3. Kliknij OK, aby zainstalować usługi IIS. Gdy zobaczysz komunikat „System Windows zakończył żądane zmiany", kliknij Zamknij.

4. Aby sprawdzić, czy usługi IIS zostały zainstalowane, kliknij przycisk Start, wpisz inetmgr i naciśnij klawisz Enter, aby otworzyć Menedżera usług IIS (patrz rysunek 10-3). Kliknij Pomoc na pasku menu, a następnie kliknij Informacje o usługach Internet Information Services. Jaka wersja usług IIS jest zainstalowana na Twoim komputerze?

5. Uruchom przeglądarkę internetową, wpisz adres URL http://localhost, a następnie naciśnij klawisz Enter, aby przejść na stronę powitalną usług IIS. Odwiedzający Twoją witrynę zobaczy wiadomość powitalną IIS, ponieważ nie utworzyłeś jeszcze domyślnej strony HTML. Jednak kliknięcie grafiki przekieruje Cię do oficjalnej witryny Microsoft IIS, gdzie możesz dowiedzieć się więcej o IIS.

Po zakończeniu zamknij okno przeglądarki.

6. Następnie musisz utworzyć folder na swoim serwerze internetowym, aby przechowywać wszystkie utworzone przez Ciebie strony HTML. Po zainstalowaniu IIS na dysku C zostanie utworzony nowy folder o nazwie inetpub. Otwórz Eksplorator plików. Na dysku C (jeśli Twoja instalacja jest inna, zamień literę dysku na właściwą), kliknij, aby rozwinąć folder inetpub, a następnie otwórz folder wwwroot.

7. Kliknij prawym przyciskiem myszy folder wwwroot, wskaż polecenie Nowy, a następnie kliknij polecenie Folder. Jako nazwę folderu wpisz YourFirstName (zastępując swoje imię), a następnie naciśnij klawisz Enter.

8. Zamknij wszystkie otwarte okna i pozostaw system Windows uruchomiony do następnej czynności.

Aby uniemożliwić atakującym poznanie struktury katalogów, którą tworzysz na serwerze internetowym IIS, zaleca się utworzenie katalogu wirtualnego, aby ścieżka, którą użytkownik widzi w przeglądarce internetowej, nie była rzeczywistą ścieżką na serwerze internetowym. Katalog wirtualny jest wskaźnikiem do katalogu fizycznego. Na przykład w przypadku katalogów wirtualnych użytkownik może zobaczyć https : // www . mycompany . com/jobs/default.aspx zamiast https : // www . mycompany . com/security/positions/CEH_Cert/ default.aspx. Prostsza struktura oferowana przez katalog wirtualny jest często łatwiejsza do zapamiętania i nawigacji dla użytkowników. Korzystanie z

tej strategii projektowania zwiększa również bezpieczeństwo, ponieważ pomaga ukryć rzeczywistą strukturę katalogów przed atakującymi.

Tworzenie katalogu wirtualnego

Czas trwania: 15 minut

Cel: Dowiedz się, jak utworzyć katalog wirtualny na serwerze internetowym IIS.

Opis: Po zainstalowaniu IIS i utworzeniu katalogów fizycznych administrator witryny powinien utworzyć katalogi wirtualne, które uniemożliwią odwiedzającym witrynę zobaczenie struktury katalogów fizycznych. W tej aktywności utworzysz katalog wirtualny, używając katalogu utworzonego w Aktywności 10-1.

1. Kliknij przycisk Start, wpisz inetmgr i naciśnij klawisz Enter. W oknie Menedżera usług IIS kliknij, aby rozwinąć nazwę komputera, Witryny i Domyślną witrynę internetową (patrz Rysunek 10-4).

2. Kliknij prawym przyciskiem myszy folder YourFirstName utworzony w Aktywności 10-1, a następnie kliknij Dodaj katalog wirtualny.

3. W polu tekstowym Alias wpisz swoje imię. Wpisz (lub przejdź do) ścieżkę fizyczną folderu utworzonego w Ćwiczeniu 10-1 (C:\inetpub\wwwroot\YourFirstName), a następnie kliknij przycisk OK, aby utworzyć katalog wirtualny, do którego użytkownicy będą mogli uzyskać dostęp przez Internet.

4. Zamknij wszystkie otwarte okna i pozostaw system Windows uruchomiony do następnej aktywności.

Wcześniej w tym kursie napisałeś stronę internetową HTML. Wkrótce przyjrzysz się stronie internetowej zawierającej instrukcje ASP.NET. Najlepszym sposobem nauki ASP.NET jest utworzenie strony internetowej z jego użyciem. Aby to zrobić, potrzebujesz trzech komponentów: edytora tekstu (np. Notatnika), serwera internetowego (takiego jak serwer internetowy IIS) i przeglądarki internetowej (takiej jak Chrome, Microsoft Edge lub Firefox).

Tworzenie strony internetowej ASP.NET

Czas trwania: 20 minut

Cel: Rozpoznawanie stron internetowych ASP.NET i używanie ASP.NET do tworzenia dynamicznych stron internetowych.

Opis: Strony internetowe ASP.NET są tworzone na serwerze internetowym i umożliwiają deweloperom tworzenie dynamicznych stron internetowych. W tej aktywności utworzysz stronę internetową ASP.NET i użyjesz przeglądarki internetowej, aby ją wyświetlić.

1. Aby uruchomić Notatnik z uprawnieniami administratora, kliknij przycisk Start, wpisz Notatnik, kliknij prawym przyciskiem myszy Notatnik, a następnie kliknij polecenie Uruchom jako administrator. (W razie potrzeby kliknij Tak w polu komunikatu UAC). W Notatniku wpisz następujący kod:

```
<html>

<head><title>Moja pierwsza strona internetowa ASP.NET</title></head>

<body>

<h1>Witajcie, specjaliści ds. bezpieczeństwa</h1>
```

Data i godzina to <%=DateTime.Now %>.

</body>

</html>

2. Zapisz plik jako First.aspx w folderze C:\inetpub\wwwroot\YourFirstName. Upewnij się, że plik jest zapisany z rozszerzeniem .aspx, a nie .txt. Zamknij Notatnik.

3. Aby przetestować stronę internetową First.aspx, uruchom przeglądarkę internetową, wpisz http : // localhost /YourFirstName/First.aspx, a następnie naciśnij Enter. Strona internetowa pokazuje bieżącą datę i godzinę Twojej lokalizacji, co oznacza, że jest dynamiczna. Oznacza to, że zmienia się za każdym razem, gdy przeglądarka internetowa wywołuje stronę internetową. Znaczniki <% i %> informują serwer internetowy o wykonaniu kodu między znacznikami i wyrenderowaniu wyniku podczas ładowania strony.

4. Kliknij prawym przyciskiem myszy stronę internetową First.aspx w przeglądarce internetowej i wybierz opcję Wyświetl źródło strony. Czy kod źródłowy pokazuje wprowadzone polecenia ASP.NET?

5. Zamknij przeglądarkę internetową i wyloguj się z systemu Windows w celu wykonania następnej czynności.

Apache Web Server

Jako tester bezpieczeństwa powinieneś znać Apache, inny program serwera WWW. W 2021 r. Apache Web Server miał 31,7% udziału w rynku serwerów WWW w porównaniu do 6,7% dla IIS. Znajomość Apache może być pomocna w zawodzie testera bezpieczeństwa. Apache ma ważne zalety w porównaniu z konkurencją: działa na niemal każdej platformie *nix, a także w systemie Windows i jest bezpłatny. Instalacja Apache w systemie Linux różni się od instalacji IIS w systemie Windows, ale nie musisz się martwić instalacją, ponieważ demon serwera WWW Apache (httpd) jest instalowany domyślnie w systemie Kali Linux. Nginx to kolejny darmowy program serwera WWW typu open source i ma mniej więcej taki sam udział w rynku jak Apache. Nginx jest również instalowany domyślnie w systemie Kali Linux. W Ćwiczeniu 10-4 zapoznasz się z serwerem WWW Apache.

Praca z serwerem Apache Web Server

Czas trwania: 35 minut

Cel: Poznanie podstawowych ustawień i zadań w serwerze Apache Web Server.

Opis: Bez wątpienia natkniesz się na systemy Apache Web Server podczas przeprowadzania testu bezpieczeństwa. Ponieważ Apache jest zaawansowanym, modułowym serwerem internetowym, opanowanie jego funkcji i opcji może zająć sporo czasu. Układ Apache różni się w zależności od systemu operacyjnego. Na przykład Apache w systemie Fedora Linux różni się od Apache w systemie Ubuntu Linux. W tej aktywności poznasz podstawowe polecenia serwera Apache Web Server i dowiesz się, jak znaleźć i zmodyfikować niektóre opcje konfiguracji (nazywane „dyrektywami Apache”). Celem tej aktywności jest skonfigurowanie serwera internetowego z katalogiem wymagającym uwierzytelnienia.

1. Uruchom system Kali Linux. W poniższych krokach wpisz polecenie dokładnie tak, jak pokazano, ponieważ Linux rozróżnia wielkość liter.

2. Otwórz powłokę terminala. W wierszu poleceń wpisz sudo systemctl start apache2 i naciśnij Enter. Jeśli usługa apache2 nie uruchamia się, może to być spowodowane tym, że serwer WWW nginx jest już uruchomiony. Zatrzymaj nginx, wprowadzając polecenie sudo systemctl stop nginx, a następnie

wprowadź polecenie sudo systemctl start apache2. Potwierdź, że usługa apache2 jest uruchomiona, używając polecenia systemctl status apache2, które powinno wyświetlić status active (uruchomiony).

3. Uruchom przeglądarkę internetową. W pasku adresu wpisz localhost i naciśnij Enter. Witryna wyświetli instrukcje dotyczące manipulowania domyślną konfiguracją apache. Przeczytaj tę stronę.

4. Otwórz powłokę terminala. W wierszu polecenia wpisz sudo systemctl stop apache2 i naciśnij Enter.

5. Aby wyświetlić domyślne pliki konfiguracji apache, najpierw wpisz cd /etc/apache2 i naciśnij Enter w powłoce terminala, aby zmienić katalogi. Wpisz grep Include apache2.conf i naciśnij Enter, aby wyświetlić listę plików i katalogów, w których serwer Apache szuka dodatkowych dyrektyw podczas uruchamiania (patrz Rysunek 10-5). Zwróć uwagę na przedostatni wiersz, IncludeOptional sites-enabled/*.conf. W tym katalogu Apache sprawdza pliki konfiguracji witryny. Możesz dodać witrynę, dodając jej plik konfiguracji do tego katalogu bez konieczności zmiany głównego pliku konfiguracji apache2.conf.

6. Wpisz cd /etc/apache2/sites-enabled && ls i naciśnij Enter.

7. Otwórz plik w edytorze gedit, wpisując sudo gedit 000- default.conf i naciskając Enter.

8. Wprowadź następujące wiersze na końcu pliku, poniżej wiersza </VirtualHost>:

<Directory /var/www/html/restricted>

Options Indexes FollowSymLinks

AllowOverride AuthConfig

Order allow,deny

allow from all

</Directory>

9. Zapisz zmiany i wyjdź z edytora gedit.

10. W powłoce terminala utwórz nowy katalog, wpisując sudo mkdir /var/www/html/restricted i naciskając Enter.

11. Wpisz cd /var/www/html/restricted i naciśnij Enter, aby przejść do katalogu utworzonego w kroku 10. Wpisz sudo touch secret.txt i naciśnij Enter, aby utworzyć plik tekstowy o nazwie secret.txt w tym katalogu.

12. Następnie utwórz plik .htaccess w tym samym katalogu. Ten plik jest plikiem konfiguracji katalogu lokalnego określonym w apache2.conf przez dyrektywę AccessFileName. Jeśli plik .htaccess znajduje się w dowolnym katalogu witryny, Apache najpierw go sprawdza. W tym pliku .htaccess wskazujesz Apache lokalizację pliku AuthUserFile (w zasadzie pliku hasła). Wpisz sudo vi .htaccess i naciśnij Enter. W tym przypadku musisz użyć vi zamiast gedit, ponieważ gedit nie umieści prawidłowych znaków Unix (LF) na końcu każdego wiersza, a Apache nie odczyta pliku prawidłowo. Po wejściu do vi wpisz i, aby rozpocząć wstawianie następującego tekstu:

AuthType Basic

AuthName „Password Required”

AuthUserFile /etc/apache2/.htpasswd

Require user tester

13. Zapisz zmiany i wyjdź z edytora, naciskając Esc, a następnie wpisując :wq! i naciskając Enter. W powłoce terminala utwórz plik hasła, wpisując sudo htpasswd -c /etc/apache2/.htpasswd tester i naciskając Enter. Gdy pojawi się monit, wprowadź hasło, potwierdź je, a następnie zanotuj je. Plik .htaccess utworzony w kroku 12 nakazuje Apache'owi sprawdzenie w pliku .htpasswd hasła użytkownika testera. Możesz uruchomić polecenie cat /etc/apache2/.htpasswd, aby wyświetlić skrót hasła dla nowego użytkownika.

14. Uruchom ponownie Apache, wpisując sudo systemctl restart apache2 i naciskając Enter. W przeglądarce internetowej Kali przejdź do http://localhost/restricted, a następnie wprowadź nazwę użytkownika tester i hasło utworzone w kroku 13. Jaki plik jest wyświetlany? Jeśli chcesz ponownie zostać poproszony o podanie hasła, musisz zamknąć i ponownie otworzyć przeglądarkę.

15. Sprawdź, czy inne komputery w Twojej sieci mogą uzyskać dostęp do Twojego folderu z ograniczeniami, wpisując http : // yourIPaddress/ restricted w ich przeglądarkach (zastępując yourlPaddress swoim adresem IP). W razie potrzeby wpisz ifconfig eth0 i naciśnij Enter, aby znaleźć swój adres IP.

16. Dlaczego wprowadzanie danych uwierzytelniających na stronie internetowej niezabezpieczonej protokołem SSL, takiej jak ta strona, jest problemem? Jakie jest rozwiązanie tego problemu?

17. Przeprowadź badania „Podstawowe słabości zabezpieczeń uwierzytelniania” w Internecie. Czy występują jakieś problemy z podstawowym uwierzytelnianiem?

18. Zamknij powłokę terminala, wyjdź z przeglądarki internetowej Kali i wyloguj się z Kali Linux.

Korzystanie z języków skryptowych

Strony internetowe można tworzyć za pomocą kilku języków skryptowych, takich jak VBScript i JavaScript. Nie nauczysz się, jak zostać programistą internetowym, przeglądając języki skryptowe omówione w tym module, ale powinieneś być w stanie rozpoznać, kiedy jeden z nich jest używany, ponieważ wiele narzędzi do testowania bezpieczeństwa jest napisanych w językach skryptowych. Większość wirusów makro i wszystkie robaki, które wykorzystują luki w zabezpieczeniach cross-site scripting (omówione później w module), są oparte na języku skryptowym.

Procesor hipertekstowy PHP

Podobnie jak ASP i ASP.NET, procesor hipertekstowy PHP (PHP) umożliwia programistom internetowym tworzenie dynamicznych stron internetowych. PHP, język skryptowy typu open source po stronie serwera, jest osadzony w stronie internetowej HTML za pomocą znaczników PHP <?php i ?>. Ponieważ strony internetowe PHP działają na serwerze, użytkownicy nie mogą przeglądać kodu źródłowego w swoich przeglądarkach internetowych. PHP był pierwotnie używany głównie w systemach UNIX, ale obecnie jest szerzej używany na wielu platformach, w tym Macintosh i Windows. Poniższy fragment to przykład kodu dla statycznej strony internetowej PHP pokazujący użycie znaczników PHP:

UWAGA

Pogrubione linie w tych przykładach kodu pokazują, jak są wskazywane różne języki skryptowe.

<html>

<head>

<title>My First PHP Program</title>

</head>

<body>

<?php echo "<h1>Hello, Security Testers!</h1>"; ?>

</body>

</html>

Tę stronę trzeba by utworzyć na serwerze WWW jako plik .php, podobnie jak stronę WWW ASP.NET, którą utworzyłeś w Aktywności 10-3. Po zidentyfikowaniu, że serwer WWW używa PHP, powinieneś użyć metod poznanych na tym kursie, aby zbadać dalej konkretne luki. Na przykład kilka starszych wersji PHP działających w systemie Linux może zostać wykorzystanych z powodu wiersza w pliku Php.ini: Wiersz file_uploads=on zezwala na przesyłanie plików; jednak to ustawienie może pozwolić zdalnemu atakującemu na uruchomienie dowolnego kodu z podwyższonymi uprawnieniami. Najlepszym rozwiązaniem jest uaktualnienie do najnowszej wersji PHP, ale jeśli nie jest to możliwe, zmień wiersz na file_uploads=off.

UWAGA: Powinieneś również znać LAMP (skrót od Linux, Apache, MySQL i PHP), ponieważ jest to zbiór oprogramowania typu open source używanego w wielu zaawansowanych aplikacjach internetowych o dużym natężeniu ruchu. LAMP jest znany jako stos rozwiązań, ponieważ łączy kilka programów w jedno zintegrowane rozwiązanie aplikacji internetowej. Aby uzyskać więcej informacji, wyszukaj w Internecie termin „LAMP" w połączeniu z wersją systemu Linux, której używasz, taką jak Ubuntu lub Fedora. Wersja tego stosu dla systemu Windows nazywa się WAMP.

ColdFusion

ColdFusion to kolejny język skryptowy po stronie serwera do tworzenia dynamicznych stron internetowych. Stworzony przez Allaire Corporation, obecnie jest własnością Adobe Systems, Inc. ColdFusion integruje technologie przeglądarki internetowej, serwera internetowego i bazy danych. Używa zastrzeżonych tagów napisanych w ColdFusion Markup Language (CFML). Aplikacje internetowe napisane w CFML mogą zawierać inne technologie po stronie klienta, takie jak HTML i JavaScript. Poniższy kod jest przykładem HTML z tagiem CFML, który przekierowuje użytkownika do strony internetowej. Wszystkie tagi CFML zaczynają się od liter CF. Na przykład tag kolumny to <CFCOL>.

<html>

<head>

<title>Using CFML</title>

</head>

<body>

<CFLOCATION URL="www.isecom.org" ADDTOKEN="NO">

</body>

</html>

Podobnie jak w przypadku przykładu PHP, testerzy bezpieczeństwa powinni zapoznać się z lukami w zabezpieczeniach związanymi z serwerem internetowym korzystającym z ColdFusion. Szybkie

przeszukanie strony zabezpieczeń Adobe (https://helpx.adobe.com/security.html) może zawęzić czas badań, pozwalając skupić się na lukach w zabezpieczeniach, które dotyczą Twojej organizacji.

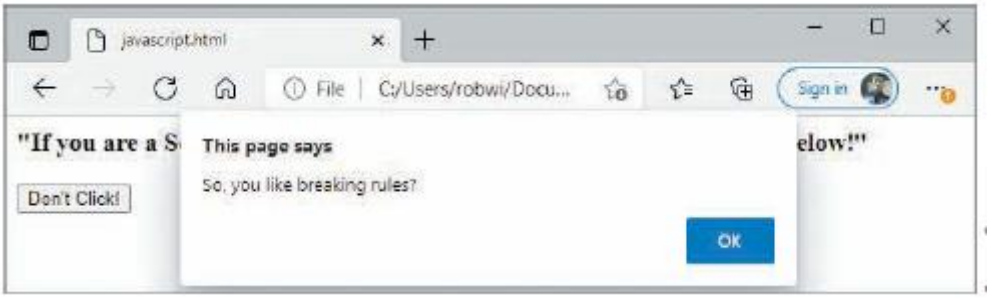
JavaScript

JavaScript, popularny język skryptowy do tworzenia dynamicznych stron internetowych, ma również moc języka programowania. W JavaScript możesz rozgałęziać, zapętlać i testować (BLT, którego nauczyłeś się wcześniej) oraz tworzyć funkcje i procedury na stronach internetowych HTML. Poniższy kod to prosty fragment kodu HTML z dodanym kodem JavaScript:

```
<html>

<head>

<script type="text/javascript">

function chastise_user()

{

alert ("So, you like breaking rules?")

document.getElementById("cmdButton").focus ()

}

</script>

</head>

<body>

<h3>"If you are a Security Tester, please do not click the command button below!"</h3>

<form>

<input type="button" value="Don't Click!" name="cmdButton"

onClick="chastise_user()" />

</form>

</body>

</html>
```

Ten kod jest bardziej złożony niż poprzednie przykłady, ale pokazuje, jak języki skryptowe mogą zawierać funkcje i alerty. Trzeci wiersz określa, że JavaScript jest używanym językiem. Następnie zdefiniowano funkcję chastise_user(); ta funkcja po prostu wyświetla komunikat alertu. Funkcja getElementById() jest metodą (sekwencją instrukcji, które wykonują rutynę lub zadanie) zdefiniowaną przez Document Object Model (DOM) World Wide Web Consortium (W3C). Zasadniczo zwraca obiekt — w tym przypadku przycisk polecenia, który klikasz. Pozostały kod jest dość oczywisty. Aby zobaczyć, jak działa ten kod, spójrz na dane wyjściowe pokazane na rysunku 10-6.

Jeśli użytkownik zaakceptuje ostrzeżenie bezpieczeństwa i kliknie przycisk polecenia, zostanie wyświetlone okno komunikatu alertu pokazane na rysunku 10-7.



JavaScript jest szeroko stosowany, a w starszych przeglądarkach internetowych wykorzystano wiele luk. Testerzy bezpieczeństwa i administratorzy powinni sprawdzać każdy komputer pod kątem niezałatanych lub nieaktualnych wersji przeglądarki i być na bieżąco z lukami. Na przykład Cyber Security Alert AA21-076A, TrickBot Malware (https://us-cert.cisa.gov/ncas/alerts/aa21-076a) opisuje trojana TrickBot. TrickBot używa złośliwego JavaScript do komunikowania się z serwerem poleceń i kontroli (C2) złośliwego aktora w celu pobrania złośliwego oprogramowania na system ofiary.

Łączenie się z bazami danych

Większość stron internetowych wyświetlających użytkownikom informacje o firmie jest przechowywana na serwerze bazy danych. Strony internetowe, które proszą użytkownika o podanie informacji, takich jak imię i nazwisko, numer telefonu, adres itp., przechowują informacje wprowadzane przez użytkowników w bazie danych. Technologia używana do łączenia aplikacji internetowej z serwerem bazy danych może się różnić w zależności od systemu operacyjnego, ale teoria jest taka sama. W poniższych sekcjach omówiono niektóre technologie używane do łączenia się z bazą danych lub zewnętrznym systemem plików z aplikacji internetowej.

Otwarta łączność z bazą danych

Otwarta łączność z bazą danych (odbc) to standardowa metoda dostępu do bazy danych opracowana przez SQL Access Group. Interfejs ODBC umożliwia aplikacji dostęp do danych przechowywanych w systemie zarządzania bazą danych (DBMS), takim jak Microsoft SQL, Oracle lub dowolnym systemie, który może rozpoznawać i wydawać polecenia ODBC. Interoperacyjność między systemami DBMS typu back-end jest kluczową cechą interfejsu ODBC, umożliwiającą programistom skupienie się na aplikacji bez martwienia się o konkretny system DBMS. Interfejs ODBC realizuje tę interoperacyjność, definiując następujące elementy:

• Znormalizowana reprezentacja typów danych

• Biblioteka wywołań funkcji ODBC, która umożliwia aplikacji łączenie się z systemem DBMS, uruchamianie instrukcji SQL i pobieranie wyników

• Standardowa metoda łączenia się i logowania do systemu DBMS

Baza danych łącząca i osadzająca obiekty

baza danych łącząca i osadzająca obiekty (ole db) to zestaw interfejsów, które umożliwiają aplikacjom dostęp do danych przechowywanych w systemie DBMS. Firma Microsoft zaprojektowała ją tak, aby była szybsza, wydajniejsza i bardziej stabilna niż jej poprzednik, ODBC. OLE DB opiera się na ciągach połączeń, które umożliwiają aplikacji dostęp do danych przechowywanych na urządzeniu zewnętrznym. W zależności od źródła danych, z którym się łączysz, możesz użyć innego dostawcy. Na przykład połączenie z bazą danych SQL wymaga użycia SQLOLEDB jako dostawcy zamiast Microsoft.ACE. Podczas przeprowadzania testu bezpieczeństwa na serwerze internetowym należy sprawdzić, w jaki sposób serwer internetowy łączy się z bazą danych i oczywiście, jaki typ bazy danych lub zasobów jest zbierany. Poniższy wiersz kodu jest przykładem ciągu połączenia używanego do uzyskiwania dostępu do danych w bazie danych Microsoft Access o nazwie Personnel: Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Personnel.accdb; User ID=; Password=;

Obiekty danych ActiveX

Obiekty danych ActiveX (ADO) to interfejs programistyczny służący do łączenia aplikacji internetowych z bazą danych. ActiveX definiuje technologie, które umożliwiają aplikacjom, takim jak Word lub Excel, interakcję z siecią. Obsługa ActiveX została wyeliminowana z niemal każdej przeglądarki internetowej ze względów bezpieczeństwa. System Windows 10 nadal obsługuje ActiveX, ale tylko w przeglądarce Internet Explorer, a nie w Edge. Firma Microsoft nie usunęła całkowicie obsługi ActiveX, ponieważ wiele firm nadal z niej korzysta. Mimo że ActiveX jest na wylocie, nadal jest przydatny do demonstrowania sposobu łączenia się z bazami danych. Na przykład możesz wstawić arkusz kalkulacyjny programu Excel na stronę internetową. Aby uzyskać dostęp do bazy danych ze strony internetowej ASP, wykonaj następujące ogólne kroki:

1. Utwórz połączenie ADO z bazą danych, do której chcesz uzyskać dostęp.

2. Otwórz połączenie z bazą danych utworzone w kroku 1.

3. Utwórz zestaw rekordów ADO, który zawiera wiersze z tabeli, do której uzyskujesz dostęp.

4. Otwórz zestaw rekordów.

5. Wybierz potrzebne dane z zestawu rekordów na podstawie określonych kryteriów. 6. Zamknij zestaw rekordów.

7. Zamknij połączenie z bazą danych.

Następnie sprawdź, jak te kroki są wykonywane i jak wygląda wynik na stronie internetowej ASP lub ASP.NET. Poniższy kod tworzy i otwiera połączenie ADO:

```
<%

set conn=Server.CreateObject("ADODB.Connection")

conn.Provider="Microsoft.Jet.OLEDB.4.0"

conn.Open "c:\MyDatabase\employee.accdb"

%>
```

Now you need to create a recordset to contain records from a table in your employee.accdb database:

```
<%
set rs=Server.CreateObject("ADODB.recordset")
rs.Open "Select * FROM Employee", conn
…..
rs.close
conn.close
%>
```

Prawdopodobnie użyłbyś pętli, aby wydrukować wszystkie rekordy na stronie internetowej, ale to nie jest tutaj ważne. Chcesz zrozumieć technologię, aby móc rozpoznawać luki w zabezpieczeniach, gdy istnieją. Teraz, gdy masz już solidne podstawy dotyczące komponentów aplikacji internetowej, w poniższej sekcji omówiono niektóre z tych luk w zabezpieczeniach.

ZROZUMIENIE LUK W ZABEZPIECZENIACH APLIKACJI SIECIOWYCH

Do zaprojektowania witryny internetowej można użyć wielu platform i języków programowania. Każda platforma ma swoje zalety i wady. Niektóre są bezpłatne, a inne sporo kosztują; niektóre wymagają jedynie podstawowych umiejętności tworzenia aplikacji internetowych, a inne dogłębnej wiedzy z zakresu programowania. Niezależnie od platformy, specjaliści ds. bezpieczeństwa muszą ocenić system i zbadać potencjalne metody ataku na niego. Bezpieczeństwo sieci jest niezbędne do ochrony danych i zasobów firmy przed atakami. Bezpieczeństwo aplikacji, często określane jako AppSec, było kiedyś pomijane przez specjalistów, ponieważ jest to specjalistyczna praktyka. Jednym z powodów jest to, że wielu specjalistów ds. bezpieczeństwa ma doświadczenie w sieciach, ale niewielkie lub żadne doświadczenie w programowaniu. W rzeczywistości większość kursów dotyczących bezpieczeństwa sieci nie obejmuje zbyt wiele programowania, ponieważ temat ten może przytłoczyć studentów. Bez względu na to, jak wydajne są zapory sieciowe firmy lub systemy wykrywania włamań, większość systemów zezwala na zawartość ruchu HTTPS. Dlatego atakujący może ominąć domniemane granice bezpieczeństwa, a także wszelkie wzmocnienia systemu operacyjnego, które wykonali administratorzy sieci. Mówiąc prościej, ochrona warstwy sieciowej nie zawsze zapobiega atakom na warstwę aplikacji. Atakujący potrzebuje jedynie zrozumienia podstawowych pojęć programowania lub języków skryptowych. Aby dodać chaosu, atakujący zazwyczaj nie potrzebują specjalnych narzędzi, a wykrycie ręcznego ataku na aplikację internetową jest często trudne. Po przejęciu kontroli nad serwerem internetowym atakujący mogą użyć szeregu działań poeksploatacyjnych, w tym:

• Szkodzenie witryny internetowej

• Próba zniszczenia bazy danych aplikacji lub sprzedanie jej zawartości

• Próba przejęcia kontroli nad kontami użytkowników

• Uruchamianie ataków wtórnych z serwera internetowego lub infekowanie systemów odwiedzających witrynę złośliwym oprogramowaniem

• Próba uzyskania dostępu do innych serwerów, które są częścią infrastruktury sieciowej

Luki w zabezpieczeniach aplikacji i środki zaradcze

Na szczęście istnieje organizacja, która pomaga specjalistom ds. bezpieczeństwa zrozumieć luki w zabezpieczeniach aplikacji internetowych. Podobnie jak ISECOM, Open Web Application Security Project (OWASP) to fundacja non-profit, której celem jest wyszukiwanie i zwalczanie przyczyn luk w zabezpieczeniach aplikacji internetowych. OWASP (www.owasp.org) publikuje dokument „Ten Most Critical Web Application Security Risks", który został wbudowany w standard bezpieczeństwa danych Payment Card Industry (PCI) (DSS). PCI DSS jest wymogiem dla wszystkich firm sprzedających produkty online. Zaleca się odwiedzenie witryny OWASP, aby dowiedzieć się więcej o lukach w zabezpieczeniach aplikacji internetowych. Dokument OWASP i jego lista 10 najważniejszych zagrożeń są aktualizowane co kilka lat. Najnowsza edycja ma datę wydania w 2021 r. Jako tester bezpieczeństwa możesz potrzebować przeanalizować luki w zabezpieczeniach, takie jak te z listy OWASP top 10:

• A1 — Luki w zabezpieczeniach typu injection występują, gdy niezaufane dane są akceptowane jako dane wejściowe do aplikacji bez ich prawidłowej weryfikacji. Każdy fragment danych wysłany z przeglądarki internetowej na serwer może zostać zmanipulowany, a zatem stanowi potencjalny punkt ataku. Jeśli atakujący może założyć, w jaki sposób dane mogą być obsługiwane na serwerze, może podjąć świadome próby wykorzystania serwera. Typy luk w zabezpieczeniach typu injection obejmują SQL, kod, LDAP i wstrzykiwanie poleceń.

• A2 — Luki i słabości uwierzytelniania są powszechne, gdy do kontrolowania lub ochrony procesu uwierzytelniania używane są słabe zarządzanie sesjami, słabe schematy szyfrowania lub słaba logika. Programiści często „tworzą własne" schematy uwierzytelniania lub szyfrowania zamiast wykorzystywać istniejące, sprawdzone biblioteki. Jedno małe niedopatrzenie programisty może prowadzić do poważnych słabości.

• A3 — Ujawnienie poufnych danych ma miejsce, gdy nie zostaną podjęte odpowiednie środki ostrożności w celu ochrony danych aplikacji w stanie spoczynku i w trakcie przesyłania. Ujawnienie po stronie klienta może obejmować poufne informacje, które są buforowane i pozostają na dysku twardym użytkownika po użyciu aplikacji. Jest to szczególnie niebezpieczne, jeśli użytkownicy sprawdzają stany swoich kont bankowych na publicznym komputerze, takim jak udostępniony w bibliotece, a buforowane informacje zawierają poufne dane bankowe, których atakujący może użyć do przeprowadzenia oszustwa. Szyfrowanie danych w stanie spoczynku po stronie serwera powinno być używane w celu ochrony poufnych danych, takich jak hasła i inne informacje o klientach. Aby zachować poufność danych w trakcie przesyłania, szyfrowanie musi być zawsze wymuszane przez aplikację.

• A4 — Zewnętrzne jednostki XML (XXE) są problematyczne, gdy starsze lub źle skonfigurowane procesory XML oceniają odwołania do zewnętrznych jednostek w dokumentach XML. Zewnętrzne jednostki mogą być używane do ujawniania wewnętrznych plików za pomocą programu obsługi URI plików, wewnętrznych udziałów plików, wewnętrznego skanowania portów, zdalnego wykonywania kodu i ataków typu „odmowa usługi".

• A5 — Złamana kontrola dostępu ma miejsce, gdy reguły dotyczące tego, co uwierzytelnieni użytkownicy mogą robić, nie są prawidłowo egzekwowane. Atakujący mogą wykorzystać te luki, aby uzyskać dostęp do nieautoryzowanych funkcji lub danych, takich jak wyszukiwanie kont innych użytkowników, przeglądanie poufnych plików, modyfikowanie danych innych użytkowników i zmiana praw dostępu.

• A6 — Błędy konfiguracji zabezpieczeń wynikają ze źle skonfigurowanych technologii, na których działa aplikacja internetowa. Należą do nich system operacyjny, serwer aplikacji, serwer WWW, usługi używane do konserwacji itd. Podstawowe linie konfiguracji i listy kontrolne mogą pomóc administratorom zapobiegać błędnym konfiguracjom zabezpieczeń.

• A7 — Luki w zabezpieczeniach typu cross-site scripting (XSS), takie jak luki w zabezpieczeniach typu injection, wynikają z akceptowania przez serwer niezaufanych, niezweryfikowanych danych wejściowych. Istnieją dwa typy luk w zabezpieczeniach XSS: przechowywane i odbite. Przechowywane, czasami nazywane „trwałym XSS", są szczególnie szkodliwe, ponieważ mogą zostać dostarczone kolejnym użytkownikom aplikacji. Odbity XSS opiera się na inżynierii społecznej, aby oszukać użytkownika i nakłonić go do odwiedzenia złośliwie spreparowanego łącza lub adresu URL. W obu przypadkach celem atakującego jest wykonanie kodu na komputerze zdalnego użytkownika. Aby to osiągnąć, atakujący wstrzykuje kod do podatnego parametru aplikacji. Serwer wysyła ten kod do przeglądarki ofiary. Przeglądarka użytkownika uruchamia następnie wstrzyknięty kod, powodując szkodliwe działanie na komputerze użytkownika.

BAJTY BEZPIECZEŃSTWA

Najbardziej niesławny robak XSS nazywał się „JS.Spacehero" lub „Samy" i rozprzestrzeniał się poprzez przejęcie przeglądarek odwiedzających witrynę MySpace. Twórca robaka przesłał złośliwy skrypt na swoją stronę profilu MySpace, a każda osoba odwiedzająca tę stronę była przekierowywana do wysłania mu prośby o dodanie do znajomych. Robak został następnie osadzony na stronie profilu przejętego użytkownika. W ciągu niecałych 24 godzin zainfekowanych zostało ponad milion stron profilu MySpace, co uczyniło robaka JS.Spacehero jednym z najszybciej rozprzestrzeniających się robaków w historii. MySpace musiał zamknąć witrynę, aby oczyścić infekcję, a twórca robaka został skazany za przestępstwo. W 2021 roku Koo, rodzimy klon Twittera w Indiach, musiał załatać lukę w zabezpieczeniach XSS. Luka ta mogła umożliwić hakerom uruchomienie złośliwego JavaScript, który mógłby ukraść poufne dane wszystkich sześciu milionów aktywnych użytkowników Koo.

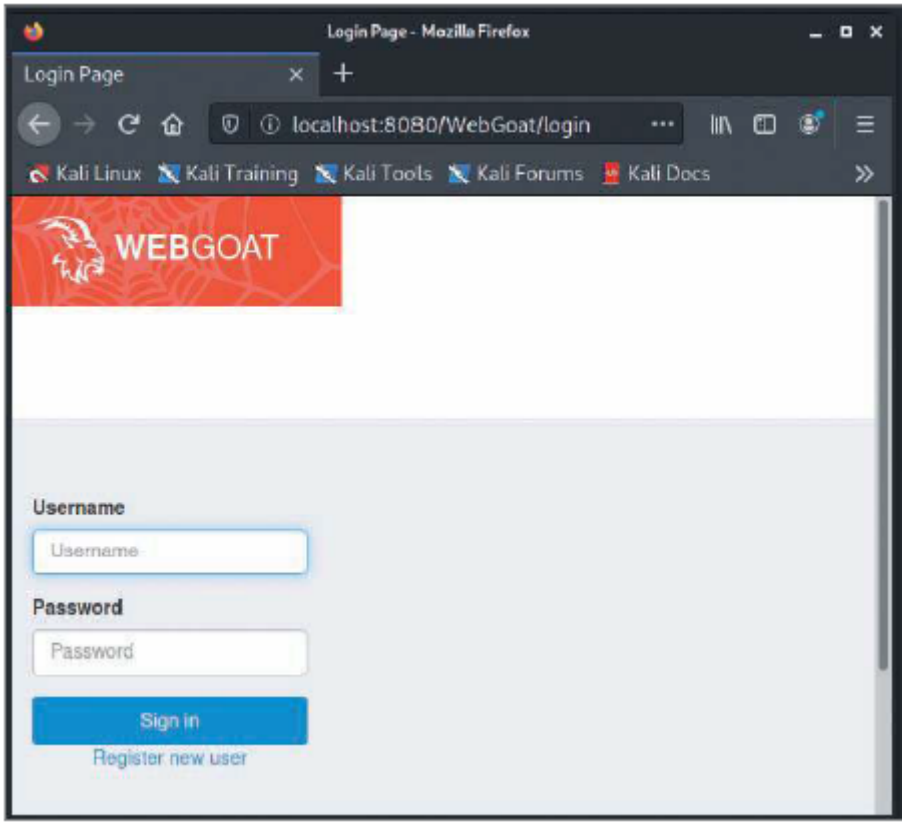
• A8 — Niebezpieczna deserializacja może prowadzić do zdalnego wykonywania kodu, ataków typu replay, ataków typu injection i ataków typu privilege-escalation. Serializacja dzieli obiekt na części i wyraża te części w innym formacie danych, który można później przywrócić. Deserializacja przywraca zserializowane części, aby utworzyć oryginalny obiekt. Jeśli hakerzy przechwycą informacje o niebezpiecznej deserializacji, mogą wykorzystać te informacje do wykonania exploitów.

• A9 — Korzystanie z komponentów ze znanymi lukami powoduje, że aplikacje internetowe korzystające z tych komponentów dziedziczą te luki. Komponenty, takie jak biblioteki, struktury i inne moduły oprogramowania, działają z tymi samymi uprawnieniami co aplikacja. Jeśli zostanie wykorzystany podatny komponent, taki atak może ułatwić poważną utratę danych lub przejęcie serwera. Aplikacje i interfejsy API korzystające z komponentów ze znanymi lukami mogą osłabić obronę aplikacji i umożliwić różne ataki i skutki.

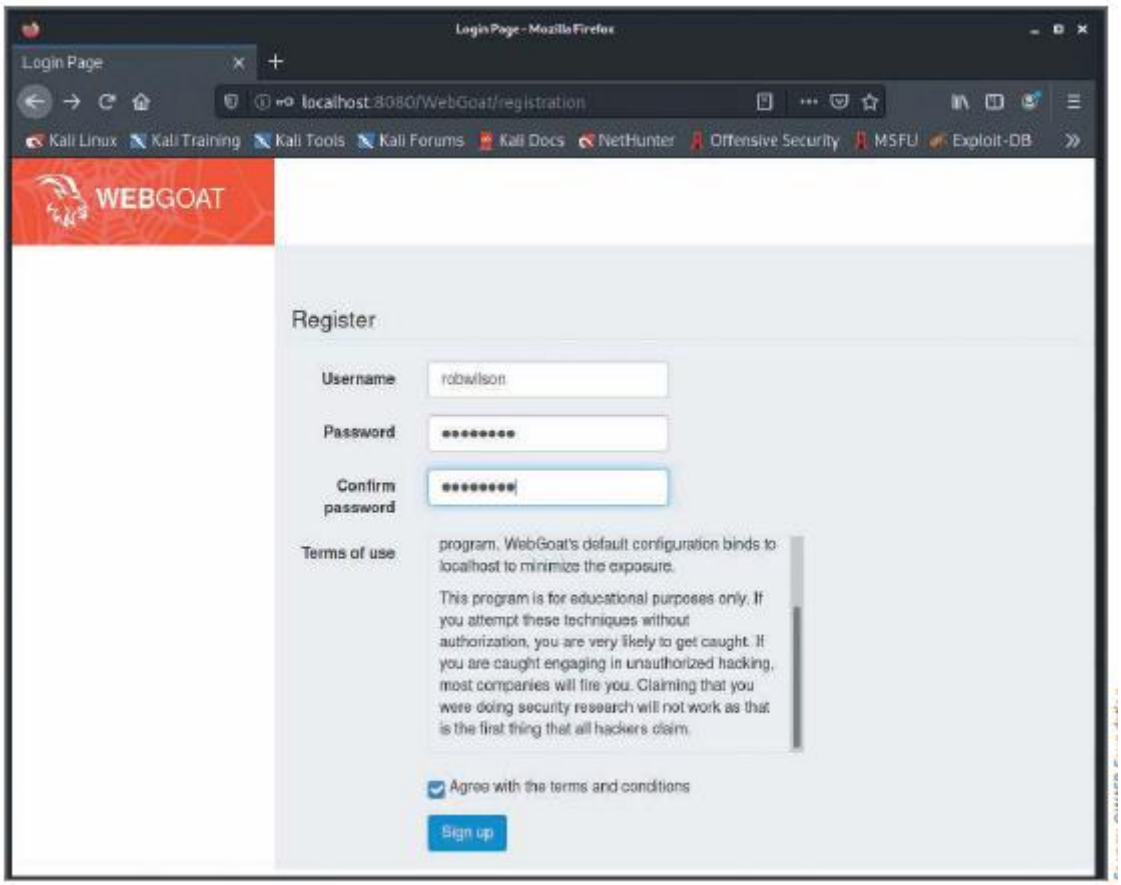
• A10 — Niewystarczające rejestrowanie i monitorowanie może pozwolić atakującym pozostać niezauważonymi. W połączeniu z brakującą lub nieskuteczną integracją z reagowaniem na incydenty, ta podatność pozwala atakującym na dalsze atakowanie systemów, utrzymywanie trwałości, przechodzenie na więcej systemów oraz manipulację, ekstrakcję lub niszczenie danych. Większość badań naruszeń pokazuje, że czas wykrycia naruszenia wynosi ponad 200 dni, zwykle wykrywane przez strony zewnętrzne, a nie przez wewnętrzne procesy lub monitorowanie.

Dokument OWASP na temat 10 największych luk w zabezpieczeniach może obejmować pewne obszary wykraczające poza umiejętności początkującego testera zabezpieczeń, dlatego OWASP oferuje Broken Web Apps i WebGoat, narzędzie online, które pomaga początkującym testerom zabezpieczeń zrozumieć luki w zabezpieczeniach aplikacji internetowych omówione na tej liście. OWASP opracował projekt WebGoat, aby pomóc testerom zabezpieczeń nauczyć się, jak przeprowadzać testy podatności w zabezpieczeniach aplikacji internetowych. Eksperci z całego świata korzystają z WebGoat i oferują swoje uwagi. Deweloperzy OWASP chcą zachęcić studentów bezpieczeństwa do zastanowienia się nad

tym, jak przeprowadzić atak, więc rozwiązania nie są podane dla wszystkich ćwiczeń. W poniższych akapitach przejdziesz przez przykład użycia WebGoat, aby dowiedzieć się o podstawowych atakach na aplikacje internetowe. Możesz śledzić, jeśli chcesz, lub po prostu przejrzeć kroki i rysunki. Zakładając, że uruchomiłeś Kali Linux, będziesz musiał pobrać łatwą do uruchomienia wersję wykonywalną jar WebGoat z github. W terminalu uruchom następujące polecenie: wget https : // github . com /WebGoat/WebGoat/releases/ download/v8.0.0.M26/webgoat-server-8.0.0.M26.jar. Następnie przejdź do miejsca, w którym pobrałeś plik jar easy-run. Wpisz java -jar webgoat-server-8.0.0.M26.jar, aby uruchomić plik wykonywalny. WebGoat jest regularnie aktualizowany, a dla nowych wydań tworzony jest nowy plik .jar. Jeśli zdecydujesz się pobrać najnowszą wersję lub inną wersję, nazwa pobieranego i uruchamianego pliku może różnić się od tej wyświetlanej tutaj. Możesz sprawdzić stronę WebGoat Releases pod adresem https://github.com/WebGoat/WebGoat/releases, aby określić numer wersji bieżącego wydania, który jest wyświetlany na górze strony. Przewiń w dół do sekcji Assets strony i najedź kursorem na plik webgoat-server .jar, aby wyświetlić nazwę ścieżki HTTPS do użycia z poleceniem wget. Otwórz przeglądarkę i przejdź do http : // localhost:8080/WebGoat, aby wyświetlić stronę startową WebGoat przedstawioną na rysunku.



Kliknij łącze Zarejestruj nowego użytkownika na dole strony logowania, aby utworzyć nazwę użytkownika i hasło. Podczas kolejnych wizyt na stronie logowania WebGoat możesz użyć tych samych danych uwierzytelniających. Rysunek przedstawia stronę rejestracji.



Po zarejestrowaniu się lub zalogowaniu zostaniesz przeniesiony do modułu wprowadzającego „Co to jest WebGoat?”. Przeczytaj wprowadzenie, a następnie kliknij łącze Ogólne w panelu nawigacyjnym po lewej stronie i wybierz Podstawy protokołu HTTP, aby wyświetlić stronę przedstawioną na rysunku 10-10. Przeczytaj informacje na pierwszej stronie, która przedstawia koncepcję, cele i sposób działania protokołu HTTP. Użyj strzałek nawigacyjnych i ikon, aby przejść do następnej strony, na której wpiszesz nazwę. Serwer akceptuje żądanie HTTP i odwraca dane wejściowe. Na przykład wpisanie nazwy „student” zwraca wartość „tneduts”. Przejdź do ostatniej strony, aby wziąć udział w krótkim quizie na temat podstaw protokołu HTTP. Ćwiczenia stają się bardziej złożone po pierwszym, więc prawdopodobnie nie będziesz w stanie wykonać ich szybko. Na przykład ćwiczenie bezpieczeństwa SQL Injection (intro) uczy podstaw SQL i SQL injection oraz zawiera dynamiczne sekcje, w których możesz samodzielnie wypróbować exploity SQL injection. SQL to skrót od Structured Query Language, popularnego języka używanego przez witryny internetowe do uzyskiwania dostępu do baz danych. SQL injection to exploit, w którym złośliwy aktor wprowadza polecenia SQL do pól wprowadzania witryny internetowej w celu obejścia zabezpieczeń i uzyskania dostępu do danych. (SQL injection jest omawiany bardziej szczegółowo później w tym module). Inne ćwiczenia pokazują różne aspekty bezpieczeństwa sieci. Na przykład jedno ćwiczenie wymaga skonfigurowania konfiguracji klient/serwer, aby można było wykryć ruch zawierający poświadczenia do witryny internetowej; inne obejmuje korzystanie z podatnych komponentów, takich jak funkcje z repozytoriów kodu open source GitHub, które mają znane luki w zabezpieczeniach. Ostatni zestaw ćwiczeń, nazywany wyzwaniami, wprowadza początkujących uczniów na wyższy poziom, wymagając od nich złamania schematów uwierzytelniania, wymuszenia zresetowania hasła administratora i głosowania w ankiecie bez logowania się.

Wykonywanie testów aplikacji internetowych

Aplikację można testować przy użyciu dwóch głównych technik: statycznego testowania bezpieczeństwa aplikacji (SAST) i dynamicznego testowania bezpieczeństwa aplikacji (DAST). SAST analizuje kod źródłowy aplikacji pod kątem luk w zabezpieczeniach i dlatego jest możliwe tylko wtedy, gdy kod źródłowy aplikacji jest dostępny. SAST to niezawodny sposób na wyliczenie większości luk w zabezpieczeniach aplikacji wynikających z błędów kodowania. SAST jest również znany jako „testowanie białej skrzynki". DAST analizuje działającą aplikację pod kątem luk w zabezpieczeniach. Może być również używany razem z AST w celu ustalenia priorytetów ustaleń SAST. Jeśli kod źródłowy nie jest dostępny dla testerów, mogą oni wykonać tylko DAST. DAST jest również znany jako „testowanie czarnej skrzynki". Inna technika testowania aplikacji zwana interaktywnym testowaniem bezpieczeństwa aplikacji (IAST) łączy elementy SAST i DAST i wykorzystuje agenta wewnątrz aplikacji do wykonywania analizy w czasie rzeczywistym w dowolnym momencie procesu rozwoju. IAST jest również znany jako „testowanie szarej skrzynki". Ta sekcja koncentruje się głównie na wykonywaniu DAST. Kilka list kontrolnych i przewodników dotyczących testowania bezpieczeństwa przeprowadza testera bezpieczeństwa przez dynamiczne testowanie każdego komponentu aplikacji internetowej, zapewniając pełne pokrycie. Poniższe sekcje są oparte na „OWASP Web Application Penetration Testing Guide" (www.owasp.org/index.php/Web_Application_Penetration_Testing), który został opracowany i sprawdzony przez ekspertów z całego świata.

Gromadzenie informacji i mapowanie architektury

W zależności od poziomu doświadczenia testera i ilości szkoleń w zakresie zabezpieczania infrastruktury internetowej, zrozumienie niuansów tworzenia aplikacji internetowych może być trudne. Pamiętaj, że w większości przypadków zespoły są wykorzystywane podczas wykonywania testu bezpieczeństwa. Jeśli masz tylko niewielkie doświadczenie w zakresie aplikacji internetowych, możesz rozważyć dodanie członka zespołu, który ma wiedzę specjalistyczną w tym temacie. Każdy obszar omówiony w poniższych sekcjach może wymagać specjalistycznej wiedzy. Pytania, które należy rozważyć w tej fazie, obejmują: Czy aplikacja ma bazę danych? Czy aplikacja wymaga uwierzytelniania? Czy aplikacja ma strony statyczne czy dynamiczne? Jakich języków i platform używa aplikacja? Czy między przeglądarką internetową a aplikacją znajdują się urządzenia zaprojektowane w celu zapobiegania atakom? W jaki sposób dane przepływają w aplikacji?

Bezpieczeństwo platformy i konfiguracja

Biorąc pod uwagę tak wiele platform dostępnych dla programistów internetowych, nic dziwnego, że istnieje tak wiele luk w zabezpieczeniach. Wiedza o tym, czy aplikacja internetowa została opracowana na serwerze IIS z ASP.NET i SQL Server, czy w systemie Linux Apache Web Server z wykorzystaniem PHP i MySQL, daje atakującym i testerom bezpieczeństwa amunicję do wykonywania swojej pracy. Pamiętaj, że przeprowadzasz odciski, aby dowiedzieć się, jakiego systemu operacyjnego i DBMS używa atakowany system. Im więcej wiesz o systemie, tym łatwiej jest zebrać informacje o jego lukach w zabezpieczeniach i typowych błędach w konfiguracjach. Pytania, które należy rozważyć w tej fazie, obejmują: Czy podstawowe platformy i komponenty zawierają znane luki w zabezpieczeniach? Czy serwer internetowy jest skonfigurowany w celu ochrony poufności użytkowników, którzy się z nim łączą? Czy istnieją interfejsy administracyjne do komponentów infrastruktury i testowanych aplikacji?

Uwierzytelnianie i testowanie sesji

Wiele aplikacji internetowych wymaga, aby serwer inny niż serwer internetowy uwierzytelniał użytkowników. Na przykład aplikacja internetowa może wymagać użycia serwera Windows Server 2019 z usługami Active Directory do uwierzytelniania. W takim przypadku należy zbadać, w jaki sposób informacje uwierzytelniające są przekazywane między dwoma serwerami. Czy używany jest szyfrowany kanał, czy też dane są przekazywane w postaci zwykłego tekstu, który można łatwo

odzyskać? Czy serwer używany do uwierzytelniania jest prawidłowo skonfigurowany i poprawiony? Czy informacje o logowaniu i haśle są przechowywane w bezpiecznej lokalizacji, czy też istnieje możliwość, że intruz uzyska dostęp do tych informacji i je odzyska? Inne pytania, które należy rozważyć w tej fazie, to: Czy sesja użytkownika odpowiednio wygasa? Czy aplikacja używa plików cookie i czy są one przechowywane w bezpieczny sposób? Czy witryna pozwala użytkownikowi na wylogowanie się i czy ta funkcja wylogowania wygasa sesję?

Testowanie autoryzacji

Autoryzacja to czynność sprawdzania uprawnień użytkownika w celu zezwolenia lub odmowy dostępu do strony, pola, zasobu lub akcji w aplikacji. Twórcy aplikacji często używają ukrytych pól w tabelach i ukrytych adresów URL, aby wymusić kontrolę dostępu zamiast sprawdzać uprawnienia użytkowników przed przetworzeniem żądania. Jeśli użytkownicy o niskich uprawnieniach wiedzą, jakich adresów URL żądać, mogą zwiększyć swoje uprawnienia w źle zaprojektowanej aplikacji. Niebezpieczne bezpośrednie odwołania do obiektów, opisane wcześniej, są odkrywane podczas testowania autoryzacji. Testowanie autoryzacji może ujawnić główne obszary zainteresowania i jest ważną częścią każdego testu aplikacji.

Walidacja danych wejściowych

Walidacja danych wejściowych to działanie polegające na filtrowaniu, odrzucaniu lub oczyszczaniu niezaufanych danych wejściowych użytkownika przed ich przetworzeniem przez aplikację. Problemy z walidacją danych wejściowych mogą prowadzić do ujawnienia, zmiany i zniszczenia danych. Gdy aplikacje akceptują niezaufane dane wejściowe, atakujący często może tworzyć złośliwe żądania, które powodują, że aplikacja wykonuje nieoczekiwane działanie. W tym miejscu pojawia się podatność na wstrzyknięcie. Atakujący może spowodować uruchomienie kodu przez serwer, przepełnienie bufora, wykonanie zapytań do bazy danych, odesłanie złośliwej zawartości z powrotem do użytkowników i wiele innych destrukcyjnych działań. Jednym z przykładów nieudanej walidacji danych wejściowych jest wstrzyknięcie SQL. Aplikacje internetowe, które proszą użytkowników o informacje lub wyświetlają dostępny inwentarz użytkownikom, zazwyczaj mają serwer bazy danych zaplecza przechowujący wszystkie te informacje. Baza danych inwentarza zawiera tabele zawierające informacje do wyświetlenia klientom, a baza danych klientów zwykle przechowuje dane o użytkownikach, które mogą obejmować informacje o kartach kredytowych. W tym przypadku bezpieczeństwo bazy danych ma pierwszorzędne znaczenie. Kto ma dostęp do tabel? Jakie oprogramowanie bazy danych i wersja są używane — na przykład Oracle 19c, Microsoft SQL Server 2019 lub MySQL? Czy istnieje możliwość użycia wstrzyknięcia kodu SQL do ataku na system? W przypadku wstrzyknięcia kodu SQL (SQLi) atakujący dostarcza polecenia SQL po wyświetleniu monitu o wypełnienie pola aplikacji internetowej. Poniżej znajduje się podstawowe polecenie SQL w celu wybrania rekordów (wierszy) w tabeli o nazwie books:

SELECT * FROM books WHERE lname ="Leno";

W przypadku wstrzyknięcia kodu SQL atakujący wstawiają („wstrzykują”) własne polecenia SQL w ramach tego polecenia. Pamiętasz formularz logowania utworzony za pomocą HTML w sekcji „Składniki aplikacji internetowej”? Aby zademonstrować luki w zabezpieczeniach SQL injection, możesz użyć strony internetowej ASP, która jest podobna do tego formularza logowania. Nie ma znaczenia, czy strona internetowa jest oparta na ASP czy ASP.NET. Luka w zabezpieczeniach SQL injection wynika ze strony internetowej korzystającej z bazy danych SQL i nie wykonującej walidacji danych wejściowych. Prawidłowa walidacja danych wejściowych może uniemożliwić użytkownikom wprowadzanie instrukcji SQL do monitów wejściowych „Wprowadź nazwę użytkownika:” lub „Wprowadź hasło:” w poniższym formularzu wejściowym strony internetowej ASP.

<form name="Validate" action="validate.asp" method="post">

Enter your username: <input type="text" name="username">

Enter your password: <input type="text" name="password">

<input type="submit">

</form>

Zawartość parametrów username i password jest przekazywana do strony ASP, validate.asp. Aby zweryfikować, czy username i password są poprawne, wiele aplikacji internetowych ma zabezpieczoną bazę danych prawidłowych nazw użytkowników i haseł. Strona validate.asp może wyglądać mniej więcej tak:

<%

1. Dim username, password, sql_statement

2. Dim conn, rs

3. username = Request.Form("username")

4. password = Request.Form("password")

5. set conn = server.createObject("ADODB.Connection")

6. set rs = server.createObject("ADODB.Recordset")

7. sql_statement ="SELECT * FROM customer

WHERE tblusername ='" & username & "' AND

tblpassword '" & password & "'"

conn.Open "Provider=SQLOLEDB; Data Source=(local);

8. Initial Catalog=CustomerDB; User Id=sa; Password="

9. rs.activeConnection = conn

10. rs.open sql_statement

11. if not rs.eof then

12. response.write "Welcome!"

13. else

14. response.write "Please reenter your username and password"

15. end if

%>

Numery wierszy są podane wyłącznie w celach informacyjnych i nie są używane w rzeczywistej aplikacji internetowej ASP. Wiersze 1 i 2 deklarują zmienne używane w pozostałej części kodu: username, password, sql_statement, conn (dla połączenia) i rs (dla zestawu rekordów). Dim oznacza dimension, który był używany w czasach programowania BASIC do deklarowania zmiennych. Wiersze 3 i 4 definiują zmienne username i password. W wierszach 5 i 6 polecenia set conn i set rs tworzą obiekt ciągu

połączenia i obiekty zestawu rekordów, które zostaną użyte. W wierszu 7 zmienna sql_statement przechowuje instrukcję SQL używaną do zapytania bazy danych. Wiersze 8 i 9 pokazują, że SQLOLEDB, dostawca OLE DB dla SQL Server, jest używany do łączenia się z serwerem bazy danych. W tym przypadku uzyskuje się dostęp do bazy danych o nazwie CustomerDB, jak pokazano w instrukcji Catalog. Wiersz 10 pokazuje, że przechowując instrukcję SQL w zmiennej, można ją później uruchomić za pomocą rs.open sql_statement. Linia 11 sprawdza znacznik końca pliku (EOF). Jeśli nie zostaną znalezione żadne rekordy zgodne z tym, co wprowadził klient (nazwa użytkownika, hasło), przeglądarka internetowa wyświetli komunikat o konieczności ponownego wprowadzenia nazwy użytkownika i hasła. Jeśli zostanie znalezione dopasowanie, polecenie SELECT wyświetli listę wszystkich rekordów w tabeli customer. Więc jaki jest problem, pytasz? Spójrz na tabelę customer, która została utworzona i polecenie wstawiania do niej czterech rekordów:

```
CREATE TABLE customer

(

tblCustomerID CHAR (10);

tblusername VARCHAR(25);

tblpassword VARCHAR(25);

/

INSERT INTO customer (tblusername, tblpassword)

VALUES ("bob", "password");

INSERT INTO customer (tblusername, tblpassword)

VALUES ("ted", "pa$$w0rd");

INSERT INTO customer (tblusername, tblpassword)

VALUES ("alice", "G0uLd");

INSERT INTO customer (tblusername, tblpassword)

VALUES ("carol", "n@tw00d");
```

Jeżeli Bob zaloguje się przy użyciu swoich danych logowania, polecenie SELECT będzie wyglądać następująco:

```
SELECT * FROM customer

WHERE tblusername = 'bob>' AND tblpassword ='password'
```

Załóżmy, że Bob wpisuje poniższe dane, gdy zostanie poproszony o podanie nazwy użytkownika:

```
OR 1=1- -
```

W takim przypadku polecenie SQL wygląda następująco:

```
SELECT * FROM customer

WHERE tblusername = ' OR 1=1-- AND tblpassword = ' '
```

Ponieważ 1 = 1 jest zawsze prawdą, zapytanie jest wykonywane pomyślnie. Podwójne myślniki (- -) są używane w SQL do oznaczania komentarza. Czy istnieją inne sztuczki hakowania bazy danych? Przyjrzyj się kilku innym rzeczom, które atakujący mógł wprowadzić, gdy został poproszony o nazwę użytkownika i hasło:

Please enter username: ' OR 1=1--

Please enter password: ' OR 1= 1—

W takim przypadku polecenie SQL wygląda następująco:

SELECT * FROM customer

WHERE tblusername = ' OR 1=1-- AND tblpassword = ' OR 1=1- -

Zamiast porównywania wartości przez polecenie SQL, użytkownik wprowadza wartości w tabeli klienta, porównuje cudzysłów z innym cudzysłowem, co oczywiście zwraca prawdziwy warunek. W związku z tym zwracane są wszystkie wiersze. Zaskakujące jest, że wiele systemów podłączonych do Internetu ma tę lukę. Nie należy testować tej luki, próbując wstrzyknąć kod SQL na stronach internetowych, ponieważ ten atak jest uważany za inwazyjny i podlega karze karnej. Należy jednak testować wszystkie aplikacje internetowe, gdy wykonuje się test bezpieczeństwa i jest się do tego upoważnionym na piśmie. Podstawowe testy powinny obejmować:

• Czy można wprowadzać tekst zawierający znaki interpunkcyjne dowolnego rodzaju

• Czy można wprowadzać pojedynczy cudzysłów, po którym następują słowa kluczowe SQL, takie jak WHERE, SELECT, INSERT, UNION itd.

• Czy podczas próby wstrzyknięcia instrukcji SQL pojawia się jakiś błąd bazy danych (co oznacza, że wstrzyknięcie kodu SQL jest możliwe)

Czasami aplikacja internetowa nie daje testerowi żadnej wskazówki, że instrukcja SQL została uruchomiona. OWASP nazywa to „ślepym wstrzyknięciem SQL", które ma własny zestaw testów wymaganych do wykrycia. Atakujący może wstrzyknąć polecenie waitfor delay '00:00:10'– do MSSQL. Jeśli polecenie SQL zostanie pomyślnie przetworzone, serwer otrzyma polecenie, aby odczekał 10 sekund przed udzieleniem odpowiedzi. Jeśli się nie powiedzie, serwer odpowie bez opóźnienia. Tego i innych sztuczek można użyć do wykrycia ślepego wstrzyknięcia SQL.

BAJTY BEZPIECZEŃSTWA

Kiedy studenci aplikują na studia podyplomowe, oczekiwanie na list akceptacyjny może być bolesne, ale haker zaoferował im sposób na szybkie uzyskanie odpowiedzi. Haker uzyskał dostęp do wewnętrznych rejestrów przyjęć na Harvard, Stanford, MIT i inne czołowe szkoły biznesu, wykorzystując luki odkryte w aplikacji internetowej o nazwie ApplyYourself. Następnie haker opublikował wskazówki dotyczące hakowania na forum internetowym Business Week. Aby dowiedzieć się więcej, odwiedź stronę www.thecrimson.com/article/2005/3/3/hacker-tips-off-b-school-applicants-tipped/. Kandydaci, niezależnie od ich przeszłości hakerskiej, mogli teraz dowiedzieć się, czy zostali przyjęci. Harvard Business School zidentyfikowała 119 kandydatów, którzy zhakowali system i oświadczyła, że odrzuci ich przyjęcia z powodu naruszenia zasad etyki. Niektórzy ludzie uważali, że problemem był brak zabezpieczeń na serwerze internetowym ApplyYourself, co pozwalało atakującemu po prostu zmodyfikować wyświetlany adres URL serwera internetowego. Złapani kandydaci używali swoich nazw logowania i zmieniali tylko adres URL po połączeniu z serwerem internetowym. Nie próbowali ukrywać swoich śladów ani zgadywać haseł. Czy to, co robili, było

nieetyczne? Na to pytanie może być trudno odpowiedzieć, ale Harvard nie miał problemu z robieniem właśnie tego. Chociaż incydent ten miał miejsce lata temu, nadal jest to dobra lekcja na temat słabego bezpieczeństwa witryny i konsekwencji hakowania.

Obsługa błędów

Aplikację internetową można skonfigurować lub napisać tak, aby obsługiwała błędy na wiele sposobów. Na przykład, gdy aplikacja wymaga rozwiązania problemu, programiści mogą włączyć debugowanie, które zapewnia bogate informacje rejestrowania pomocne w diagnozowaniu problemów. Czasami, po rozwiązaniu problemu, tryb debugowania może pozostać włączony, co stanowi cenne źródło informacji dla atakujących. Programiści powinni zminimalizować ilość informacji udostępnianych użytkownikom, gdy aplikacja napotka błąd. W optymalnych warunkach, w takich przypadkach błędów użytkownikom nie powinny być wyświetlane żadne informacje lub tylko ogólny komunikat.

Testowanie kryptografii

Kryptografia może być onieśmielająca dla początkujących testerów bezpieczeństwa. Pamiętaj, że nie musisz rozumieć faktoryzacji liczb całkowitych i funkcji logarytmu dyskretnego, aby znaleźć wady w implementacji kryptografii. Wiele problemów w kryptografii wynika z prostych problemów: złych generatorów liczb losowych; znanej słabej metody szyfrowania; algorytmu szyfrowania ze znanymi wadami, które umożliwiają jego złamanie; aplikacji, która w rzeczywistości nie wymusza korzystania z bezpiecznych kanałów; lub certyfikatu podpisanego samodzielnie zamiast zakupionego certyfikatu. Aby odkryć wady w rzeczywistych algorytmach, może być konieczne dodanie członka zespołu, który ma doświadczenie w analizowaniu procedur kryptograficznych. Często, gdy programiści decydują się na stworzenie własnych schematów kryptograficznych zamiast korzystania ze wspólnych ram kryptograficznych, doświadczony tester może znaleźć sposób, aby je obalić.

Testowanie logiki biznesowej

Logika biznesowa odnosi się do procedury, którą użytkownik ma wykonać w aplikacji, aby osiągnąć cel. Na przykład przed wykonaniem przelewu bankowego użytkownik musi najpierw spełnić wymóg posiadania co najmniej takiej kwoty pieniędzy na koncie przelewu. Jeśli użytkownik nie ma wystarczających środków, przelew powinien zostać wstrzymany. Testowanie logiki biznesowej polega na wykorzystaniu kreatywnych sposobów na ominięcie tego typu kontroli. Czy możesz w jakiś sposób oszukać aplikację, aby pomyślała, że masz 1 000 000 USD na swoim koncie, gdy masz tylko 100 USD? Jakie może to mieć implikacje dla banku, który autoryzuje transakcję przelewem bankowym na kwotę 1 000 000 USD z konta o wartości 100 USD? Tego typu wady mogą narazić firmę na problemy prawne, finansowe i reputacyjne i powinny być punktem skupienia podczas testowania aplikacji.

Testowanie po stronie klienta

Problemy po stronie klienta wynikają z wykonywania kodu na komputerze użytkownika, zazwyczaj w przeglądarce internetowej. Często, gdy programiści nie chcą, aby pole zostało zmienione, używają JavaScript po stronie klienta, aby je wyłączyć. Zwykły użytkownik nie może edytować tego pola, ale atakujący lub tester może użyć przechwytującego serwera proxy, aby manipulować nim w trakcie przesyłania. Kontrole po stronie klienta są niewystarczające same w sobie i powinny być sparowane z kontrolkami po stronie serwera, których nie można ominąć. Oto kilka innych kluczowych pytań, które należy zadać podczas testu po stronie klienta: Czy aplikacja przechowuje poufne informacje na komputerze klienta w sposób niezabezpieczony? Czy aplikacja zezwala na przekierowanie przeglądarki klienta, jeśli serwer otrzyma specjalnie spreparowane żądanie?

Aktywność 10-5: Badanie luk w zabezpieczeniach SQL Injection i praktyczne testy SQL Injection za pomocą WebGoat

Czas trwania: 30 minut

Cel: Rozpoznanie wielu platform, które mają luki w zabezpieczeniach SQL Injection, i przeprowadzenie testu SQL Injection za pomocą WebGoat.

Opis: Po ustaleniu, że aplikacja internetowa używa serwera bazy danych zaplecza do przechowywania danych, tester bezpieczeństwa powinien spróbować przetestować aplikację internetową pod kątem luk w zabezpieczeniach SQL Injection. W tej aktywności odwiedzasz witrynę Common Vulnerabilities and Exposures (CVE), aby zidentyfikować znane luki w zabezpieczeniach.

1. W razie potrzeby uruchom przeglądarkę internetową i przejdź do https://cve.mitre.org.

2. Na stronie głównej CVE kliknij łącze wyszukiwania CVE na pasku nawigacyjnym.

3. Na stronie wyszukiwania CVE List wpisz SQL injection w polu tekstowym i kliknij Submit. Ile wpisów CVE jest wymienionych? 4. Przewiń listę luk i kandydatów, czytając opisy każdego wpisu na pierwszej stronie. Na końcu listy kliknij przycisk Wstecz w przeglądarce, wpisz SQL injection phpbb w polu tekstowym, a następnie kliknij Prześlij. Ile wpisów jest wymienionych? Mimo że wymienione CVE są dość stare, wiele witryn może być nadal podatnych na ataki phpBB.

5. Gdy atakujący odkryje lukę, tak jak zrobiłeś to w tej aktywności, następnym krokiem jest próba znalezienia firm korzystających z oprogramowania. Aby znaleźć te informacje, możesz użyć wyszukiwarki. Na przykład, historycznie witryny korzystające z oprogramowania phpBB dodają przypis do stron głównych z informacją „Obsługiwane przez phpBB". Przejdź do swojej ulubionej wyszukiwarki i wyszukaj „obsługiwane przez phpbb." (wraz ze znakami cudzysłowu). Ile witryn jest wymienionych w wynikach wyszukiwania? Czy uważasz, że większość witryn naprawiła odkrytą przez Ciebie lukę?

6. Odwiedzanie tych witryn nie jest zalecane. Ponieważ hakerzy stosują ten sam proces, aby znaleźć podatne witryny do zhakowania, prawdopodobnie niektóre z tych witryn zawierają złośliwy kod zaprojektowany w celu zainfekowania systemu. Jakie są konsekwencje bezpieczeństwa związane z wymienianiem typu oprogramowania, którego używasz na stronie internetowej?

7. Zamknij przeglądarkę internetową.

8. Przejdź do instalacji Kali Linux i zaloguj się do WebGoat. Aby zdobyć praktyczne doświadczenie w wstrzykiwaniu kodu SQL, wykonaj ćwiczenie Wstrzykiwanie kodu SQL (wprowadzenie). Jeśli czas na to pozwoli, wykonaj również zaawansowane ćwiczenia wstrzykiwania kodu SQL i ćwiczenia łagodzące.

Jak dowiedziałeś się w tej aktywności, niektóre witryny używają phpBB. Po odkryciu luki atakujący szukają jak największej liczby celów do ataku. Jak widać, powiadomienie klientów o odkryciu luki jest kluczowe — im szybciej, tym lepiej

NARZĘDZIA DLA ATAKUJĄCYCH W SIECI I TESTERÓW BEZPIECZEŃSTWA

Po odkryciu luk w zabezpieczeniach aplikacji internetowej lub platformy systemu operacyjnego testerzy bezpieczeństwa lub atakujący szukają narzędzi, które umożliwią im przetestowanie lub zaatakowanie systemu. Na przykład, jeśli dowiesz się o lukach w zabezpieczeniach CGI, następnym krokiem jest sprawdzenie, czy jakieś systemy używają CGI. Jak widziałeś w poprzedniej sekcji, wszystkie platformy i komponenty aplikacji internetowych mają luki w zabezpieczeniach. Niezależnie od tego, która platforma jest używana do opracowywania aplikacji internetowej, prawdopodobnie ma ona lukę w zabezpieczeniach, którą atakujący mogą wykorzystać, aby włamać się do systemu.

Narzędzia internetowe

Widziałeś już, że większość narzędzi do przeprowadzania testów bezpieczeństwa lub atakowania sieci można znaleźć w Internecie i zazwyczaj są one bezpłatne. Kali Linux jest wyposażony w bezpłatne narzędzia do hakowania aplikacji internetowych, które można znaleźć w menu Kali Web Application Analysis. Możesz zainstalować nowe narzędzia za pomocą polecenia apt-get install packagename. Jednak inne narzędzia mogą być bardziej odpowiednie do określonego zadania. Poniższe sekcje obejmują niektóre popularne narzędzia do hakowania aplikacji internetowych. Witryny internetowe, na których można znaleźć inne narzędzia do testowania aplikacji internetowych, to www.owasp.org/index.php/Appendix_A:_Testing_Tools i https://packetstormsecurity.org. Jako tester bezpieczeństwa powinieneś odwiedzić te witryny przed testem, aby śledzić wszelkie nowe narzędzia i przeglądać dostępne exploity. Exploity opublikowane na stronie internetowej Packet Storm i stronie internetowej Exploit Database (www.exploit-db.com/) są często dodawane do wtyczek Metasploit.

Wbudowane narzędzia programistyczne Firefox i Chrome

Firefox i Chrome są wyposażone w podobny zestaw narzędzi programistycznych, które są również przydatne dla testera bezpieczeństwa aplikacji. Narzędzia te umożliwiają atakującemu przeglądanie parametrów w żądaniach, badanie plików cookie, a nawet manipulowanie żądaniami i ponowne ich wysyłanie. Dostęp do tych narzędzi można uzyskać za pośrednictwem menu Ustawienia w przeglądarkach Firefox i Chrome.

Burp Suite i Zed Attack Proxy

Burp Suite jest zawarty w Kali Linux i oferuje testerowi szereg funkcji do testowania aplikacji internetowych i usług internetowych. Umożliwia przechwytywanie ruchu między przeglądarką internetową a serwerem, dzięki czemu można sprawdzać i manipulować żądaniami przed wysłaniem ich do serwera. Burp Suite może również indeksować, skanować i stosować brutalną siłę w aplikacjach. W rzeczywistości ma wiele podobieństw do Zed Attack Proxy, z którymi powinieneś być zaznajomiony. Burp Suite Pro i Zed Attack Proxy można często używać zamiennie. Rysunek 10-14 przedstawia funkcjonalność przechwytywania proxy Burp Suite, która umożliwia testerowi bezpieczeństwa sprawdzanie szczegółów żądań i odpowiedzi. Lewa strona rysunku 10-14 przedstawia wbudowaną przeglądarkę Burp Suite używaną do nawigacji do witryny cbc.ca. Prawa strona rysunku przedstawia próbę nawigacji przechwytywaną przez proxy Burp Suite. Burp Suite Community Edition to darmowa wersja, która jest wstępnie zainstalowana w Kali Linux.

Wapiti

Wapiti to skaner luk w zabezpieczeniach aplikacji internetowych, który wykorzystuje podejście czarnej skrzynki, co oznacza, że nie sprawdza kodu. Zamiast tego sprawdza witrynę, wyszukując z zewnątrz sposoby wykorzystania luk XSS, SQL, PHP, JSP i obsługi plików. Chociaż Wapiti może wykrywać typowe formularze, które umożliwiają przesyłanie lub wstrzykiwanie poleceń, używa „fuzzingu” — próby wstrzykiwania danych do wszystkiego, co je zaakceptuje. W ten sposób można odkryć nawet nowe luki w zabezpieczeniach. Inne skanery wyszukują tylko znane sygnatury luk w zabezpieczeniach. Możesz zainstalować Wapiti w systemie Kali Linux, używając polecenia sudo apt-get install wapiti.

PODSUMOWANIE MODUŁU

• Aplikacje internetowe można tworzyć na wielu platformach. Strony internetowe HTML mogą zawierać formularze, ASP.NET, CGI i języki skryptowe, takie jak VBScript i JavaScript. Należy jednak pamiętać, że języki skryptowe stanowią ponad połowę ataków na serwery internetowe.

• Wiele statycznych stron internetowych zostało zastąpionych dynamicznymi stronami internetowymi, które są tworzone w locie, gdy użytkownik wywołuje stronę. Dynamiczne strony internetowe można tworzyć przy użyciu różnych technik, w tym CGI, ASP.NET, ASP, PHP, ColdFusion i JavaScript.

• Formularze internetowe umożliwiają programistom tworzenie stron internetowych, z którymi odwiedzający mogą wchodzić w interakcje. Należy jednak zachować ostrożność, aby upewnić się, że pola formularza nie mogą zostać zmanipulowane przez atakujących.

• Aplikacje internetowe wykorzystują różne technologie do łączenia się z bazami danych, takie jak ODBC, OLE DB i ADO. Technologie te tworzą interfejs front-end, umożliwiając aplikacji internetowej łączenie się z bazą danych back-end.

• Możesz zainstalować IIS, aby przetestować swoje strony internetowe w systemie Windows.

• Luki w zabezpieczeniach aplikacji internetowych mogą mieć szkodliwe konsekwencje dla firmy. Atakujący może być w stanie zniszczyć witrynę firmy, zniszczyć krytyczną bazę danych, uzyskać dostęp do kont użytkowników, a nawet uzyskać dostęp do konta administratora lub dostępu root do innych serwerów aplikacji w sieci.

• Podczas przeprowadzania testów bezpieczeństwa aplikacji internetowych określ, czy użyto dynamicznych stron internetowych, czy aplikacja internetowa łączy się z bazą danych zaplecza, czy do uwierzytelniania użytkowników użyto osobnego serwera i jaka platforma została użyta do opracowania aplikacji internetowej.

• Aplikacje internetowe, które wchodzą w interakcje z bazami danych, mogą być podatne na ataki typu SQL injection.

• Dostępnych jest wiele narzędzi do testowania luk w zabezpieczeniach aplikacji internetowych (i atakowania serwerów internetowych), takich jak Burp Suite i Wapiti. Ponadto OWASP oferuje oprogramowanie typu open source, które pomaga specjalistom ds. bezpieczeństwa dowiedzieć się więcej o lukach w zabezpieczeniach aplikacji internetowych.