Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji

Kierunek: Cyberbezpieczeństwo

Bezpieczeństwo aplikacji webowych

Zadanie rozgrzewkowe

Politechnika Wrocławska

Krzysztof Bocian

Wrocław, 2021

1. Wstęp

Celem tego zadania jest stworzenie raportu z przejścia poszczególnych poziomów w grach programistycznych skierowanych na temat bezpieczeństwa webowego. Zadanie składa się z dwóch gier posiadających różne zadania na różnym poziomie trudności.

2. Pierwsza gra

Pierwsza gra znajduje się pod linkiem https://uw-team.org/hackme/.

2.1 Poziom 1

Na pierwszym poziomie hasłem jest “a jednak umiem czytac”. Można je znaleźć wyświetlając źródło strony.

2.2 Poziom 2

W kolejnym zadaniu hasłem jest “to bylo za proste”. Na tym poziomie należy ponownie wyświetlić źródło strony. Tym razem hasło jest schowane w pliku widocznym w źródle strony “haselko.js”.

2.3 Poziom 3

Poziom trzeci wymagał zrozumienia funkcji znajdujących się w źródle strony. Prawdziwie hasło przechowywane jest w zmiennej “ost”. Korzysta ona ze zmiennych “literki” oraz “dod”. “Ost” tworzone jest z 2-4 znaków “literki”, “qwe” oraz 3-6 znaków “dod”. Wiedząc jak powstaje hasło i znając zmienne możemy sami skonstruować hasło, którym jest “cdqwenow”. Należy pamiętać, że ciąg znaków zapisany w zmiennej zaczynamy numerować od zera.

Ciekawostką jest funkcja “right”. Jest to funkcja zabezpieczająca przed używaniem prawego przycisku myszy np. do badania elementu. Kiedyś było to używane do zabezpieczenia strony.

2.4 Poziom 4

Czwarte hasło ukryte jest pod działaniem matematycznym w źródle strony. Całe wyrażenie wygląda następująco:

Math.round(6%2)(258456/2))+(300/4)*2/3+121*

Znając działania matematyczne oraz kolejność wykonywania działań wynikiem tego wyrażenia jest “171”, które jest hasłem tego poziomu.

2.5 Poziom 5

Poziom piąty wykorzystywał w swoim haśle element czasowy. Strona odlicza co sekundę do 60 sekund. Hasło wykorzystuje działanie matematyczne:

(seconds(seconds-1))/2)*(document.getElementById('pomoc').value%2*

Całe działanie przyrównywane jest do 861. Aby odnaleźć hasło należy znaleźć sekundę oraz liczbę pomocniczą “pomoc” tak aby działanie wynosiło 861. Na liczbie pomocniczej wykonywane jest działanie modulo 2, co oznacza, że ostatecznie liczba będzie wyrażona 0 lub 1. Jedynym możliwym wyborem liczby pomocniczej jest 1 lub liczba która po modulo 2 da nam 1 np. 3, 5 itp. Wystarczy znaleźć odpowiednią sekundę do wpisania 1 aby otrzymać 861, a tak taką liczbę która pomnożona przez siebie o 1 mniejszą da wynik 1722. Jest to 42. Zatwierdzając liczbę 1 jako liczbę pomocniczą w 42 sekundzie przechodzimy poziom 5.

2.6 Poziom 6

Na poziomie 6 przydatna jest znajomość działania pętli. W kodzie źródłowym istnieje pętla, która wykonuje się trzy razy. Za każdym razem bierze jedną literę z zmiennej podanej w kodzie źródłowym i dodaje do niej “x” lub “_” w zależności od parzystości zmiennej “licznik” zwiększanej o 1 przy każdym wywołaniu pętli. W pętli do zmiennej “hsx” dopisywane są kolejne wywołania pętli. W ten sposób otrzymujemy “bxd_ex”. Po pętli mamy również linijkę kodu, która dodaje do zmiennej “hsx” trzy ostatnie jej znaki tworząc hasło “bxd_ex_ex”.

2.7 Poziom 7

Poziom 7 polega na odszyfrowaniu hasła. W kodzie źródłowym można zauważyć, że każdej literze przypisany jest inny znak. Dodatkowo podana jest informacja, że po

zaszyfrowaniu naszego hasła ma ono formę “plxszn_xrv”. Wystarczy odwrócić proces szyfrowania hasła, aby je otrzymać. W tym przypadku odpowiedzią jest “kocham cie”.

2.8 Poziom 8

Ostatni poziom składa się z dwóch etapów. Pierwszy jaki zrobiłem był pętlą, która wykonuje się 6 razy, dając ostatecznie po jednej literze na pętle biorąc znak z zmiennej “alf”. Pętla bierze znak (qet+i), gdzie “i” w pętli zwiększa się o 2 a “qet” o 1 co wykonanie pętli. Z tej części otrzymujemy ciąg znaków “qrupjf”, który jest pierwszą częścią hasła. Podana poniżej linijka kodu pokazuje nam z czego składa się hasło oraz czego nam jeszcze brakuje:

*wyn+=eval(ax*bx*cx)*

W zmiennej “wyn” jest zapisana wcześniej utworzona część hasła i dodawany jest do niej wynik funkcji “eval”, jednak zmienne tej funkcji nie są podane wprost w kodzie źródłowym. Na początku kodu znajduje się zapisek “src="%7A%73%65%64%63%78%2E%6A%73"”. Jest to zapis w “URL Encoded”. Korzystając z konwertera online możemy dowiedzieć się, że jest to nazwa pliku “zsedcx.js”. Dodając do adresu strony “/zsedcx.js” możemy dostać się do pliku w którym są zapisane zmienne (trzeba dodatkowo je obliczyć). Wykorzystując znalezione zmienne obliczamy wynik funkcji “eval” i otrzymujemy liczbę “162”, którą dodajemy do naszego hasła “wyn”. Odpowiedzią na to zadanie jest “qrupjf162”.

3. Druga gra

Druga gra znajduje się pod linkiem https://uw-team.org/hm2/.

3.1 Poziom 1

Na pierwszym poziomie badając źródło strony widzimy że podawane hasło musi się zgadzać z polem “formularz”, które jest ukrytym polem na stronie. Możemy je odsłonić badając element strony i usuwając atrybut “hidden”. Hasłem jest “text”.

3.2 Poziom 2

Drugi poziom posiada hasło zaszyfrowane w swoim kodzie źródłowym. Hasło zapisane jest w “encoded url”. Tłumacząc wiadomość na ascii otrzymujemy hasło “banalne”.

3.3 Poziom 3

Na trzecim poziomie hasło było zaszyfrowane tak samo jak na poprzednim poziomie lecz w systemie binarnym. Hasło: “1234”.

3.4 Poziom 4

Poziom czwarty jest bardziej skomplikowanym poziomem. Najpierw należy wyłączyć Jave w naszej przeglądarce, żeby dostać się do właściwej strony z kodem źródłowym. W nim znajdziemy znowu hasło zapisane w URL oraz informację, że hasło jest w systemie szesnastkowym. Po odszyfrowaniu informacji otrzymujemy liczbę 258, którą w systemie szesnastkowym zapisujemy jako 102 i jest to nasze hasło.

3.5 Poziom 5

Piąty poziom wymaga zaznajomienia się z językiem PHP. Skrypt podany jest wprost na stronie. Przejście dalej jest możliwe jedynie przy ustawieniu zmiennych “has” i “log” na wartość 1. Język PHP umożliwia to wpisując odpowiednio wartości w adres strony. Wpisując te wartości w okno logowania na stronie w adresie pokazują się zmienne “haslo” i “login”. Wystarczy je zmodyfikować i nadać odpowiednią wartość przy przejść ten etap.

3.6 Poziom 6

Na szóstym poziomie podpowiedzią była informacja o ciasteczku. Należy zbadać element strony i przejść do zakładki “network”. W tej zakładce jesteśmy w stanie podejrzeć ciasteczka związaną z daną stroną. W tym przypadku znajduje się tam ciasteczko z informacją pod jakim adresem znajduje się kolejny poziom i jest to dopisek do adresu strony “ciastka.htm”.

3.7 Poziom 7

Badając źródło siódmego poziomu otrzymujemy podpowiedź, aby wykorzystać właściwości serwera apache. Istnieje możliwość przeglądania plików jakie posiada serwer apache, jeśli nie jest on zabezpieczony, za pomocą przechodzenia do konkretnych folderów tego serwera wykorzystując adresy. Jednym z domyślnych folderów jest folder “include”. Przeglądając ten folder znajdujemy plik cosik.js, w którym znajduje się adres kolejnego poziomu “listing.php”.

3.8 Poziom 8

Przechodząc do etapu ósmego trudnością jest załadowanie strony. Ominąłem to wyłączając java w ustawieniach przeglądarki. Hasło na tym poziomie ukryte jest w źródle strony ale podane jest wprost. Hasłem jest “kxnxgxnxa”.

3.9 Poziom 9

Ostatnim poziomem są podziękowania dla “hakujących”. Kopiując wiadomość i dekodując ją z systemu binarnego do ASCII otrzymujemy wiadomość “gratulacje! udało Ci się ukończyć te wersje Hackme.”