

Bezpieczeństwo usług sieciowych

Laboratorium 2: Hackme

Szymon Bagiński

16 listopada 2018

1 Cel zadania

Celem zadania było ukończenie wszystkich poziomów gry Hackme i Hackme 2.0 dostępnych na stronie <https://uw-team.org/>.

2 Hackme

2.1 Level 1

W tym zadaniu należało odnaleźć funkcję `sprawdz()` w skrypcie znajdującym się na końcu źródła strony. w tej funkcji wartość pola z hasłem była porównywana do tekstu “a jednak umiem czytać”, który trzeba wpisać w pole, żeby ukończyć poziom.

2.2 Level 2

W nagłówku strony widać, że dołączany jest skrypt z pliku `haselko.js`. Znajdujemy w nim zmienną `has` z wartością “to było za proste”, którą należy wpisać jako hasło.

2.3 Level 3

Podane hasło w tym zadaniu jest porównywane do wartości zmiennej `ost`. Po przeanalizowaniu źródła strony widać, że są do niej przypisywane litery o indeksach 2 i 3 ze słowa “abcdefgh” (czyli “cd”). Dalej dopisywany jest napis “qwe”, a na końcu 3 ostatnie litery słowa “unknow”. Pełne hasło zatem brzmi “cdqwenow”.

2.4 Level 4

W źródle strony widzimy, że hasłem jest wartość wyrażenia `Math.round(6%2)*(258456/2)+(3004)*23+121`.

Nie musimy liczyć tego w pamięci. Można wyrażenie wkleić w konsoli przeglądarki i otrzymamy wynik: 171.

2.5 Level 5

W tym zadaniu należy sprawić aby zmienna `ile`, która jest obliczana według wyrażenia `((seconds*(seconds-1))2)*(document.getElementById('pomoc').value%2)` miała wartość 861 podczas sprawdzania hasła. Z tego wyrażenia wynika, że wartość pomocnicza jest dzielona modulo 2, więc możemy wywnioskować, że na pewno będzie to liczba nieparzysta, tak aby całe wyrażenie się nie wyzerowało. Dalej możemy wywnioskować, że wartość zmiennej `seconds` pomnożona przez liczbę o jeden od siebie mniejszą, a następnie podzielona przez 2 ma być równa 861. Po rozwiązaniu takiego prostego równania otrzymujemy wartość 42. Teraz wystarczy poczekać, aż licznik sekund przyjmie tę wartość i wcisnąć przycisk.

2.6 Level 6

W tym zadaniu należy przeanalizować pętlę i wywnioskować wartość zmiennej `hsx` podczas sprawdzania hasła na końcu funkcji `sprawdz`. Pętla wykona się 3 razy, a zmienna `i` będzie miało wartości: 1, 3 oraz 5. Następnie widzimy, że `licznik` na początku każdego wykonania pętli jest inkrementowany, więc dalej będzie miał wartości 1, 2 i 3. Na końcu pętli dodajemy literę o indeksie równym wartości zmiennej `i` z napisu "abdcqepolsrc" czyli w kolejnych iteracjach będą to: b, d oraz e. W każdej iteracji dodajemy także znak "_" jeśli `licznik` jest parzysty, oraz "x" jeśli tak nie jest. W kolejnych wykonaniach pętli zostaną dodane zatem "bx", "d_" oraz "ex". Po wykonaniu pętli duplikowane są trzy ostatnie litery wartości, która już jest w zmiennej `hsx`, więc otrzymane hasło to "bxd_ex_ex".

2.7 Level 7

W tym zadaniu musimy umieścić w zmiennej `wyn` napis "plxszn_xrv". Po przeanalizowaniu funkcji `sprawdz` widzimy, że każda litera, którą podamy jest zastępowana według wielu konstrukcji `if`, które się tam znajdują. Mapując litery odwrotnie można wywnioskować, że hasłem jest "kocham cie".

2.8 Level 8

W tym zadaniu dołączone są skrypty, których nazwa jest zapisana w systemie szesnastkowym. Nie musimy jednak ich konwertować na tekst. Wystarczy wkleić wartość w pasku adresu, a przeglądarka przekieruje nas do skryptu. Po otworzeniu tych plików okazuje się, że jeden z nich nazywa się "passwd.js". Jest to niestety tylko podły żart i nie ma w nim hasła. W drugim skrypcie, nazwanym "zsedcx.js" znajdziemy zdefiniowane pewne zmienne, które posłużą do odgadnięcia hasła. Teraz znowu możemy posłużyć się konsolą przeglądarki zamiast liczyć ręcznie jakie powinno być hasło. Kopiując wszystkie potrzebne zmienne ze skryptu oraz ze źródła strony, a także operacje, które są odpowiedzialne za obliczenie hasła otrzymujemy wartość "grupjf162".

3 Hackme 2.0

3.1 Level 1

3.2 Level 2

3.3 Level 3

3.4 Level 4

3.5 Level 5

3.6 Level 6

3.7 Level 7