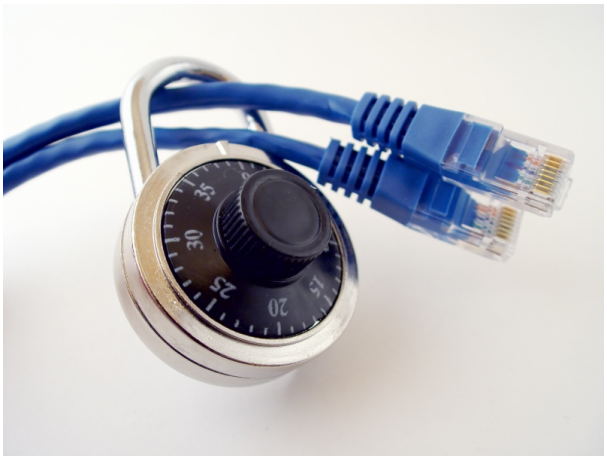

3 SPOSOBY NA BEZPIECZEŃSTWO WORDPRESS'A

Posted on 17 maja 2014 by Maciej Cybulski

Na warsztatach, na których omawiałem bezpieczeństwo WordPress'a, któryś z uczestników zapytał mnie o alternatywne metody zabezpieczania strony przed nieautoryzowanym dostępem...

Categories: aktualności, bezpieczeństwo, dla początkujących

Ostatnio na warsztatach, na których omawiałem bezpieczeństwo WordPress'a, któryś z uczestników zapytał mnie o alternatywne metody zabezpieczania strony przed nieautoryzowanym dostępem. Postanowiłem przyjrzeć się, co nowego w tej materii się pojawiło.

Alternatywne metody zabezpieczania strony i jak też samego logowania, to temat rzeka, natomiast warte uwagi są metody, które są najprostsze w obsłudze i zapewniające względnie wysoki poziom bezpieczeństwa. Tutaj wybór pada oczywiście na wieloskładnikowe metody autentykacji, a dokładniej 2FA - metody gdzie oprócz standardowego logowania loginem i hasłem dochodzi nam jeszcze jeden czynnik (klucz sprzętowy, smartfon lub skanowanie kodu)

Istotnym elementem wieloczynnikowego systemu zabezpieczeń m.in. jest istnienie w ścieżce autoryzacji „instytucji" potwierdzającej nasze prawa do dostępu do danej strony - Centrum

Autoryzacyjne. Czyli, by móc skorzystać z takiego rozwiązania musimy najczęściej mieć konto na stronie projektu, lub w inny sposób być jednoznacznie identyfikowani.

Wiadomo, każde rozwiązanie ma swoje mocne i słabe strony. Zatem przyjrzyjmy się kilku ciekawym rozwiązaniom.

RUBLON

Nasz polski „wynalazek" :) Bardzo ciekawa aplikacja. Funkcjonuje jako plugin do WordPressa i tej jej funkcjonalności się przyjrzymy. Służy do bezpiecznego logowania się do stron www, poprzez tzw. zaufane komputery. Wymaga założenia konta na stronie projektu.

By przystąpić do zabezpieczania strony należy mieć zainstalowany plugin WordPress Rublon, mieć założone konto na stronie projektu, oraz posiadać smartfona z zainstalowaną aplikacją Rublon. Zabezpieczanie naszej strony w tym przypadku jest już proste. Aktywujemy wtyczkę, która uruchamia nam wyświetlanie na ekranie QR kodu. Kod ten skanujemy aplikacją Rublon w smartfonie i potwierdzamy, że komputer, na którym się QRkod wyświetlił jest komputerem zaufanym.

Od tego momentu logowanie z tej przeglądarki na tym komputerze będzie już znacznie bezpieczniejsze mimo iż np. stosowalibyśmy słabe hasła. Po prostu po podaniu loginu i hasła w tle następuje też zweryfikowanie naszej przeglądarki (czy ma status zaufanego komputera). Jeśli weryfikacja przejdzie pomyślnie, zostaniemy zalogowani, jeśli nie wyświetli się nam QRkod i logowanie będzie musiało być potwierdzone przez naszego smartfona.

Świetne i wygodne rozwiązanie. Gdy z naszego komputera tylko my korzystamy, oraz jest on bezpieczny (ma skutecznego antywirusa) takie zabezpieczenie jest praktyczne nie do obejścia.

No właśnie jeśli ktoś inny nie ma dostępu do naszego komputera….

Kolejne bardzo ciekawe rozwiązanie „zaatakowało" mnie, gdy na jednej ze stron aktualizowałem motyw. Okazało się, że w motywie przewidziano wykorzystanie pluginu Clef.

Zaintrygowany przyjrzałem się mu bliżej. Do uruchomienia zabezpieczenia wymagane jest posiadanie konta na stronie projektu. Czyli musi być Centrum Autoryzacji.

Ok a co poza tym ciekawego ma sama wtyczka? Tak jak poprzednia wymaga do logowania użycia smartfona z zainstalowaną aplikacją Clef, tyle że tym razem jest już troszkę inaczej.

Po skonfigurowaniu strony i smartfona, by zalogować się musimy zeskanować w aplikacji w smarftonie ciekawy dynamiczny „kod kreskowy”. Zatem tylko przy pierwszej konfiguracji podajemy login i hasło, później już tylko ten dynamiczny „kod kreskowy” i smartfon służą nam do logowania.

Całkiem ciekawe rozwiązanie, mocno „zbajerowane” :)

Czy faktycznie trudne do złamania? Trudno powiedzieć. Natomiast z pewnością warto przetestować. Na pewno zabezpiecza przed atakami bruteforce, bo w ustawieniach można wyłączyć logowanie za pomocą loginu i hasła, pozostawiając tylko dynamiczny kod paskowy.

Do tego rozwiązania możemy też dorzucić „w parze” rozwiązanie dla przeglądarki Chrome pod nazwą Waltz - aplikacja ta umożliwia wykorzystanie „dynamicznego kodu kreskowego” do logowania do Facebooka, Googla i innych stron. Porównywane jest z LastPass, 1Password tylko podobno lepsze… No to kwestia ocenna. Natomiast ma całkiem ciekawą właściwość, mianowicie umożliwia logowanie na określony czas. Czyli wyloguje nas automatycznie po określonym czasie na który mamy wpływ.

Oba powyższe rozwiązania są bezpłatne, ale mają jeden czynnik którego nie ominiemy i na który nie mamy wpływu - centrum autoryzacyjne. Musimy stworzyć konto na stronie projektu by być autoryzowanym.

Rozwiązaniem które idzie krok dalej jest klucz sprzętowy Yubikey. firmy Yubico.

Jest to rozwiązanie płatne ale i niezwykle wszechstronne skuteczne. Mamy do dyspozycji kilka rodzajów kluczy, zróżnicowanych parametrami technicznymi i ceną. Ceny nie są wysokie patrząc na to, że rozwiązanie to ma gwarantować bezpieczeństwo stronie.

Do współpracy niezbędny jest też plugin dla WordPressa umożliwiający wykorzystanie technologii Yubico na stronie.

W zamian za 40$ otrzymujemy klucz, który można wykorzystać nie tylko na stronach internetowych, ale i w znacznie większej ilości przypadków m.in:

- logowanie do serwisów internetowych z silnym uwierzytelnianiem
- dostęp zdalny do komputerów i dostęp VPN
- zarządzanie aplikacjami do przechowywania haseł (Password Managment)
- logowanie do komputera
- logowanie za pomocą witryn SSO (Single Sign-On) np. OpenID, czyli rozwiązanie gdzie logujemy się w jednym miejscu a login i hasło „wędruje” z nami na kolejne strony. (Często widzimy na stronach „Zaloguj za pomocą Facebooka”)

- szyfrowanie dysków (współpraca z TrueCrypt)

Można też przykładowo połączyć kilka technik ze sobą np. na stronie internetowej zainstalować

Rublona i Yubikey. Nastepnie z naszego komputera zrobić „komputer zaufany” do którego dostępu będzie nam bronił klucz Yubikey. I dodatkowo ten sam klucz będzie bronił też dostępu do strony.

Lekko paranoiczne zabezpieczenie, ale pozwala spać spokojnie :)… no to możemy dalej zabrnąć z tą paranoją…

Jeśli boimy się zaufać Centrum Autoryzacji Yubico, to… producent dostarcza biblioteki programistyczne umożliwiające samodzielne stworzenie sobie na swoim serwerze własnego centrum autoryzacyjnego :)

Podsumowując. Co daje nam logowanie za pomocą 2FA w porównaniu z tradycyjnym logowaniem za pomocą loginu i hasła? Przyjrzyjmy się dokładniej:

1. Tradycyjne logowanie do wielu stron najczęściej wymaga pamiętania wielu loginów i haseł. Logowanie za pomocą 2FA nawet jeśli wymaga podania loginu i hasła, to mogą one być takie same dla wszystkich stron, gdyż drugi czynnik (QRkod, czy zaufany komputer) powoduje, że nawet przy uzyskaniu przez włamywacza loginu i hasła, zalogowanie nie będzie możliwe.
2. Mając na stronie wielu użytkowników w przypadku braku 2FA, musimy wymuszać na użytkownikach stosowanie bezpiecznych haseł.
   Rozwiązanie 2FA niejako nie daje innej możliwości jak stosowanie bezpiecznego logowania. Już nie pojawia się problem z długością haseł i problemem choćby z rotacyjnym ich zmienianiem.
3. Przechowywane przez WordPressa hasła są często celem ataków.
   Rozwiązanie 2FA eliminuje hasła lub dorzuca kolejny czynnik, który nie jest tak prosty do złamania… dlaczego? Już piszę…
4. Hakerzy wykorzystując współczesne techniki są w stanie w ciągu 24h złamać 90% haseł. Zatem oczywisty login powoduje drastyczne zmniejszenie bezpieczeństwa strony, gdyż zostaje „tylko” złamanie hasła. Stąd można by się pokusić o stwierdzenie, że unikalny login jest ważniejszy od samego hasła…
   Natomiast opisane powyżej rozwiązania stosują bądź hasła jednorazowe, bądź klucze RSA o długości 2048bitów, co póki co czyni je niezwykle trudnymi do złamania.

Zatem czas na testy na stronach produkcyjnych a rezultaty… czas pokaże.

Jeśli interesuje Cię temat zabezpieczeń WordPressa, lub po prostu uznasz że warto podzielić się tym artykułem z innymi - rób to śmiało i napisz w komentarzu. Nic tak nie mobilizuje do publikowania i dalszego zgłębiania tematu jak merytoryczna dyskusja :)

A może masz własne doświadczenia z zabezpieczaniem stron? … konieczni napisz o tym poniżej :)

There are no comments yet.