„Bezpieczna” E-Bankowość

security@man.poznan.pl

Zespół Bezpieczeństwa

PCSS

Technologia stosowana do realizacji bezpiecznych połączeń z portalami banków elektronicznych jest uznawana za bezpieczną, jeżeli zostanie poprawnie użyta. Raport ten zawiera opis oraz ocenę tej konfiguracji dla kilkudziesięciu portali bankowych w Polsce. Ocena ta została sporządzona na podstawie publicznie dostępnych informacji. Poza danymi statystycznymi w dokumencie znajdują się przykłady wykorzystania ujawnionych błędów konfiguracyjnych przez potencjalnego atakującego. Raport zakończony jest podsumowaniem oraz tabelarycznym zestawieniem portali, których dotyczyły testy wraz z wykrytymi błędami bezpieczeństwa.

Wersja 1.08, 15 luty 2006

Zespół Bezpieczeństwa PCSS zajmuje się analizami, badaniami oraz konsultacjami w zakresie bezpieczeństwa teleinformatycznego. Bierze udział w wielu europejskich projektach badawczych oraz świadczy usługi w zakresie testów penetracyjnych dla klientów komercyjnych. Więcej informacji na stronach zespołu (http://security.psnc.pl/).

Spis treści

Wprowadzenie

Bankowość elektroniczna jest zjawiskiem niemalże tak powszechnym jak bankomaty. Na stronach internetowych większości banków bez trudu można odnaleźć panel logowania, poprzez który możemy dostać się do informacji o naszym koncie, założyć lokatę, złożyć wniosek kredytowy. Oczywiście są to dane poufne. Banki zapewniają więc, że dbają o ich bezpieczeństwo, ale czy wypada wierzyć bankom na słowo?

Logując się do portalu banku można przeczytać, że stosuje on bezpieczne połączenie, szyfrowane połączenie, bezpieczne logowanie, itp. Niezależnie od tego jak nazywa się to na stronie banku technologia z której bank korzysta w celu zapewnienia bezpiecznego połączenia to SSL (*Secure Socket Layer*). Bezpieczne połączenie pomiędzy przeglądarką internetową a serwerem banku to zabezpieczenie podstawowe, pierwsza linia obrony. Samo w sobie nie zapewnia absolutnego bezpieczeństwa, jest to natomiast fundament jakiejkolwiek dalszej analizy bezpieczeństwa elektronicznego systemu bankowego przeprowadzanej z zewnątrz. Chociażby dlatego, że zapewnia poufność danych służących do logowania się do systemu, danych na temat stanu konta, danych osobowych, wszelkich danych wprowadzanych i wyprowadzanych z systemu. Zespół Bezpieczeństwa Poznańskiego Centrum Superkomputerowo – Sieciowego postanowił sprawdzić, na ile bezpieczne jest reprezentowane przez „kłódeczkę" „bezpieczne połączenie" do elektronicznych systemów bankowych. Analiza obejmowała kilkadziesiąt banków internetowych w Polsce i dotyczy stanu na dzień 8 luty roku 2006. Banki zostały wybrane na podstawie listy banków portalu *Twoje Finanse* [6].

Przeprowadzone testy dotyczyły tylko i wyłącznie konfiguracji SSL, opierały się na informacjach jawnych udostępnianych przez serwery systemów bankowych oraz wiedzy i narzędziach Zespołu Bezpieczeństwa PCSS. Wszystkie domniemane w czasie zbierania danych słabości zostały następnie potwierdzone podczas prób łączenia się z systemami bankowymi.

Sporządzony dokument kierowany jest do managerów, administratorów systemów bankowych oraz użytkowników bankowości elektronicznej. Dlatego też sporządzony został tak, aby zawierał jak najmniej szczegółów technicznych, a niezbędne do zrozumienia pojęcia zostały przybliżone w ramkach.

Przedstawione tutaj informacje służą jedynie podniesieniu poziomu świadomości oraz kwalifikacji. W tekście nie zawarto gotowych przepisów, którymi mogłyby posłużyć się osoby nieświadome. Dla tych, którzy są doskonale zorientowani w problemie raport nie będzie stanowił nowości. W ten sposób Zespół Bezpieczeństwa PCSS nie stwarza dodatkowego zagrożenia.

60 sekund

Pierwszym testem dokonywanym na serwerze banku elektronicznego było sprawdzenie możliwości łączenia się protokołem SSL w wersji 2.0. Jest to wersja, której od 1996 roku nie zaleca się używać, a w roku 1999 została praktycznie wyparta przez wersję 3.0. Najprostszym sposobem wykonania takiego testu, czymś co może zrobić praktycznie każdy, jest próba nawiązania połączenia z serwerem przy pomocy przeglądarki internetowej z włączoną obsługą jedynie protokołu SSL w wersji 2.0. Podczas przygotowywania raportu dla uproszczenia zastosowano specjalistyczne narzędzie, które automatyzuje procedurę. Cel tego testu jest jasny, protokół SSL w wersji 2.0 posiada wiele słabości i nie powinien być używany tam gdzie bezpieczeństwo jest szczególnie istotne [9].

Drugim testem, będącym rozwinięciem pierwszego była enumeracja wszystkich szyfrów, które serwer banku wspiera dla bezpiecznych połączeń. Celem tego testu było wykazanie, jaką minimalną długość klucza szyfrującego obsługuje serwer. W uogólnieniu długość klucza warunkuje kwotę nakładów finansowych oraz czasowych, jakie trzeba ponieść, aby ten klucz odtworzyć na podstawie zaszyfrowanych danych. Dla bardzo krótkich kluczy przy kwocie kilkuset euro czas ten czas ten jest bardzo krótki [7,8,9], co powoduje, że może to zrobić praktycznie każdy. Po odtworzeniu klucza nasze dane, które przesyłaliśmy tym połączeniem stają się automatycznie widoczne dla napastnika.

Protokół SSL 2.0 i 3.0

Protokół SSL pierwotnie został zaprojektowany przez firmę Netscape. Jego zadaniem było zapewnienie poufności oraz integralności danych przesyłanych przez Internet. Dodatkowo protokół miał zapewniać uwierzytelnianie stron połączenia – obowiązkowe dla serwera i opcjonalne dla klienta. Myślą inżynierów było też zaprojektowanie protokołu uniwersalnego, tak aby mogły z niego korzystać różne protokoły aplikacyjne, jak np. HTTP, FTP, TELNET, etc.). Wersja 2.0 była pierwszą wersją publicznie dostępną, pojawiła się w 1994 roku [1].

Szybko okazało się, że protokół ma wiele słabości [2]. Odpowiedzią na te słabości była wersja 3.0 protokołu SSL opublikowana w 1996 roku, a od 1999 roku uznawana za powszechnie używaną i wiodącą wersję tego protokołu [3].

Wreszcie, trzecim testem było sprawdzenie czy do serwisu internetowego banku elektronicznego można dostać się również przez połączenie kompletnie nie szyfrowane. Oczywiście takie połączenie musiałby nawiązać klient, ale aby to zrobił wystarczy drobna pomyłka we wpisywanym adresie czy też nieuważne kliknięcie np. przy przechodzeniu z jednej strony portalu na inną. Uznaje się, że bezpiecznym wzorcem projektowym jest umieszczenie serwisu pod osobnym adresem [10], do którego dostęp może odbywać się tylko poprzez kanał szyfrowany. Generalnie odradza się współdzielenie nazwy domenowej przez serwis z dostępem szyfrowanym i serwis z dostępem nie szyfrowanym, przynajmniej tam gdzie do kwestii bezpieczeństwa podchodzi się poważnie

Mając przygotowane środowisko oraz narzędzia przeprowadzenie takich testów dla pojedynczego portalu zajmuje ok. 60 sekund, mniej więcej tyle co przeczytanie tego punktu.

W kolejnym punkcie opisane są: słabości protokołu SSL 2.0, atak SSL 2.0 Rollback na zestaw protokołów SSL oraz problemy związane ze współdzieleniem domeny. Teoria ta jest uzasadnieniem przeprowadzanych testów.

Podstawy przeprowadzonych testów

Słabości protokołu SSL 2.0

Protokół SSL w wersji 2.0 posiada wiele słabości, dokładnych ich opis znajduje się w książce [2]. Skrócony opis przedstawiony został poniżej:

- Ponieważ protokół w tej wersji nie posiada zabezpieczenia procedury nawiązywania połączenia (*handshake*) możliwe jest wpłynięcie przez stronę trzecią na wybór algorytmu szyfrowania. Atakujący może wymusić użycie słabego algorytmu szyfrującego.
- W przypadku wersji 2.0 strona trzecia w sposób niedostrzeżony może usunąć część danych przesyłanych szyfrowanym połączeniem.
- Jedynym algorytmem zapewniającym integralność przesyłany dany w ramach połączenia SSL w wersji 2.0 jest MD5. Dzisiaj algorytm ten jest uznawany za słabo bezpieczny.
- Protokół SSL 2.0 używa tego samego klucza w celu zapewnienia poufności oraz integralności. Co powoduje, że złamanie jednego klucza umożliwia przeprowadzenie dowolnego ataku.
- Błąd projektowy protokołu w tej wersji umożliwia przechwycenie danych autoryzacyjnych klienta przez stronę trzecią połączenia. Chodzi tutaj o autoryzację i uwierzytelnianie protokołu SSL, a nie HTTP.

Wszystkie te wymienione problemy uzasadniają, dlaczego odradza się stosowania protokołu w tej wersji.

Atak SSL 2.0 Rollback

W celu zapewnienia kompatybilności z aplikacjami obsługującymi starszą wersję protokołu SSL w wersji 3.0 została wprowadzona możliwość nawiązania połączenia w wersji 2.0 jeśli jedna ze stron nie wspiera wersji 3.0, a obie wspierają wersję 2.0. Procedura takiej degradacji nazywa się w terminologii SSL procedurą wycofania (*rollback*).

Aby zapobiec atakom typu MITM (*man-in-the-middle*), powodującym taką degradację połączenia przez stronę trzecią, klient w momencie przejścia z wersji 3.0 na 2.0 przechodzi również w inny tryb kodowania pewnych informacji, co jest wykrywane przez serwer. Jeśli więc serwer wykryje inny tryb kodowania u klienta (co oznacza, że klient który wspiera wersję 3.0 łączy się korzystając z wersji 2.0) uniemożliwia nawiązanie połączenia. Oczywiście serwer jest w stanie wykryć ten tryb kodowania jeżeli sam wspiera wersję 3.0 protokołu SSL, a więc wydawałoby się że problem jest rozwiązany.

Z powodu niepoprawnej obsługi protokołu SSL przez część starszych aplikacji, w niektórych implementacjach protokołu SSL, jak np. OpenSSL została wprowadzona możliwość pomijania tej weryfikacji kodowania co umożliwiło korzystanie z tych aplikacji w połączeniu z serwerami obsługującymi różne wersje protokołu SSL. Umożliwiło również przeprowadzanie ataków MITM i degradację kanału szyfrowanego pomiędzy klientem a serwerem przez stronę trzecią. Z tego powodu w OpenSSL taka opcja została usunięta. Informacje na ten temat pojawiły się na stronach projektu OpenSSL 11 października 2005. Do dnia pisania tego raportu minęły więc już blisko 4 miesiące. Obecnie opcja taka jest nawet niepotrzebna. Aplikacje, które jej wymagały, wyszły już praktycznie z użycia. Poza tym jeżeli w grę wchodzi bezpieczeństwo to pewne kompromisy są niemożliwe do osiągnięcia

Atak typu *man-in-the-middle* (MITM) jest sytuacją, w której pomiędzy dwie strony połączenia ingeruje strona trzecia – atakujący. Jest on w stanie przechwytywać dane przesyłane w obu kierunkach, odczytywać je, modyfikować lub usuwać. Jako przykład takiego rodzaju ingerencji można podać sytuację, w której ktoś obcy jest w stanie czytać nasze emaile, sms'y oraz je modyfikować nie mają fizycznego kontaktu ani z nadawcą ani z odbiorcą albo sytuację w której składamy podpis cyfrowy pod deklaracją majątkową, a w rzeczywistości podpisaliśmy zupełnie inny dokument, np. poważne zobowiązanie finansowe.

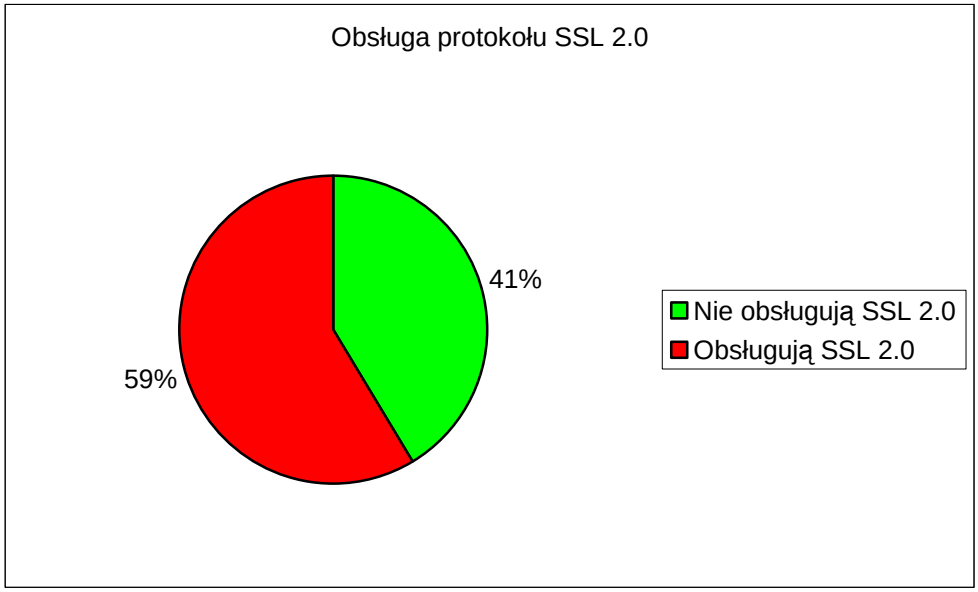
Problemy związane ze współdzieleniem domeny

URL identyfikujący aplikacje jest w pewnym sensie niezależny od protokołu dostępu do tej aplikacji. Można serwer WWW skonfigurować tak aby udostępniał aplikację poprzez HTTP oraz HTTPS. Jednak dobrym wzorcem projektowym jest taka konfiguracja, aby z aplikacją do której transmisja danych ma być bezpieczna, istniała możliwość łączenia się tylko poprzez kanał szyfrowany. Zabieg ten ma na celu m. in. uniemożliwić odwołanie się do serwera protokołem z otwartym kanałem danych. Ma to zapobiec przechwyceniu ewentualnie wysyłanych przez przeglądarkę *cookies*. W *cookies* mogą znajdować się poufne dane, jak np. identyfikator sesji umożliwiający przechwycenie sesji klienta z serwerem. Ponadto jeżeli z aplikacją można połączyć się poprzez HTTP, klient nie jest w stanie stwierdzić, podczas takiego połączenia, czy w połączenie nie ingeruje strona trzecia. Można łatwo sobie wyobrazić sytuację, w której na stronie banku umieszczony zostaje link do strony logowania do portal e-bankowego, w którym omyłkowo zamiast HTTPS użyto HTTP. Większość użytkowników zanim zorientuje się, że połączenie niej jest szyfrowane wyśle swoje dane w postaci czytelnej dla strony trzeciej. Gdyby portal e-bankowy uniemożliwiał nawiązanie połączenia kanałem otwartym, pomyłka taka zostałaby szybko wykryta i niebezpieczeństwo by nie zaistniało.

> ***Cookies*** czyli internetowe ciasteczka, są to informacje wysyłane pomiędzy serwerem, a przeglądarką w celu ich tymczasowego przechowywania. Mogą to być informacje o preferencjach użytkownika o ostatnio odwiedzanej przez niego części portalu lub informacje identyfikujące.

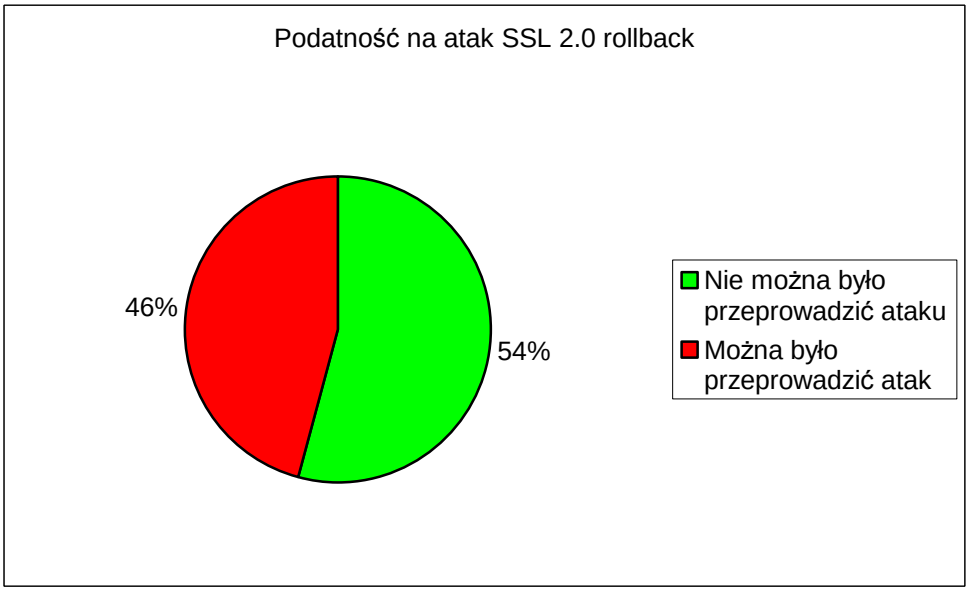
Zebrane dane

Zebrane dane dotyczą 41 portali należących do 31 banków. Z analizy statystycznej wynika, że wśród 41 portali, które były sprawdzane, aż 24 obsługują niebezpieczną wersję protokołu SSL (Rysunek 1). Z kolei z tych 24 aż dla 11 udało się przeprowadzić skuteczny atak na sesję przeglądarki ze stroną internetową (Rysunek 2). Mniej więcej świadczy to o tym, że od czasu publikacji informacji o podatności na atak SSL 2.0 rollback, tj. od 11 października 2005, w przypadku 46% portali banki nie zdecydowały się na instalację łaty bezpieczeństwa. Ponadto aż 63% z całkowitej liczby portali obsługuje słabe wersje protokołów szyfrujących (Rysunek 3). Wśród portali obsługujących protokół SSL 2.0 problem ten dotyczy 67% portali (Rysunek 4).

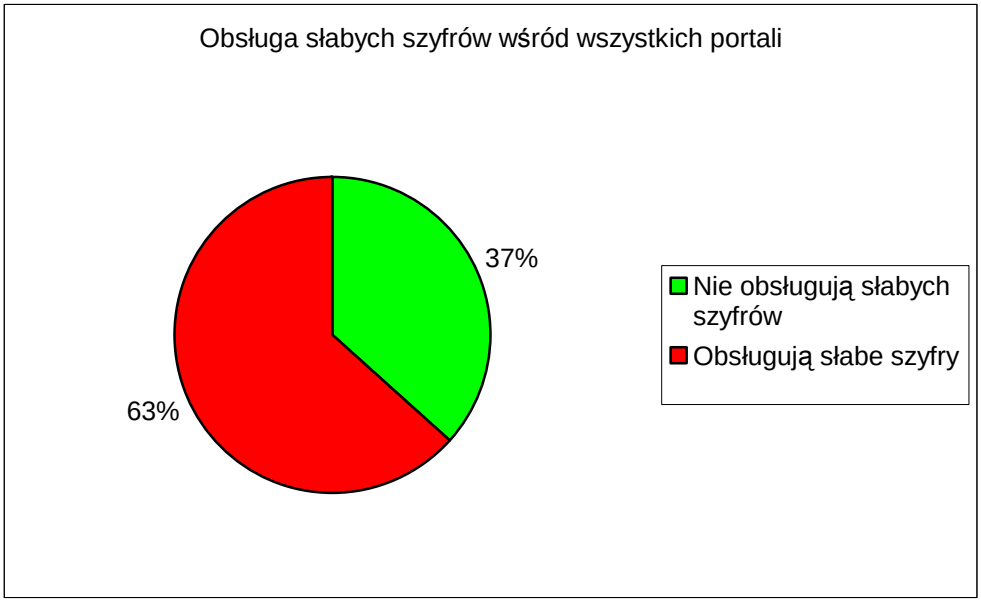


Rysunek 1. Procent portali bankowych obsługujących niebezpieczną wersję protokołu SSL

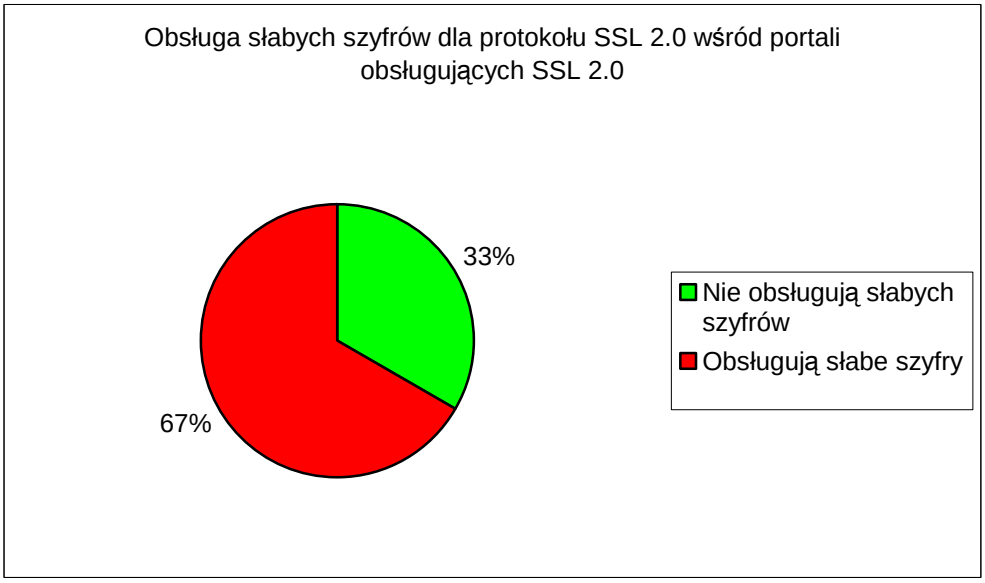
Okazało się również, że jedynie 16 banków nie umożliwia nawiązania połączenia protokołem nie szyfrowanym, 23 portale umożliwiają takie połączenie i w momencie połączenia wysyłają do klienta przekierowanie nakazujące mu nawiązać bezpieczne połączenie, 2 umożliwiają połączenie się z serwerem oraz przesyłanie danych do systemu poprzez kanał nie szyfrowany.



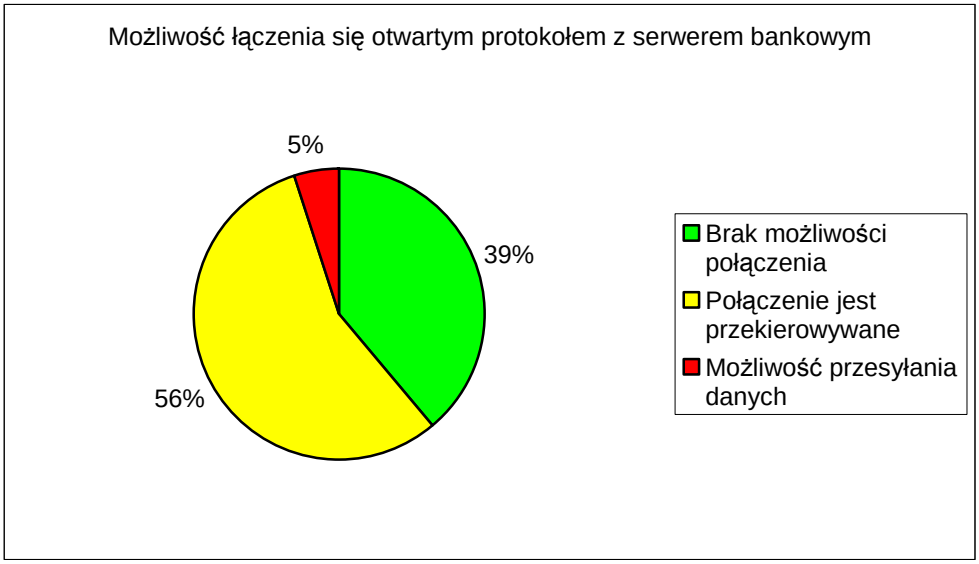
Rysunek 2. Procent serwerów dla których zweryfikowano podatność na atak SSL 2.0 Rollback



Rysunek 3. Odsetek serwerów obsługujących połączenia ze słabym szyfrowaniem



Rysunek 4. Liczba serwerów, które obsługują słabe szyfrowanie dla protokołu SSL 2.0



Rysunek 5. Statystyka dotycząca możliwości łączenia się z serwerem kanałem nie szyfrowanym w sytuacji gdzie takiej możliwości być nie powinno

Szersze komentarze na temat zgromadzonych danych znajdują się w podsumowaniu, natomiast lista poszczególnych banków – na końcu raportu.

Przykłady wykorzystania błędów bezpieczeństwa

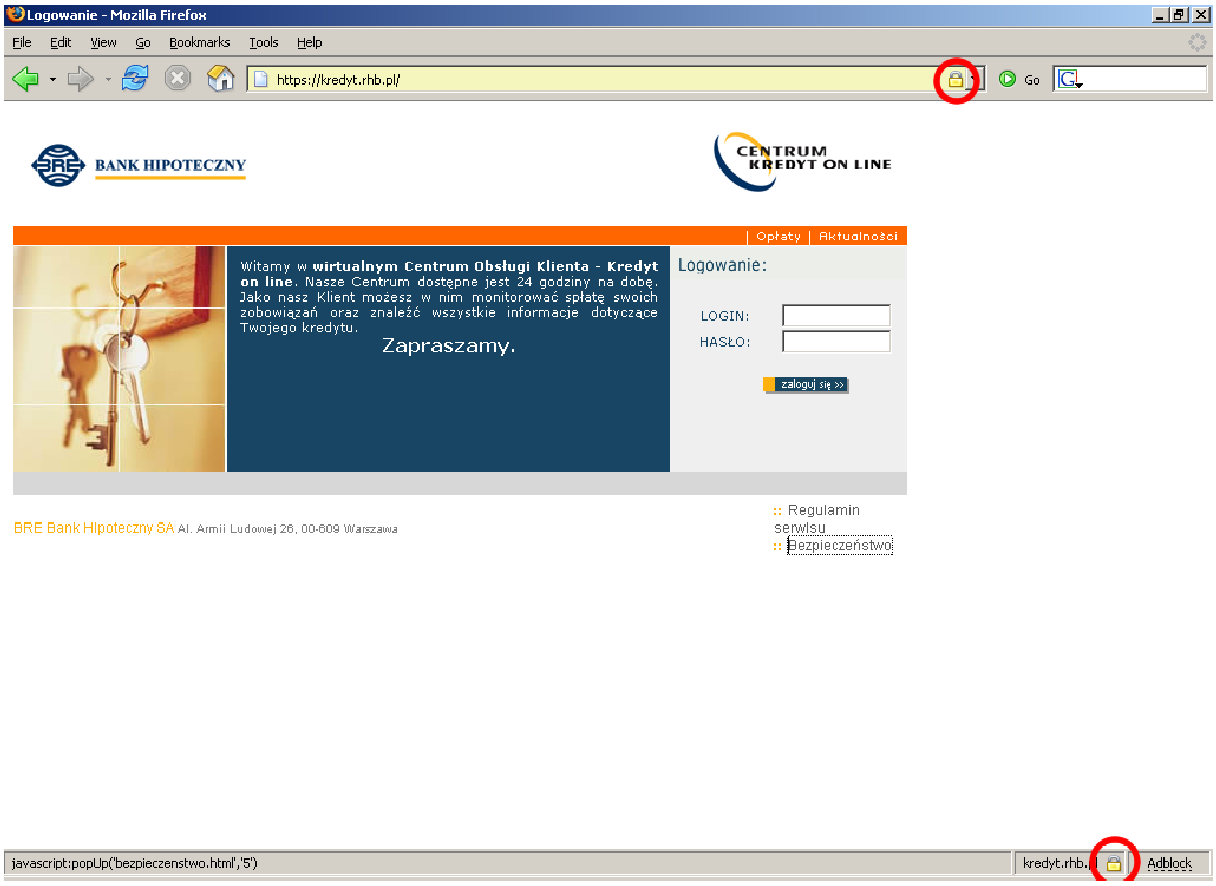
To co jest opisane powyżej to suche dane statystyczne oraz teoria dotycząca ataków. Jak wiadomo nie jest to coś co przemawia do użytkownika, dlatego postanowiliśmy zilustrować problem przykładem. Przykłady zostały dodatkowo poparte scenariuszami hipotetycznych, aczkolwiek realnych, ataków. Scenariusze uwzględniają atak z sieci lokalnej, ale jak pokazują badania, w ostatnich latach to właśnie sieć lokalna jest największym źródłem zagrożeń.

Przykład 1

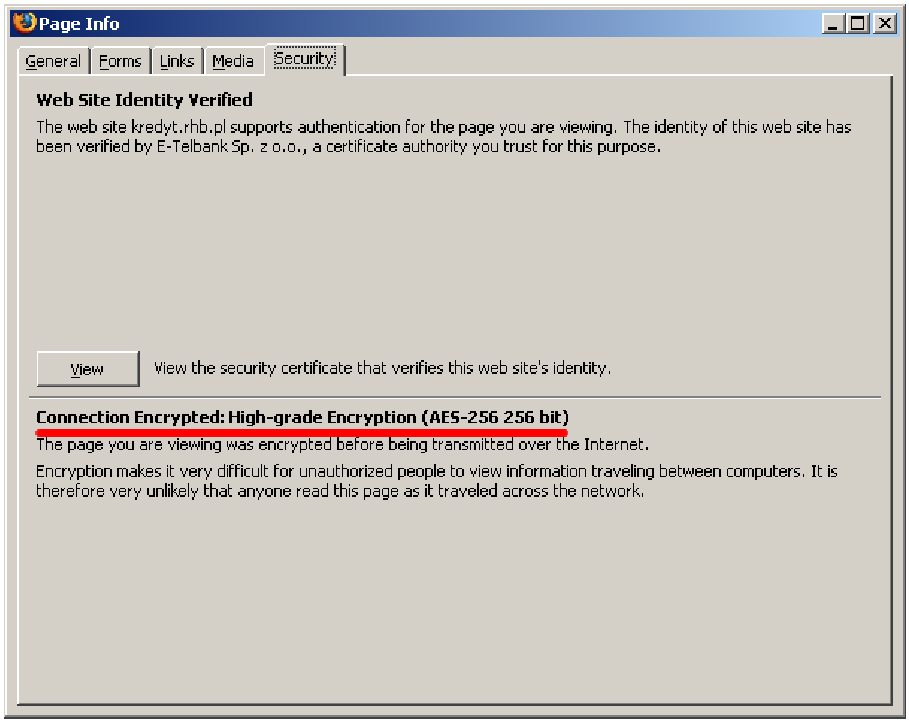
Do zilustrowania przykładu wybraliśmy BRE Bank Hipoteczny ponieważ w jego konkretnym przypadku nałożyło się kilka problemów. W standardowej sytuacji, kiedy mamy poprawnie skonfigurowaną przeglądarkę po wpisaniu w nią adresu https://kredyt.rhb.pl/ lub też przejściu za pomocą referencji ze strony głównej banku możemy zobaczyć ekran logowania tak jak na Rysunku 6. Symbol kłódki znajdujący się na końcu paska adresu oraz w dolnym pasku przeglądarki informuje nas o tym, że oglądamy stronę za pośrednictwem szyfrowanego połączenia. Możemy na tę kłódkę kliknąć, a wtedy otrzymamy bardziej szczegółowe informacje na temat tego bezpiecznego połączenia (Rysunek

7). Jak widać używany szyfr jest bezpieczny, jest to algorytm AES z kluczem o długości 256 bitów, protokół SSL w wersji 3.0. Umiejscowienie kłódki zależy od przeglądarki, której używamy. Jednak sam znak kłódki jest rozpoznawany jako sygnalizator bezpiecznego połączenia.

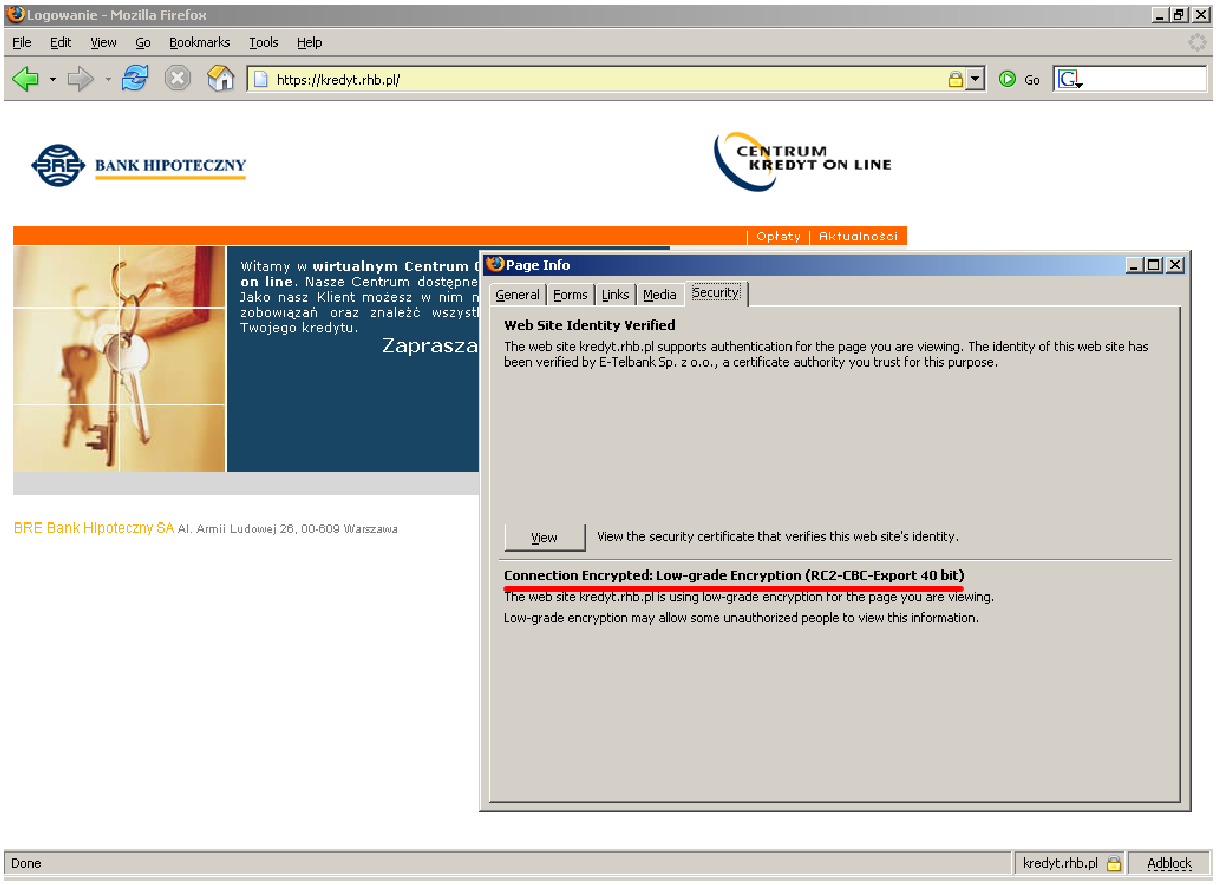
W tym czasie hipotetyczny atakujący (czyli nasz program) zostaje uruchomiony w tej samej sieci lokalnej, jego zadaniem jest przechwycić połączenie w sposób niezauważalny dla obu stron oraz wpłynąć na negocjację parametrów sesji SSL tak aby zdegradować połączenie do protokołu SSL 2.0 z użyciem słabego szyfrowania (przeprowadzić atak MITM opisany wcześniej).



Rysunek 6. Strona logowania do banku BRE Bank Hipoteczny



Rysunek 7. Informacja na temat algorytmu szyfrowania przy połączeniu do strony banku BRE Bank Hipoteczny podczas normalnego połączenia



Rysunek 8. Strona logowania się do banku BRE Bank Hipoteczny oraz informacja o stosowanym algorytmie szyfrowania po przeprowadzeniu próby ataku

Po odświeżeniu strony w swoich miejscach nadal widnieje kłódeczka. Wszystko wygląda w porządku, przeglądarka nie wyświetla ostrzeżeń. Możemy się zatem wreszcie zalogować? Nie polecam, wystarczy kliknąć ponownie na kłódeczkę, aby przekonać się że nie wszystko jest w porządku. „Bezpieczne" połączenie zostało zdegradowane do protokołu SSL 2.0 z szyfrem o kluczu o długości 40 bitów (Rysunek 8). Klucz wystarczająco łatwy do złamania aby się o to pokusić.

Co się w takim razie stało? Otóż, jak wcześniej wspomniano, nałożyło się tutaj klika problemów. Po pierwsze serwer banku BRE Bank Hipoteczny obsługuje połączenia SSL w wersji 2.0 (pierwszy błąd konfiguracyjny). Ponadto oprogramowanie serwera nie było aktualizowane do wersji nie posiadającej błędu pozwalającego na wykonanie ataku *rollback* (drugi błąd konfiguracyjny). Kolejny problem jest taki, że konfiguracja serwera umożliwia stosowanie słabego szyfrowania transmisji (trzeci błąd konfiguracyjny).

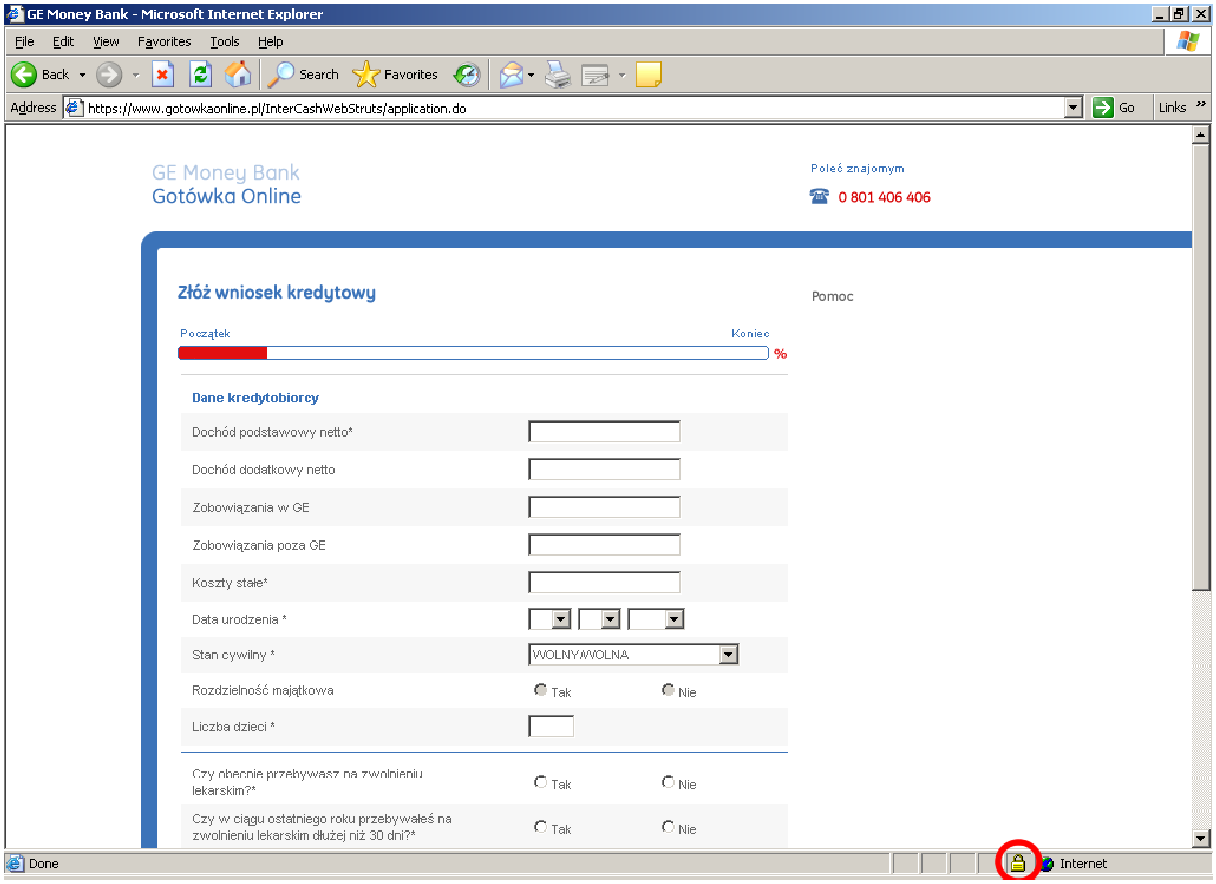
Przykładowy scenariusz realnego ataku (przykład 1)

1. Atakujący umiejscawia się tam gdzie może łatwo przeprowadzić atak. Może być to *Hot Spot* sieci bezprzewodowej w hotelu lub np. zwykła przewodowa sieć lokalna w firmie.
2. Pojawia się ofiara, która chce dokonać operacji bankowej przez Internet, np. sprawdzenia stanu konta.
3. Wpisuje w przeglądarkę internetową adres banku. Przechodzi do strony logowania. Strona wyświetla się bez problemów, klient sprawdza zgodność certyfikatu oraz algorytm szyfrowania. Ponieważ wszystko wygląda w porządku uznaje połączenie za bezpieczne.
4. W tym czasie atakujący, wpływa na dalsze połączenia klienta z portalem bankowym w sposób trudny do wykrycia (brak komunikatów przeglądarki informujących ofiarę o potencjalnym ataku). Bezpieczeństwo połączeń zostaje zdegradowane do poziomu umożliwiającego łatwe odszyfrowanie przesyłanych danych. Atakujący zbiera zaszyfrowane dane.
5. Klient zakańcza transakcję i wylogowuje się z systemu.
6. Atakujący spokojnie odchodzi i w krótkim czasie odszyfrowuje zakodowane dane, jest w posiadaniu danych identyfikacyjnych klienta oraz informacji o jego stanie konta.

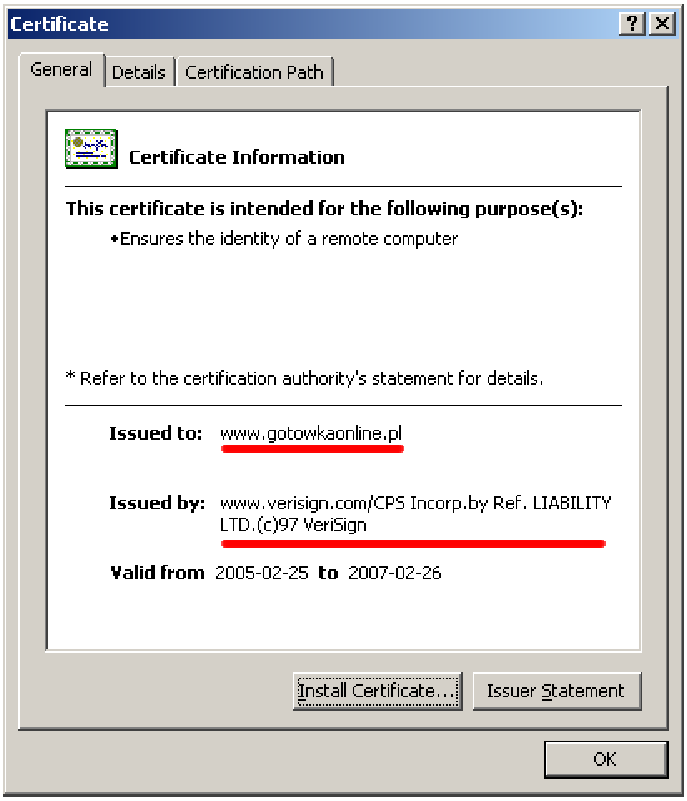
Przykład 2

Rozpatrzmy w takim razie inny przykład. Dotyczący trzeciej testowanej przez nas kwestii, czyli dostępu do serwera kanałem nie szyfrowanym. Tutaj posłużymy się stroną internetową GE Money Banku. Po wejściu na stronę http://www.gotowkaonline.pl/ zostaniemy przekierowani do połączenia szyfrowanego ze stroną, czyli https://www.gotowkaonline.pl/InterCashWebStruts/menu.do?referer=. Jeżeli interesuje nas wypełnienie wniosku, możemy wypełnić wniosek on-line. Prowadzi nas do tego link ze strony głównej. Otwiera się strona jak na Rysunku 9. Połączenie jest szyfrowane ponieważ będziemy wpisywać nasze dane osobowe, dane o zarobkach, itd. Certyfikat wystawiony przez VeriSign potwierdza wiarygodność portalu i tym samym zapewnia bezpieczeństwo naszej transakcji (Rysunek 10). Dla lepszego uwidocznienia problemu została użyta tutaj inna popularna przeglądarka – *Internet Explorer* (nota bene najpopularniejsza przeglądarka internetowa).

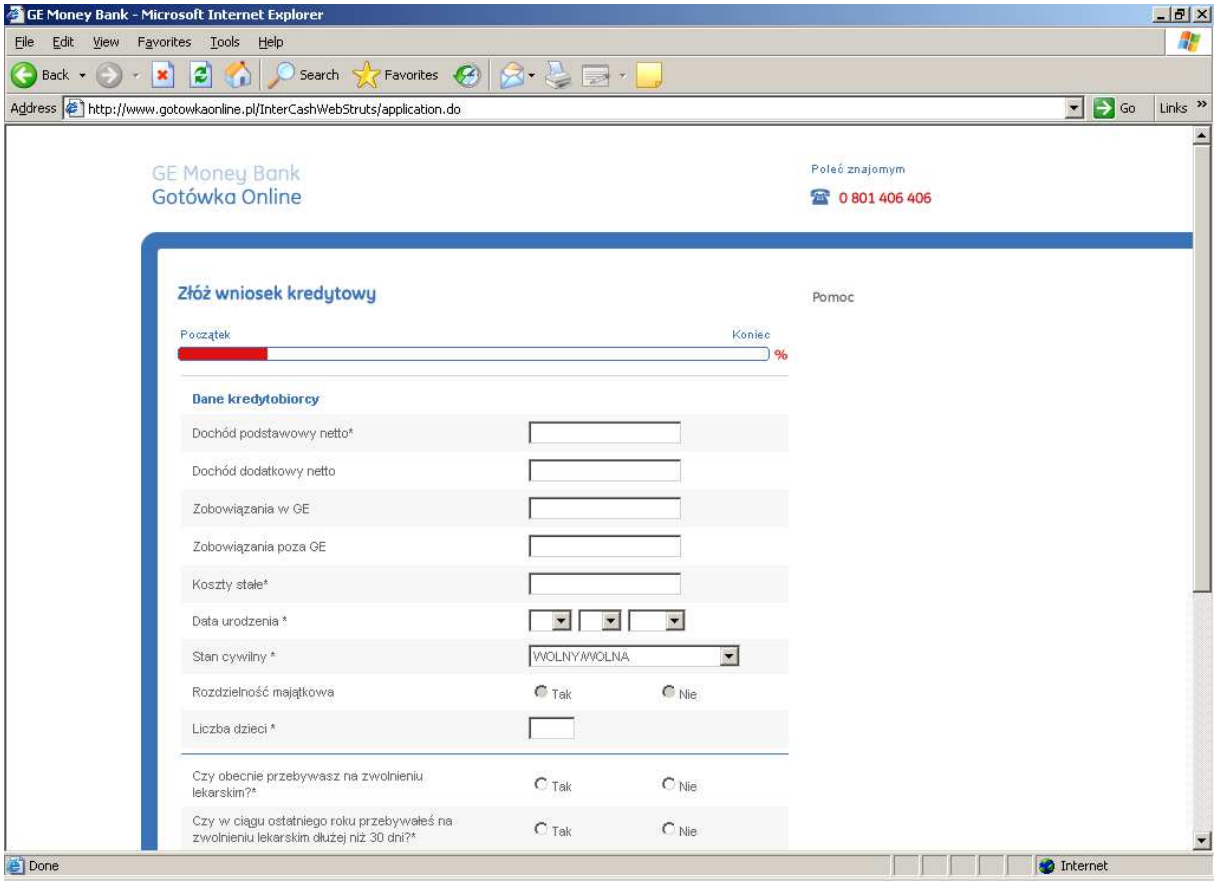
Okazuj się jednak, że jeśli do składania wniosku zostaniemy zaproszeni poprzez link http://www.gotowkaonline.pl/InterCashWebStruts/menu.do?referer=, czyli praktycznie identyczny, tylko bez 's' za http, to bezpieczeństwo będzie zapewniał co najwyżej łut szczęścia. Strona będzie wyświetlana identycznie, tyle że połączenie nie będzie szyfrowane (Rysunek 11).



Rysunek 9. Strona GE Money Bank wyświetlana z użyciem szyfrowanego połączenia



Rysunek 10. Informacje o certyfikacie służącym do zabezpieczania połączeń z GE Money Bank



Rysunek 11. Strona GE Money Bank wyświetlona z użyciem nie szyfrowanego połączenia

Jak widać, ktoś zapomniał o przekierowywaniu połączeń dla innych stron niż strona główna (zła praktyka konfiguracyjna, serwer nie powinien umożliwiać połączeń kanałem otwartym jeśli przewidziany jest jedynie do obsługi połączeń szyfrowanych). Oczywiście wniosek zostanie złożony, przeglądarka nie wyświetli żadnego komunikatu z ostrzeżeniem ponieważ technicznie wszystko działa poprawnie. Sytuacja taka może istnieć przez długi czas zanim ktoś zauważy ten problem.

Przykładowy scenariusz realnego ataku (przykład 2)

1. Ciekawski pracownik pewnej firmy pragnie poznać zarobki współpracowników. Podczas rozmowy o kredytach w gronie zainteresowanych zachwala kredyty banku X, obiecuje przesłać współpracownikom adres strony banku. „Omyłkowo" w adresie zamiast *https* występuje *http*.
2. Część współpracowników zachęcona – wedle ich mniemania – korzystną ofertą banku oraz ciepłymi słowami współpracownika decyduje się na wypełnienie formularza.
3. Wszystkie dane, które wpisują do formularzy wysyłane są otwartym nie szyfrowanym kanałem.
4. Ciekawski współpracownik triumfuje ponieważ w łatwy sposób posiadł wiedze o zarobkach oraz innych poufnych danych współpracowników.

Wnioski

Zebrane wyniki ukazują, że istnieje kilka realnych problemów. Pierwszy i najważniejszy być może wniosek jest taki, że nawet bezpieczną technologię można źle użyć. Dane pokazują, że nieco ponad połowa portali (57%) obsługuje połączenia SSL w wersji 2.0 – coś czego od 10 lat nie zaleca się stosować, jeżeli myślimy poważnie o bezpieczeństwie – jest to co najmniej zastanawiające.

Alarmującym faktem jest, że aż 46% portali bankowych (11 z 24 testowanych) posiada oprogramowanie, które prawdopodobnie, po prostu nie jest zaktualizowane. Jeżeli podobnie potraktowane zostało pozostałe oprogramowanie na serwerze (a nagłówki protokołów zwracane przez serwer wskazują, że tak), mamy prawo podejrzewać, że serwer ten ma dużo poważniejsze błędy w konfiguracji i problemy z bezpieczeństwem.

Blisko 2/3 portali umożliwia łączenie się z nimi przy użyciu słabych protokołów. Nie jest to co prawda poważny błąd sam w sobie, ale w połączeniu z innymi, jak widać na przykładzie banku BRE Bank Hipoteczny, może zaowocować poważnymi konsekwencjami.

Problem dotyczący możliwości łączenia się kanałem nie szyfrowanym, czy też problem współdzielenia domeny przez aplikację z dostępem szyfrowanym i dostępem nie szyfrowanym, dotyczy 60% portali. Fakt, że serwer przekierowuje połączenia przychodzące na port protokołu nie szyfrowanego nie jest rozwiązaniem do końca skutecznym, ponieważ w żądaniu przeglądarki, które co prawda zostanie przekierowane, ale za chwilę, mogą być przesyłane poufne dane jak np. *cookies* z identyfikatorem sesji. Dużo lepszym rozwiązaniem jest po prostu uniemożliwienie łączenia się na port protokołu nie szyfrowanego. Jest to przykład nie stosowania dobrego powszechnie znanego wzorca projektowego.

Na podstawie zebranych danych można wysunąć przypuszczenie, że blisko 2/3 portali posiada domyślną konfigurację SSL. Używane w celu realizacji bezpiecznych połączeń oprogramowanie oraz osprzęt posiada fabryczne ustawienia. Co z kolei może świadczyć o braku procesowego podejścia do bezpieczeństwa. Bezpieczeństwo nie jest produktem, który można kupić i wstawić do serwerowni.

Pojawiające się różnice w konfiguracji portali należących do tego samego banku z kolei mogą świadczyć o braku spójnej polityki zarządzania systemami informatycznymi jako elementu systemu zarządzania bezpieczeństwem.

Wiele z portali banków na swoich stronach informuje o posiadanych systemach IDS/IPS, które mają dodatkowo chronić bank i klientów. Niewątpliwie są to potrzebne w takich instytucjach systemy, ale zadziwiający jest fakt, że żaden z nich nie wykrył i nie zablokował prób degradacji

połączenia. System detekcji intruzów i zapobiegania incydentom też musi być poprawnie skonfigurowany i dostosowany do potrzeb, aby spełniał swoje zadanie. W innym wypadku jest tylko figurą marketingową.

Innym mniej formalnym wnioskiem jest teza, że testy penetracyjne oraz audyty bezpieczeństwa przeprowadzane przez niezależną stronę trzecią są niezbędne w celu utrzymania odpowiednio wysokiego poziomu bezpieczeństwa infrastruktury teleinformatycznej.

Reasumując, trudno wśród portali bankowych byłoby znaleźć serwer z konfiguracją kompletnie pozbawioną usterek, ale te kilka, które taką konfigurację posiada świadczy o tym, że jest to możliwe. Ignorowanie tych usterek może sprawić, że kiedy następnym razem zostanie opublikowany błąd w implementacji czy też założeniach projektowych protokołu podatne na atak będzie z dnia na dzień 60% serwerów. Mimo, że są to instytucje finansowe, gdzie bezpieczeństwo jest istotne, widać w większości przypadków brak podejścia procesowego, brak kompletności systemu zabezpieczeń. Widać szereg usterek, które świadczą o zbyt małym nacisku położonym na bezpieczeństwo teleinformatyczne. Poza tym jak praktyka pokazuje ilość błędów odkryta w pewnej części systemu jest wprost proporcjonalna do ilości błędów w całym systemie.

Literatura

1. SSL 2.0 Protocol Specification, http://wp.netscape.com/eng/security/SSL_2.html, 1994.
2. Eric Rescorla, SSL and TLS, Designing and Building Secure Systems, Addison-Wesley, 2000.
3. The SSL Protocol Version 3.0, http://wp.netscape.com/eng/ssl3/draft302.txt, 1999.
4. The TLS Protocol Version 1.0, ftp://ftp.rfc-editor.org/in-notes/rfc2246.txt, 1999.
5. Potential SSL 2.0 rollback, http://www.openssl.org/news/secadv_20051011.txt, 11 październik 2005.
6. Portal Twoje Finanse, http://www.twojefinanse.net/katalogi/banki.php, luty 2006.
7. ECRYPT Yearly Report on Algorithms and Keysizes (2004), http://www.ecrypt.eu.org/documents/D.SPA.10-1.1.pdf, 17 marzec 2005.
8. David Hulton, High-Speed Computing & Co-Processing with FPGAs, http://www.ccc.de/congress/2004/fahrplan/files/340-fpga-slides.pdf, grudzień 2004.
9. OWASP Web Application Penetration Checklist, http://www.owasp.org/documentation/testing.html, lipiec 2004.
10. Gunterl Ollman, Security Best Practice: Host Naming & URL Conventions, http://www.ngssoftware.com/papers/NISR-BestPracticesInHostURLNaming.pdf, luty 2005.

Szczegółowa lista portali bankowych

Celem Zespołu Bezpieczeństwa nie było tworzenie rankingu banków tylko uświadomienie problemu, stąd też banki nie zostały w żaden sposób posortowane, lista która znajduje się poniżej jest w kolejności przypadkowej, a co najwyżej alfabetycznej. Generalnie wartość 0 w komórce tabeli oznacza brak danej słabości, wartość niezerowa występowanie podatności. Poszczególne kolumny oznaczają:

- **Obsługa SSL 2.0** – wartość 1 w tym polu informuje o tym czy serwer wspiera połączenia SSL w wersji 2.0,
- **Obsługa słabych szyfrów** – wartość 1 w tym polu oznacza, że serwer umożliwia łączenie się z wykorzystaniem słabych protokołów bez rozgraniczenia na wersje SSL,
- **Obsługa słabych szyfrów dla SSL 2.0** – tutaj wartość 1 figuruje dla tych serwerów, dla których jest możliwość połączenia się protokołem SSL w wersji 2.0 oraz z wykorzystaniem słabego szyfrowania,
- **Możliwość degradacji połączenia do SSL 2.0** – wartość 1 oznacza, że przeprowadzona została symulacja ataku polegająca na degradacji połączenia do protokołu SSL w wersji 2.0 i że zakończyła się ona pomyślnie (dla strony atakującej),
- **Możliwość nawiązywania połączeń nie szyfrowanych** – 0 oznacza, że połączenie z serwerem kanałem nie szyfrowanym jest niemożliwe, 1 że następuje przekierowanie do kanału szyfrowanego, natomiast 2, że można przesyłać dane do serwera korzystając z połączenia nie szyfrowanego.

URL	Obsługa SSL 2.0	Obsługa słabych szyfrów	Obsługa słabych szyfrów dla SSL 2.0	Możliwość degradacji połączenia do SSL 2.0	Możliwość nawiązywania połączeń nie szyfrowanych
https://www.bph.pl/	0	0	0	0	1
https://www.bgk24biznes.pl/	0	1	0	0	0
https://home6poland.cd.citibank.pl/	0	1	0	0	1
https://www.bmbise.pl/	1	1	1	0	1
https://bosbank24.pl/	0	1	0	0	1
https://www.pekao24.pl/	1	1	1	0	1
https://www.bankpocztowy.pl/	1	1	1	0	1
https://bank.cui.pl/	1	1	1	0	0
https://klucz.bwe.pl/	1	1	1	1	1
https://www.centrum24.pl/	0	1	0	0	1
https://www.integrum.pl/	0	0	0	0	1
https://makler.bmbgz.pl/	1	1	1	1	0
https://www.ibre.com.pl/	0	1	0	0	1

https://bi.bresa.com.pl/	1	1	1	1	0
https://www.brebank.pl/	0	1	0	0	2
https://www.e-dominet.pl/	0	0	0	0	1
https://ebank.db-pbc.pl/	1	0	0	1	0
https://ebusinessbank.db-pbc.pl/	1	0	0	1	0
https://planet.fortisbanking.com.pl/	0	1	0	0	0
https://korporacja.gb24.pl/	0	1	0	0	0
https://www.gotowkaonline.pl/	1	1	1	1	2
https://ssl.bsk.com.pl/	1	1	1	0	1
https://uslugi.ingnn.pl/	0	1	0	0	0
https://secure.inteligo.com.pl/	1	0	0	1	1
https://www.investkonto.pl/	1	0	0	0	1
https://www.investbank24.pl/	1	0	0	0	1
https://www.kb24.pl/	0	0	0	0	1
https://kredyt.rhb.pl/	1	1	1	1	0
https://e-bank.lukas.com.pl/	1	0	0	1	0
https://bank.lukas.com.pl/	1	1	1	0	0
https://www.mbank.com.pl/	1	1	1	0	1
https://moj.multibank.pl/	1	1	1	0	1
https://www.millenet.pl/	0	0	0	0	1
https://www.managerland.pl/	0	0	0	0	0
https://www.nordeasolo.pl/	1	1	1	0	0
https://www.pkobp.pl/	1	1	1	0	1
https://www.pkointeligo.pl/	1	0	0	1	1
https://www.r-bank.pl/	1	1	0	1	1
https://login.vwbankdirect.pl/	0	0	0	0	0
https://biznesbanking.vwbankdirect.pl/	0	0	0	0	0
https://www.ofi.pl/	1	1	1	0	1