Michał KONIUSZKO
Państwowa Wyższa Szkoła Techniczno-Ekonomiczna w Jarosławiu, Polska

Robert PĘKALA
Uniwersytet Rzeszowski, Polska

Andrzej PASZKIEWICZ
Politechnika Rzeszowska, Polska

Wspomaganie certyfikacji witryn internetowych za pomocą urzędów lokalnych

Wstęp

Obecnie temat bezpieczeństwa danych dostępnych w sieciach i intersieciach, a zwłaszcza w intersieci globalnej (internecie), jest bardzo istotny. Dane przesyłane połączeniami sieciowymi lub intersieciowymi są w wielu przypadkach poufne, a osoby trzecie nie powinny mieć do nich dostępu.

Jednym z filarów bezpieczeństwa informatycznego jest protokół SSL funkcjonujący na bazie tzw. certyfikatów, które wydawane są przez urzędy certyfikacji. Przedsiębiorstwa i instytucje dbające o bezpieczeństwo swoich danych, również wewnątrz firmy, często używają własnych urzędów certyfikacji (*certification authority* – CA). Umożliwia to połączenie z siecią firmową bądź dostęp do zasobów przedsiębiorstwa z jednoczesnym podniesieniem poziomu bezpieczeństwa.

Zagłębiając się w technologię SSL oraz tematykę lokalnych urzędów certyfikacji, można dojść do wniosku, że tworzenie takich certyfikatów w pewnych przypadkach jest dosyć uciążliwe. O ile w systemach serwerowych firmy Microsoft pewnym ułatwieniem jest interfejs graficzny GUI (*Graphical User Interface*), to w przypadku platform linuksowych jesteśmy zwykle zdani na wykonywanie skomplikowanych i złożonych komend za pomocą odpowiednich plików z poziomu wiersza linii poleceń (*command-line interface* – CLI). Dodatkowym utrudnieniem jest docelowa powłoka OPENSSL, w której niestety nie występuje znane podpowiadanie składni czy możliwość powrotu do poprzednich poleceń za pomocą strzałek nawigujących. Nie umożliwia ona również ustawienia pozycji kursora w dowolnym miejscu polecenia bez kasowania wcześniej napisanego tekstu.

W niniejszym artykule zaprezentowano podejście, w którym proponuje się kompleksowe rozwiązanie wspomnianego powyżej problemu poprzez wykorzystanie utworzonej aplikacji internetowej umożliwiającej w łatwy i szybki sposób utworzenie lokalnego CA oraz podpisywanie nim certyfikatów w standardzie X.509 niezależnie od platformy systemowej, na której pracujemy. Aplikacja

znacząco ułatwia wdrażanie polityki bezpieczeństwa sieciowego w zakresie zabezpieczania firmowych witryn internetowych poprzez wykorzystanie kryptografii implementowanej przez bibliotekę OPEN SSL. Do jej budowy wykorzystano język PHP, SQL, graficzny interfejs użytkownika powstał przy użyciu HTML5, Java Script oraz CSS3. Aplikację wdrożono na serwerze HTTP – Apache2 zainstalowanym na platformie Ubuntu 14.04 LTS [Gajda 2007; Lis 2007]. Aplikacja dostępna jest pod adresem: http://auth-center.ddns.net.

Certyfikaty standardu X.509

Technologia SSL wykorzystuje rozwiązanie znane jako Infrastruktura Klucza Publicznego (*Public Key Infrastructure* – PKI). Trzon PKI stanowi sieć zaufanych urzędów certyfikacji, które są instytucjami wystawiającymi certyfikaty oraz certyfikującymi inne CA. Oprócz tego w ramach PKI funkcjonują urzędy rejestracji (*registration authority* – RA), które zbierają wnioski o wydanie certyfikatu, weryfikując przy tym tożsamość wnioskodawcy według ustalonych reguł. Mechanizm działania PKI opiera się na zaufaniu między wszystkimi stronami procesu. Związane jest to z udziałem tzw. zaufanej strony trzeciej, którą jest np. lokalne centrum autoryzacji. Oznacza to, że wszystkie certyfikaty wystawione przez dany organ muszą być akceptowane przez wszystkie podmioty ufające danemu wystawcy.

Z technicznego punktu widzenia niezwykle istotny jest standard o nazwie X.509, który m.in. definiuje strukturę kluczy oraz mechanizmy ich wykorzystania w ramach certyfikatów wydawanych przez CA. Protokół SSL jest jednym z protokołów, który wspiera X.509. Certyfikaty zgodne z X.509 są jednym z najczęściej stosowanych sposobów autoryzacji użytkowników, serwerów czy usług internetowych, a także są wykorzystywane w technologii podpisów cyfrowych. Jako konkretna, określona struktura informacyjna zdefiniowane są w dokumencie RFC 6101 [https://tools.ietf.org/html/rfc6101].

Procedury generowania certyfikatu za pomocą CLI

Rozważmy mechanizmy tworzenia CA na platformie Ubuntu za pomocą konsoli OPENSSL. Procedurę należy rozpocząć od wygenerowania klucza prywatnego np. za pomocą algorytmu RSA o zadanej długości. W katalogu, w którym tworzony będzie CA, muszą znajdować się foldery „demoCA”, w nim pusty plik o nazwie „index.txt”, plik „serial”, w którym zapisana jest dowolna liczba większa od zera oraz pusty folder o nazwie „newcerts”. Nowy klucz, np. o długości 2048b, generowany jest za pomocą polecenia *genrsa*:

```
OpenSSL> genrsa -des3 -out ca.pem 2048
```

gdzie: *des3* określa algorytm szyfrujący DES3, *out* definiuje ścieżkę i nazwę generowanego pliku. Liczba podawana na końcu polecenia określa długość klu-

cza w bitach. W trakcie tworzenia klucza należy podać oraz zweryfikować hasło zabezpieczające.

Kolejnym krokiem jest utworzenie głównego certyfikatu CA (klucza publicznego) w standardzie X.509 przy użyciu wcześniej wygenerowanego klucza. Jest to tzw. *Self Signed Certificate*, gdyż jest podpisany swoim własnym kluczem prywatnym. Do jego utworzenia można posłużyć się następującym poleceniem:

```
OpenSSL> req -new -x509 -days 365 -key ca.pem -out ca.cer
Enter pass phrase for ca.pem:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
OpenSSL>
```

gdzie: *new* określa nowy certyfikat, *x509* definiuje standard certyfikatu, *days* – określa okres jego ważności, *key* określa ścieżkę dostępu do pliku klucza prywatnego, natomiast *out* definiuje nazwę oraz lokalizację generowanego pliku certyfikatu. Jak widać, wykonanie polecenia wymaga podania hasła użytego wcześniej do zabezpieczenia klucza prywatnego, a następnie wypełnienia następujących danych: Country Name (2 letter code) [AU] – dwuliterowy kod państwa, State or Province Name (full name) [Some-State] – nazwa stanu, Locality Name (eg, city) – nazwa miejscowości, Organization Name (eg, company) [Internet Widgits Pty Ltd] – nazwa organizacji, Organizational Unit Name (eg, section) – nazwa jednostki organizacji, Common Name (e.g. server FQDN or YOUR name) – nazwa zwyczajową (przeważnie jest to domena, dla której tworzony jest certyfikaty), Email Address [] – adres e-mail.

Poprawne wypełnienie formularza DN (*distinguished names*) skutkuje utworzeniem w określonej lokalizacji pliku certyfikatu o zadanej nazwie. Utworzona w ten sposób para klucza prywatnego i certyfikatu jest docelowym lokalnym urzędem certyfikacji.

Posiadając utworzony CA, można wygenerować ostatecznie certyfikat docelowy. Procedurę należy zacząć od utworzenia klucza prywatnego, np. RSA o długości 2048b:

```
OpenSSL> genrsa -des3 -out client.pem 2048
```

Następnym krokiem jest utworzenie tzw. żądania podpisania certyfikatu (*certificate sign request* – CSR). Etap ten został pominięty przy tworzeniu lokalnego CA ze względu na to, iż tworzony certyfikat urzędu certyfikacji był certyfikatem *self signed*.

Do tego celu służy polecenie: *req*, np.:

```
OpenSSL> req -new -key client.pem -out cerreq.csr
```

gdzie: *new* definiuje nowy certyfikat, *key* określa położenie utworzonego wcześniej pliku klucza prywatnego, *out* – położenie oraz nazwę pliku żądania podpisania certyfikatu. Posiadając plik żądania, można wygenerować certyfikat, podpisując go za pomocą certyfikatu oraz klucza prywatnego CA. Dedykowane polecenie może wyglądać jak niżej:

```
OpenSSL> ca -policy policy_anything -cert ca.cer -in cerreq.csr -keyfile ca.pem
- days 365 -out client.cer
```

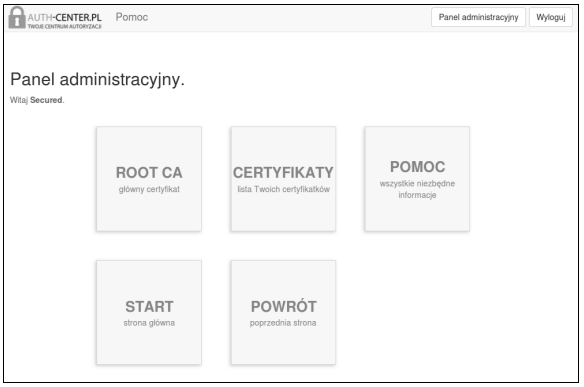
gdzie: *policy* określa wymagane pola wypełnianego *CSR*, definicja *policy_ anything* określa wszystkie pola DN opcjonalne oprócz commonName, które jest zawsze wymagane, *cert* określa położenie pliku certyfikatu CA, *in* określa lokalizację pliku CSR, *keyfile* wyznacza ścieżkę pliku klucza prywatnego lokalnego urzędu certyfikacji, *days* definiuje okres ważności, *out* określa nazwę oraz położenie generowanego pliku certyfikatu.

Z powyższego przykładu obrazującego mechanizmy tworzenia lokalnego CA oraz podpisania nim nowego certyfikatu można wywnioskować, że proces tworzenia i podpisywania certyfikatów wymaga stosowania skomplikowanych, złożonych poleceń, co może stwarzać wiele problemów. Dodatkowym utrudnieniem jest konieczność poznania nowej, unikalnej powłoki CLI OPENSSL.

Aplikacja wspomagająca certyfikację witryn internetowych

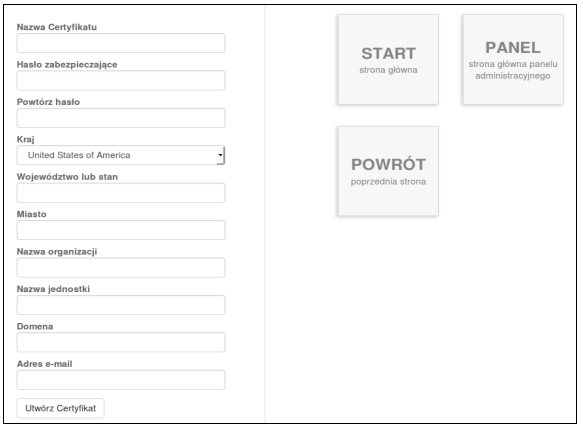
Biorąc pod uwagę niedogodności związane z generowaniem certyfikatów w trybie CLI, zaproponowano aplikację internetową wspomagającą to zadanie. Jej zadaniem jest ułatwienie procesu tworzenia lokalnego centrum autoryzacji oraz wystawiania podpisanych certyfikatów w standardzie X.509 dedykowanych usłudze *www*. Urząd certyfikacji w momencie tworzenia będzie przypisany tworzącemu go użytkownikowi i przechowywany wraz z wystawianymi certyfikatami w bezpieczny sposób w bazie danych. W każdym momencie użytkownik będzie miał dostęp do plików certyfikatów oraz kluczy prywatnych, będzie mógł je pobrać z bazy danych i zapisać na dysku po wcześniejszej autoryzacji. W celu zabezpieczenia oraz identyfikacji użytkowników aplikacja wymaga rejestracji. Aby się zarejestrować, użytkownik musi podać unikalną nazwę, adres e-mail oraz hasło. Dane rejestracyjne użytkownika są przechowywane w bazie danych. Dodatkowym zabezpieczeniem jest przechowywanie hasła w formie zaszyfrowanej. Autoryzacja dostępu do konta odbywa się przez podanie pary: nazwa

użytkownika oraz hasło, która porównywana jest z danymi zapisanymi w bazie danych. W przypadku pomyślnej autoryzacji ustanawiana jest sesja użytkownika oraz przyznawany dostęp do panelu administracyjnego (rys. 1), który pozwala na korzystanie z wszystkich funkcjonalności aplikacji.



Rys. 1. Wygląd panelu administracyjnego aplikacji

Z poziomu panelu użytkownik ma dostęp do wszystkich funkcji aplikacji, tj. utworzenia urzędu certyfikacji oraz tworzenia, przeglądania, pobierania i kasowania swoich plików certyfikatów i kluczy. Na rys. 2 przedstawiono okno dla opcji tworzenia CA.



Rys. 2. Tworzenie urzędu certyfikacji

Utworzenie urzędu certyfikacji wymaga wypełnienia formularza, którego pola pokrywają się formularzem DN, jak to pokazano wcześniej. Funkcjonalność aplikacji pozwalająca na tworzenie i podpisywanie certyfikatów bazuje na bibliotece PHP – OpenSSL, która implementuje metody kryptograficzne oraz

wiele funkcji OpenSSL [http://php.net/manual/en]. Umożliwia także modyfikacje konfiguracji OpenSSL na potrzeby danej funkcji. Na potrzeby aplikacji zostały utworzone specjalne klasy implementujące funkcje biblioteki OpenSSL odpowiedzialne za poprawne przeprowadzenie procesu tworzenia certyfikatów.

Podsumowanie

Analiza protokołu SSL oraz zrozumienie mechanizmów jego działania z wykorzystaniem certyfikatów w standardzie X.509 pozwoliło na stworzenie internetowej aplikacji omówionej w niniejszym artykule. Dzięki niej skomplikowany proces tworzenia certyfikatów, kluczy RSA oraz podpisywanie certyfikatów z wykorzystaniem technologii OpenSSL został sprowadzony do uzupełnienia odpowiednich formularzy w przeglądarce internetowej. Technologie wykorzystane do budowy aplikacji zapewniają bezpieczeństwo i stabilność, ale przede wszystkim dostępność niezależnie od platformy programowej. Wykorzystany do utworzenia aplikacji Framework Symfony2 dodatkowo podnosi poziom bezpieczeństwa i przyśpiesza działanie.

Literatura

Gajda W. (2007): *HTML, XHTML i CSS. Praktyczne projekty*, Gliwice.
http://symfony.com/doc/current/book/index.html.
http://php.net/manual/en.
https://tools.ietf.org/html/rfc6101.
Lis M. (2007): *Ćwiczenia praktyczne. JavaScript*, Gliwice.

Streszczenie

W artykule przedstawiono zagadnienia związane z problematyką technologii SSL wykorzystującej certyfikaty. Zaproponowano aplikację, która w znaczący sposób ułatwia wdrażanie technologii SSL. Aplikacja ta posiada wszystkie niezbędne funkcjonalności pozwalające na zarządzanie kluczami i certyfikatami w połączeniu z technologiami bazodanowymi.

Słowa kluczowe: certyfikat SSL, urząd certyfikacji, OpenSSL.

Certification Support of Internet Websites with Using Local Authorities

Abstract

The article presents issues related to the SSL technology involving certificates. The appropriate internet application has been proposed, which significantly simplifies the deployment of SSL technology. The application has all the necessary functionality to manage keys and certificates, in conjunction with database-technologies.

Keywords: SSL certificate, local authorities, OpenSSL.