1. Mechanizmy bezpieczeństwa w systemie Comcute

Piotr Szpryngier

Politechnika Gdańska,

Wydział Elektroniki, Telekomunikacji i Informatyki,

Katedra Architektury Systemów Komputerowych

e-mail: piotrs@eti.pg.gda.pl

Streszczenie

Celem niniejszego dokumentu jest wyróżnienie podstawowych problemów związanych z bezpieczeństwem przetwarzania w systemie utrzymania wielkiej mocy obliczeniowej w sytuacjach kryzysowych Comcute. Ponadto na przykładzie architektury systemu modelowego będą przedstawione mechanizmy bezpieczeństwa przydatne do zastosowania w projekcie.

Słowa kluczowe: polityka bezpieczeństwa, poufność, wiarygodność, PKI, certyfikat klucza publicznego.

1.1. Założenia i wymagania

System laboratoryjny służy do zbadania możliwości rozproszenia obliczeń wymagających dużej mocy na komputery zlokalizowane w wewnętrznej sieci lokalnej (laboratorium) lub w Internecie. Model systemu laboratoryjnego (wg zamierzeń projektantów) może stać się prototypem systemu docelowego. Z tego względu mechanizmy bezpieczeństwa odnoszące się do prototypu mają także takie same znaczenie jak w przypadku systemu docelowego.

Jednakże niezbędne jest poczynienie jednej zasadniczej uwagi. O ile system laboratoryjny pracuje w środowisku zamkniętym (symulując mechanizmy działania systemu docelowego), to system docelowy na różnych poziomach będzie dostępny poprzez Internet. Ponadto system docelowy zapewne będzie przechowywał informacje pochodzące od swoich klientów. Z tego względu jest niezbędne zaplanowanie, opracowanie i wdrożenie polityki bezpieczeństwa. Polityka bezpieczeństwa w postaci dokumentu zatwierdzonego przez zarząd organizacji (właściciela systemu docelowego) powinna być udostępniania klientom jako pewnego rodzaju gwarancja jakości właściwego (oczekiwanego, zgodnego z przepisami i normami) postępowania z danymi klientów. Taki dokument powinien zawierać następujące elementy [1,4,6, 9, 10]:

1. Postanowienia ogólne, podstawy prawne, słownik pojęć;
2. Zasady identyfikacji i uwierzytelniania;

3. Zakres świadczonych usług;
4. Opis zasobów organizacji udostępnianych w ramach systemu obliczeniowego;
5. Analiza ryzyka względem kosztów zabezpieczeń i spodziewanych korzyści;
6. Wymagania operacyjne bezpieczeństwa (czynności stron, protokoły komunikacji, forma dostarczenia usługi, itp.);
7. Środki bezpieczeństwa dla spełnienia wymagań operacyjnych;
8. Zarządzanie bezpieczeństwem (kontrole, audyty, zmiany), polityki względem personelu i klientów.

Przyjmijmy założenie, że w systemie [2] będą pracować następujące rodzaje użytkowników:

- administratorzy konfiguracji;
- administratorzy bezpieczeństwa – zarządcy uprawnień, ról, itp.;
- operatorzy systemu (konfiguracja zadań od klientów, nadzór nad przebiegiem obliczeń);
- klienci – zleceniodawcy obliczeń.

W zależności od liczby użytkowników należy rozważyć różne modele kontroli dostępu. Dla małej liczby (kilkaset) można rozważyć wprowadzenie modelu restrykcyjnego (np. BLP – MAC+NTK) [6] lub łagodniejszego (DAC z separacją obiektów) [5,6] . Dla dużej liczby użytkowników (>1000) kosztowny we wprowadzeniu, ale łatwy w zarządzaniu jest model RBAC [5].

1.2. Zarządzanie bezpieczeństwem informacji

Ogólny model warstwowy zarządzania bezpieczeństwem można przedstawiono w tab. 1.1.

Tab. 1.1. Warstwy ogólnego systemu zarządzania bezpieczeństwem

Zarządzanie bezpieczeństwem (aplikacje, bazy danych, EDE, e-mail, itp.)
Agenci bezpieczeństwa, Protokoły bezpieczeństwa (uwierzytelnianie, zarządzanie kluczami, itp)
Usługi bezpieczeństwa (poufność, integralność, niezaprzeczalność, itp.)
Mechanizmy bezpieczeństwa (podpis cyfrowy, uwierzytelnianie)
Moduły podstawowe (algorytmy, tryby pracy)

W ramach niniejszego rozdziału przedyskutowano elementy z warstwy 3 (usługi bezpieczeństwa) wraz z mechanizmami (warstwa 2 – powszechnie znane i opracowane) i częściowo protokoły bezpieczeństwa (warstwa 4). Elementy zarządzania bezpieczeństwem muszą być powiązane z polityką bezpieczeństwa (której tu nie opisujemy), natomiast podstawowe moduły

bezpieczeństwa (algorytmy, tryby pracy, generatory kluczy, itp.) są powszechnie znane i opracowane [3,4,5,6]. Objaśnimy tu jedynie dwa podstawowe protokoły służące do budowy podstawowych mechanizmów bezpieczeństwa. Pierwszy z nich to protokół przesyłania wiadomości zaszyfrowanej łącznie z sesyjnym kluczem szyfrującym [3,4,5,6]. Celem tego protokołu jest zapewnienie poufności treści przesyłanej wiadomości, a także wiarygodności jej odbiorcy. Przebiega on następująco:

A) Nadawca przygotowuje wiadomość M, losuje klucz sesyjny K i tworzy szyfrogram $C_K(M)$ z użyciem dobrej jakości symetrycznego algorytmu szyfrującego.

B) Nadawca pobiera z zaufanej bazy danych klucz publiczny odbiorcy O_{pu} bądź wydobywa go z certyfikatu podpisanego przez zaufanego wystawcę, a następnie szyfruje klucz sesyjny, używając tego samego symetrycznego algorytmu szyfrującego co w p. A pracującego w trybie ECB [3]. Powstaje dodatkowy szyfrogram $C_{Opu}(K)$, który dołącza do szyfrogramu $C_K(M)$ i wysyła to wszystko odbiorcy.

C) Odbiorca najpierw odszyfrowuje klucz sesyjny K z szyfrogramu, używając do odszyfrowania klucz prywatnego O_{pr}, występującym w parze z kluczem publicznym O_{pu} i otrzymuje $K=D_{Opr}(C_{Opu}(K))$.

D) Następnie odbiorca odszyfrowuje wiadomość M, używając uzyskanego w p. C klucza sesyjnego: $M=D_K(C_K(M))$

Drugim podstawowym protokołem jest podpis elektroniczny, będący tzw. załącznikiem wiadomości. Ma on następujący przebieg [4,11]:

A) Nadawca przygotowuje wiadomość M, oblicza skrót wiadomości $H(M)$, a następnie, używając określonego schematu podpisu elektronicznego i prywatnego klucza podpisującego K_S, oblicza podpis $S_{Ks}(H(M))$.

B) Nadawca pobiera wiadomość M, dołącza do niej podpis $S(H(M))$ i wysyła to wszystko odbiorcy.

C) Nadawca pobiera z zaufanej bazy danych klucz publiczny nadawcy K_{Spu} bądź wydobywa go z certyfikatu podpisanego przez zaufanego wystawcę, a następnie wylicza wartość skrótu otrzymanej wiadomości $H(M')$ i używa obu tych wielkości w procedurze weryfikacji poprawności podpisu elektronicznego.

D) Jeśli podpis jest poprawny (np. dla schematu RSA skrót wiadomości odebranej $H(M')$ jest zgodny ze skrótem $H(M)$ odszyfrowanym z podpisu z użyciem klucza publicznego nadawcy K_{Spu}), to akceptuje wiadomość. W przeciwnym wypadku wiadomość jest odrzucana jako pochodząca z niewiadomego źródła.

Infrastruktura klucza publicznego PKI [4,11] to sieć serwerów wystawiających certyfikaty CA (*Certification Authority*), serwerów rejestrujących użytkowników RA (*Registration Authority*) wraz z polityką bezpieczeństwa określającą działania i procedury związane z zarządzaniem certyfikatami kluczy publicznych. Certyfikat klucza publicznego to struktura danych podpisana przez wystawcę certyfikatu. Te podpisane dane zawierają co najmniej dane

ubiegającego się o certyfikat (a zatem posiadacza klucza prywatnego do pary z tym umieszczonym w certyfikacie), jego klucza publicznego, identyfikatora certyfikatu, danych i podpisu wystawcy. Certyfikat może ponadto zawierać wiele dodatkowych danych: datę wystawienia i okres ważności, przeznaczenie certyfikatu (kluczy), wersję standardu, ograniczenia stosowania, delegacje użycia, itp. Niezwykle istotne jest stwierdzenie, że certyfikat klucza publicznego de facto bezwzględnie łączy ze sobą użytkownika z jego kluczem prywatnym, którego oczywiście ujawnić nie można. Dzięki temu, wykorzystując klucz publiczny zawarty w certyfikacie, możemy zweryfikować autentyczność wszelkich czynności wykonanych przez użytkownika z użyciem jego klucza prywatnego (do pary z tym opublikowanym w certyfikacie) [11].

1.3. Bezpieczeństwo interfejsu: klient zewnętrzny – system Comcute

W tym punkcie będą opisane aspekty bezpieczeństwa interfejsu klienta – czyli warstwy komunikacji warstwy *W* ze światem zewnętrznym [2]. Zgodnie z dokumentem pt. „Architektura systemu – wymagania od strony klienta", ten interfejs powinien pozwolić na realizację następujących czynności:

- zdefiniowanie problemu wraz z niezbędnymi danymi i określeniem parametrów,
- zdefiniowanie kodów programów niezbędnych do rozproszonego wykonania obliczeń w systemie Comcute,
- odczytanie wyników i statusu wykonywanego zadania (kodu poprawności),
- ewentualnie odczytanie informacji o kompletności wyników.

Typowe zagrożenia pojawiające się w tym obszarze systemu to [3,5,6]:

- podsłuch komunikacji – naruszenie poufności (bierny atak),
- zniszczenie, modyfikacja komunikacji, podszywanie się, itp. (ataki czynne),
- analiza ruchu.

Mamy co najmniej trzy alternatywne sposoby komunikacji klienta z systemem Comcute:

a. poprzez klasyczny interfejs użytkownika – formularz na stronie WWW systemu,
b. jako usługa sieciowa, opublikowana i dostępna jedynie dla wybranych klientów,
c. inny sposób komunikacji – dowolnym bezpiecznym kanałem (osobiście, poczta klasyczna, itp.).

W zależności od sposobu komunikacji możemy wskazać tu następujące mechanizmy niezbędne do zachowania minimum bezpieczeństwa:

a. w przypadku interfejsu www – tunel kryptograficzny SSL/TLS [5] oraz silne uwierzytelnianie (oprócz loginu i hasła dodatkowo np. protokół ZK – *Zero Knowledge* z wykorzystaniem kart inteligentnych) [3]. Ponadto warto zastanowić się nad dodatkowymi restrykcjami w zakresie przyjmowania ruchu sieciowego poprzez taką konfigurację zapory ogniowej,

by akceptować połączenia tylko z grupą wybranych adresów IP. SSL i zapora ogniowa chronią przed podsłuchem i modyfikacją komunikacji, ale nie chronią przed analizą ruchu.

b. W przypadku usługi sieciowej – jej opublikowanie podlega restrykcjom zawartym w opisie standardu OASIS opisującym cechy i wymagania dla rejestrów UDDI [7], a udostępnianie usługi muszą być chronione podobnie jak w p. a. Nie chroni to niestety przed analizą ruchu.
c. Inny sposób komunikacji (np. osobiście) może zapewnić ochronę przed analizą ruchu. Dodatkowo usługę obliczeniową będzie konfigurował i wdrażał operator – pracownik właściciela systemu Comcute. Może on uruchamiać wiele obliczeń i dla każdego z nich dobierać różne opóźnienie czasowe chwili wystartowania, więc nie będzie dokładnie wiadomo, dla kogo te obliczenia są wykonywane. W tej sytuacji mamy także ochronę przed modyfikacją i podsłuchem za pomocą tych samych środków (SSL, uwierzytelnianie operatora).

1.4. Bezpieczeństwo warstwy serwerów *W*

W tym punkcie opisano aspekty bezpieczeństwa warstwy serwerów *W* (węzłów wewnętrznych systemu) zajmującej się zarządzaniem zadaniami obliczeniowymi zleconymi przez klienta zewnętrznego bądź operatora. Zgodnie z dokumentem pt. „Architektura systemu – problemy i koncepcje" [2], w tej warstwie są realizowane następujące czynności:

- przyjmowanie zlecenia w postaci kodu i danych klienta oraz dodatkowych wymagań obliczeniowych,
- partycjonowanie danych,
- wyznaczenie kryteriów wykonalności (zakończenia) zadania,
- ustanowienie ewentualnego arbitra oceniającego postępy w obliczeniach oraz rozstrzygającego o ważności wyników,
- ewentualne przekształcenie kodu zleceniodawcy do postaci wykonywalnej akceptowalnej przez aplikacje internautów,
- gromadzenie wyników obliczeń dla każdego zadania.

Z powyższego zestawienia wynika, że warstwa serwerów *W* przechowuje kluczowe dane z punktu widzenia klienta zewnętrznego. Przyjmijmy dodatkowe założenie, że może być więcej grup serwerów *W* niż jedna, umiejscowionych w rozłącznych sieciach lokalnych. Każda taka grupa może współpracować np. z wybraną grupą serwerów S. Mogące się zmaterializować zagrożenia to: naruszenie spójności przechowywanych danych (nieuprawniona zmiana, zniszczenie), podsłuch (podgląd wyników oraz informacji, dla kogo te obliczenia są wykonywane). Z tego względu ochrona tych danych jest bardzo ważna i należy ją zaplanować na kilku poziomach:

- Sieć lokalna, zawierająca serwery grupy *W* powinna być chroniona zaporą ogniową tak skonfigurowaną, że będzie akceptować połączenia zewnętrzne tylko ze ściśle zdefiniowanymi adresami IP.
- Wszystkie połączenie pomiędzy serwerami *W* są realizowane w tunelu SSL z obustronnym uwierzytelnianiem.

- Powinny być zapewnione mechanizmy transakcyjne w dostępie do współdzielonej bazy danych albo powinna być zapewniona pełna replikacja (kopie bazy danych w każdym węźle *W*), w miarę możliwości synchroniczna [8].
- Dostęp do baz danych ograniczony tylko dla lokalnych aplikacji i użytkowników serwerów *W* i ewentualnie także serwerów *S* systemu laboratoryjnego.
- Dostęp do serwerów *W* powinien być ograniczony jedynie dla uwierzytelnionych i autoryzowanych użytkowników.

1.5. Bezpieczeństwo komunikacji pomiędzy warstwą serwerów *W* a serwerami *S*

W tym punkcie zostały opisane aspekty bezpieczeństwa komunikacji warstwy *W* z serwerami *S* związane z dystrybucją podzadania obliczeniowego. Zgodnie z dokumentem pt. „Architektura systemu – problemy i koncepcje" [2], w tej warstwie są realizowane następujące czynności:

- Wystawienie zleceń obliczeniowych przez serwer *W* dla serwera *S*.
- Odbieranie wyników obliczeń od serwera *S* przez serwer *W*.

Możliwe zagrożenia to: zniszczenie bądź zmiana treści komunikacji (kodu i danych do obliczeń), podszycie się, przechwycenie i zamiana (atak *Man-in-the-Middle*) komunikacji. Z tego względu ochrona komunikacji *W*–*S* powinna obejmować:

- Zapewnienie spójności i autentyczności przesyłanych zleceń obliczeniowych poprzez użycie podpisu elektronicznego [3].
- Ukrycie komunikacji albo w tunelu SSL, albo poprzez zastosowanie protokołu przesyłania zaszyfrowanych komunikatów wraz z zaszyfrowanym kluczem sesyjnym [3].
- W celu obrony przed ewentualnym podszyciem się użycie stosownego zarządzania kluczami publicznymi – infrastruktury PKI – zarządzającej certyfikatami kluczy publicznych [11].

1.6. Bezpieczeństwo warstwy serwerów *S* i komunikacji pomiędzy warstwą serwerów *S* a serwerami internautów *I*

W tym punkcie zostały opisane aspekty bezpieczeństwa warstwy serwerów *S* (węzłów usługowych dostawców WWW), zajmującej się zarządzaniem podzadaniem obliczeniowym zleconym przez serwer *W*. Zgodnie z dokumentem pt. „Architektura systemu – problemy i koncepcje" [2], w tej warstwie są realizowane następujące czynności:

- Przekazywanie podzadania obliczeniowego (lub wskaźnika na podzadanie obliczeniowe umieszczone na innym serwerze, np. reklamowym) przez serwer *S* internaucie *I*.

- Odbieranie przez serwer *S* wyników obliczeń podzadania od internauty *I* albo bezpośrednio, albo pośrednio – są one przekazywane do serwera reklamodawcy, a serwer *S* otrzymuje wówczas co najwyżej powiadomienie, że obliczenia podzadania ukończono.
- Przekazywanie przez serwer *S* serwerowi *W* wyników bezpośrednio albo przekazywanie informacji o tym, że wyniki są dostępne na innym serwerze, np. reklamodawcy.

Należy wyraźnie podkreślić, że komunikacja pomiędzy komputerem internauty *I* a serwerem *S* odbywa się poza wszelką kontrolą serwerów *W* (i oczywiście całego systemu Comcute). Nie można wymusić obowiązkowego szyfrowania treści podzadania (kodu i danych) bez określonej umowy pomiędzy właścicielem systemu Comcute, a właścicielem serwera *S* i/lub serwera reklamodawcy. W tej sytuacji komunikacji pomiędzy serwerem *S* a komputerem *I* zagrażają typowe ataki internetowe, a szczególnie podsłuch i analiza ruchu. Przed naruszeniem spójności treści komunikacji chroni podpis elektroniczny.

Wymaga poruszenia jeszcze jeden aspekt – z użyciem jakiego klucza (umieszczonego w certyfikacie klucza publicznego) poprawność komunikacji ma być weryfikowana. Aby uchronić się przed koniecznością każdorazowego zapytania internauty o akceptację certyfikatu klucza publicznego weryfikującego poprawność podpisu, tenże certyfikat klucza publicznego musi być wystawiony przez CA (*Certification Authority*) akceptowany automatycznie przez przeglądarki. Jest to warunek konieczny w sytuacji, jeśli nie chcemy, by uwaga właściciela komputera nie była skierowana na ten fakt bądź niepotrzebnie angażowała jego uwagę.

1.7. Niezbędna infrastruktura PKI

Z przedstawionych rozważań wynika, że będą potrzebne co najmniej dwie infrastruktury PKI:

- Certyfikat klucza publicznego pochodzący od zewnętrznego, akceptowanego automatycznie przez przeglądarki, dostawcy usług certyfikacyjnych. Stowarzyszony z nim do pary klucz prywatny powinien być używany do podpisywania podzadania obliczeniowego kierowanego do komputera internauty od serwera *W* poprzez serwer *S* oraz ewentualnie do podpisywania komunikacji od serwerów *W* do serwerów *S* i także w niektórych sytuacjach serwerów reklamodawców. Należy podkreślić, że szczególnej ochrony wymaga klucz podpisujący stowarzyszony z publicznym, umieszczonym w kwalifikowanym certyfikacie.
- Certyfikaty kluczy publicznych używane do podpisów i do zabezpieczania komunikacji w obrębie systemu Comcute, zarządzane przez własną infrastrukturę PKI. Klucz główny (*root*) może, ale nie musi, być podpisany z użyciem klucza prywatnego, którego do pary klucz publiczny jest certyfikowany przez kwalifikowanego dostawcę.

1.8. Podsumowanie

W rozdziale przedstawiono podstawowe mechanizmy niezbędne do zapewnienia bezpieczeństwa elementów składowych i całego systemu Comcute. Ze względu na powszechnie znane zagrożenia pochodzące z Internetu, najpierw powinna być opracowana polityka bezpieczeństwa, definiująca sposoby realizacji celów systemu i procedury postępowania w warunkach zagrożenia. O ile pewne elementy z poniższego zestawienia można pominąć w systemie laboratoryjnym, który jest izolowany od sieci zewnętrznej, to jednak w systemie docelowym niezbędne będzie zaprojektowanie i wprowadzenie następujących składników systemu bezpieczeństwa:

- Polityka bezpieczeństwa – dokument zatwierdzony przez zarząd organizacji – właściciela systemu Comcute;
- Infrastruktura klucza publicznego – hierarchiczna dla systemu i SPKI dla warstwy internetowej;
- Mechanizmy i protokoły zapewniające poufność i wiarygodność komunikacji.

1.9. Literatura

1. Dokumenty NIST serii NCSC-TG, 1986-1991.
2. COMCUTE – dokumentacja projektowa systemu laboratoryjnego, WETI PG 2010-2012.
3. Menezes J., Oorschot P. C. van, Vanstone S. A.: *Handbook of Applied Cryptography. (Kryptografia stosowana)*, WNT 2005.
4. Schneier: *Kryptografia dla praktyków*, wyd.2, WNT 2000.
5. Gollmann: *Computer Security*, 3rd. ed., J.Wiley&Sons 2011.
6. Stokłosa J., Bilski T., Pankowski T.: *Bezpieczeństwo danych w systemach informatycznych*, PWN 2001.
7. Nadalin, C. Kaler: *OASIS Web Services Security: WS-Security Core Specification 1.1*, http://www.oasis-open.org/ committees/ download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf, 2006
8. Özsu M. T., Valduriez P.: *Principles of Distributed Databases 3rd ed.*, Springer 2011
9. Bosworth, S., Kabay, M.E. (ed.), *Computer Security Handbook, 4th ed*., J. Wiley&Sons, 2002
10. Pieprzyk J., Hardjono T., Seberry J. – Teoria bezpieczeństwa systemów komputerowych, 2005, Helion.
11. Szpryngier, P.: *Infrastruktura klucza publicznego SPKI*. W KASKBOOK. Technologie SaaS, red. H. Krawczyk, Gdańsk: Politechnika Gdańska 2004.