WYDAWNICTWO POLITECHNIKI ŚLĄSKIEJ W GLIWICACH

ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2018
Seria: ORGANIZACJA I ZARZĄDZANIE z. 130

BEZPIECZEŃSTWO WARSTWY APLIKACJI PRZESYŁANIA INFORMACJI W SIECI

Bogdan DYBAŁA[1], Mateusz DMITRZAK[2], Ireneusz J. JÓŹWIAK[3]

[1]Politechnika Wrocławska, Wydział Mechaniczny, Wrocław; bogdan.dybala@pwr.edu.pl
[2]Politechnika Wrocławska, Wydział Mechaniczny, Wrocław; dmitrzakmateusz@gmail.com
[3]Politechnika Wrocławska, Wydział Informatyki i Zarządzania, Wrocław; Ireneusz.jozwiak@pwr.edu.pl

Streszczenie: Badania pokazują, że największą część przesyłanych informacji przez Internet stanowią usługi przesyłania poczty internetowej. W artykule przedstawiono analizę bezpieczeństwa w warstwie aplikacji modelu TCP/IP przesyłania informacji w sieci Internet oraz omówiono ich protokoły bezpieczeństwa.

Słowa kluczowe: sieć Internet, przesyłanie informacji, bezpieczeństwo.

SECURITY OF THE APPLICATION LAYER OF INFORMATION TRANSMISSION IN THE NETWORK

Abstract: Studies show that the largest part of the information transmitted over the Internet are Internet mail services. The article presents an analysis of security in the application layer of the TCP/IP model of information transmission over the Internet and discusses their security protocols.

Keywords: Internet, information transmission, security.

1. Wprowadzanie

Gwałtowny rozwój Internetu i jego wykorzystanie w celach prywatnych jak i służbowych, spowodowały zwiększenie zapotrzebowania na różnego rodzaju zabezpieczenia gwarantujące bezpieczeństwo i zwiększoną prywatność w procesie komunikacji poprzez sieć. Badania wskazują, że największą część przesyłanych informacji przez Internet stanowią usługi przesyłania poczty internetowej. Z biegiem czasu kluczowymi elementami stały się wspomniana prywatność i bezpieczeństwo przesyłanych informacji, co spowodowało

wprowadzenie na rynek wielu standardów i protokołów bezpieczeństwa. Tematem niniejszego artykułu jest bezpieczeństwo w warstwie aplikacji modelu TCP/IP, aczkolwiek warto wspomnieć o pozostałych warstwach sieci komputerowych i ich protokołach bezpieczeństwa. Do najważniejszych należą (Kizza, 2013):

- warstwa aplikacji – PGP, S/MIME, S-HTTP, HTTPS, SET, KERBEROS (zostaną szerzej omówione w dalszej części artykułu),
- warstwa transportowa – SSL, TLS,
- warstwa sieciowa – IPSec, VPN,
- warstwa dostępu do sieci – PPP, RADIUS.

Ponieważ model TCP/IP został wyparty przez model OSI, warto zwrócić uwagę, że warstwa aplikacji opisywana w tym artykule pokrywa się z następującymi warstwami modelu OSI: aplikacji, prezentacji i sesji. Warstwa procesowa czy warstwa aplikacji (ang. *process layer*) to najwyższy poziom, w którym pracują użyteczne dla człowieka aplikacje takie jak, np. serwer WWW czy przeglądarka internetowa. Obejmuje ona zestaw gotowych protokołów, które aplikacje wykorzystują do przesyłania różnego typu informacji w sieci (Model, 30.11.2017).

Do najpoważniejszych zagrożeń we współczesnym środowisku internetowym należą ataki polegające na próbach wykorzystania znanych, słabych punktów aplikacji. Hakerzy szczególnie interesują się takimi usługami, jak HTTP (port TCP numer 80) oraz HTTPS (port TCP numer 443), które w wielu sieciach są otwarte. Poprzez otwarty port w sieci rozumiemy brak blokady przez firewall właśnie na tym porcie. Oznacza to, że jakakolwiek próba połączenia do tej sieci na tym porcie nie zostanie zablokowana. Z oczywistych względów nie można zablokować na stałe podstawowych portów, na których oparte jest przeglądanie witryn internetowych, czy przesyłanie e-maili. Urządzenia kontroli dostępu nie potrafią w łatwy sposób wykryć złośliwych programów czy skryptów, których celem są wspomniane usługi i przeprowadzenie ataku (Checkpoint, 30.11.2017).

Dzięki skierowaniu ataków bezpośrednio na aplikacje, hakerzy próbują osiągnąć co najmniej jeden z kilku wymienionych poniżej celów (Checkpoint, 30.11.2017):

- zablokowanie dostępu do usług uprawnionym użytkownikom - ataki DoS (ang. *Denial of Service*). Atak ten polega na masowym, często wykonywanym przez wiele komputerów, wysyłaniu zapytań do serwera nie czekając na odpowiedź. Serwer próbując obsłużyć każde zapytanie, wykorzystuje wszystkie swoje zasoby, przez co po pewnym czasie staje się niedostępny dla użytkowników chcących skorzystać z usług serwera zgodnie z jego przeznaczeniem,
- uzyskanie dostępu z prawami administratora do serwerów lub klientów aplikacji,
- uzyskanie dostępu do baz danych,
- instalacja koni trojańskich, które umożliwiają pominięcie mechanizmów bezpieczeństwa i uzyskanie dostępu do aplikacji,

– instalacja na serwerze programów działających w trybie nasłuchu, które przechwytują identyfikatory i hasła użytkowników.

2. Bezpieczeństwo warstwy aplikacji

Warstwa aplikacji jest częstym celem ataków z kilku powodów. Po pierwsze, jest to warstwa, w której znajduje się ostateczny cel hakerów - dane użytkowników. Po drugie, warstwa aplikacji obsługuje wiele protokołów (HTTP, CIFS, VoIP , SNMP, SMTP, SQL, FTP, DNS, itp.), a zatem stwarza możliwość skorzystania z wielu potencjalnych metod ataku. I po trzecie, wykrywanie i ochrona przed atakami w warstwie aplikacji jest trudniejsza od ochrony w niższych warstwach, ponieważ warstwa aplikacji ma więcej słabych punktów (Checkpoint, 30.11.2017).

W celu skutecznego zapewnienia bezpieczeństwa poziomu aplikacji, zabezpieczenie musi zapewniać następujące cztery mechanizmy obrony (Checkpoint, 30.11.2017):

1. *Zgodność ze standardami*

Oczywiście istnieje wiele standardów określających prawidłowe przesyłanie danych poprzez wybrane protokoły. Jeśli ruch w sieci nie ich nie spełnia, jest to sygnał o możliwych próbach ataków. Ciekawym przykładem jest włączanie kodu binarnego w nagłówek HTTP. Każdy z nas posługując się przeglądarką internetową wysyła i odbiera dwa rodzaje nagłówków HTTP – POST i GET. Służą one odpowiednio do wysłania zapytania do serwera i odbiór odpowiedzi, czyli danych z serwera. Hakerzy posługując się tym mechanizmem potrafią włączyć wykonywalny kod w zapytanie HTTP, które bardzo często nie są sprawdzane przez firewall'e (ang. zapory sieciowe) i stają się groźnym narzędziem w ich rękach.

2. *Sposób wykorzystania protokołu*

W wielu przypadkach, mimo zgodności przesyłanych danych poprzez wybrane protokoły ze standardami, informacje przesyłane są wykorzystywane w innych celach niż pierwotnie zakładają to standardy. Interesującym przykładem jest wykorzystanie przez programy typu Peer 2 Peer (P2P) portu komunikacyjnego 80. Jak wiadomo, port ten jest zarezerwowany dla komunikacji HTTP. W związku z tym nie jest ona blokowana przez zapory sieciowe. Pomijając fakt, że w przypadku rozpowszechniania muzyki, nawet nieświadomie, jest to nielegalne, przesyłanie danych poprzez sieci P2P może zostać wykorzystane przez hakerów to prób włamania czy przechwycenia danych. Drugim przykładem niepoprawnego wykorzystywania protokołu HTTP są zbyt długie nagłówki. Standard nie definiuje maksymalnego rozmiaru, dlatego jest ono często wykorzystywane do przepełniania bufora (ang. *buffer overflow*).

3. Ograniczenie przenoszenia złośliwego kodu

Ograniczenia przenoszenia złośliwego kodu muszą być wrażliwe, np. na ataki Cross Site Scripting. Jak wiadomo, przeglądarki internetowe i serwery aplikacji internetowych wykorzystują bardzo często różnego rodzaju skryptu do poprawy funkcjonalności witryn. Hakerzy poprzez spreparowanie odpowiednich linków ukrytych w wizytówkach lub poczcie email potrafią w ten sposób wykraść prywatne dane z komputera ofiary. Na podobnej zasadzie działają mechanizmy generowania adresów URL zawierających, które zawierają złośliwy kod i są wykonywane w momencie ich wczytania przez przeglądarkę lub klienta poczty.

4. Kontrola operacji

Ten typ kontroli musi być szczegółowo uzgodniony, np. z przedstawicielami firmy, w które wprowadzana jest ta metoda zabezpieczeń. Przykładowo polega ona na określaniu reguł dostępu do wybranych plików na serwerze FTP. Reguła może definiować akcje jakie mogą wykonywać użytkownicy na plikach, które mają w nazwie np. słowo „płace".

3. Protokoły bezpieczeństwa

3.1. Protokół PGP – Pretty Good Privacy

Obecnie bezpieczeństwo i prywatność przesyłanej poczty elektronicznej jest jednym z ważniejszych aspektów polityk bezpieczeństwa wielu korporacji. Jak do tej pory, najlepszą metodą zabezpieczenia przed przechwyceniem niechcianej poczty jest szyfrowanie jej zawartości. W czasach nam bardzo odległych, aby przeczytać pocztę nieadresowaną do nas, należało złapać gońca nim dotarł do celu, otworzyć pieczęć, a po przeczytaniu zawartości poczty zapieczętować tak, aby adresat nie zorientował się o oszustwie. Dzisiaj przesyłanie poczty poprzez Internet znacznie ułatwia przechwycenie i odczytanie poczty bez wiedzy adresata. Co z oczywistych względów sprawia, że szyfrowanie wiadomości staje się kluczowym elementem. Dzięki temu krypto system PGP (ang. *Pretty Good Privacy)* (Pretty, 30.11.2017) jest aktualnie najbardziej popularnym systemem zapewniania bezpieczeństwa i prywatności poczty elektronicznej.

Projekt PGP został zapoczątkowany w 1991 roku przez Philipa Zimmermanna i rozwijany z pomocą społeczności programistów z całego świata. Wydarzenie to stało się pewnym przełomem - po raz pierwszy zwykły obywatel dostał do ręki narzędzie chroniące prywatność, wobec którego pozostawały bezradne nawet najlepiej wyposażone służby specjalne. Program PGP działał na platformach Unix, DOS i wielu innych, będąc dostępnym całkowicie za darmo, wraz z kodem źródłowym. Jednakże, począwszy od wersji 5.0 PGP stało się produktem komercyjnym a następnie zaprzestano rozwijania wersji dla systemów

uniksowych. Dlatego po określeniu standardu w postaci OpenPGP powstała dodatkowo jego niezależna implementacja pod nazwą GNU Privacy Guard (GPG).

Protokół PGP jest oparty na sieci zaufania (ang. *circle of trust*). Jest to zdecentralizowana metoda uwierzytelniania osób, w której nie ma hierarchicznej struktury organizacji uwierzytelniających, a zaufanie do poszczególnych certyfikatów jest sumą podpisów złożonych przez innych uczestników sieci. PGP łączy wygodę systemu publicznego Rivest-Shamir-Adleman (RSA) z szybkością tradycyjnej kryptografii, mechanizmami autoryzacji tekstów (podpisu cyfrowego), kompresją danych przed szyfrowaniem, ergonomicznym wykonaniem i wreszcie zaawansowaną filozofią zarządzania kluczami. Co więcej, PGP przeprowadza wszystkie te operacje szybciej niż większość innych implementacji tego typu. PGP jest systemem dla wszystkich. Jest oparty o mechanizmy kryptografii publicznej. W systemach kryptografii publicznej, każdy z użytkowników posiada dwa klucze - publiczny, udostępniany wszystkim, i prywatny, przechowywany pieczołowicie tylko przez właściciela. Na podstawie znajomości klucza publicznego, nie można odtworzyć klucz prywatnego, i na odwrót. Taki układ wyklucza niebezpieczeństwo przesyłania przez publiczne sieci komputerowe jakichkolwiek danych, umożliwiających dostęp do listu osobom niepowołanym (IPsec.pl, 30.11.2017).

Zasada działania protokołu PGP opiera się na pięciu usługach (Kizza, 2013): uwierzytelnienia (ang. authentication), poufności (ang. *confidentiality*), kompresji (ang. *compression*), zgodności z pocztą elektroniczną (ang. *e-mail compatibility*) oraz segmentacji (ang. *segmentation*). PGP oferuje uwierzytelnianie oparte na podpisie cyfrowym. Do szyfrowania podpisu używana jest kombinacja szyfrów SHA-1 i RSA, natomiast do podpisu alternatywnego można użyć SHA-1 i DSS. Podpis cyfrowy jest umieszczany razem z wysyłana wiadomością lub wysyłany odrębnie. Druga usługą jest poufność. PGP szyfruje wiadomości jeszcze przed ich wysłaniem. Wiadomość jest szyfrowana jedną z następujących metod: CAST-128, IDEA lub 3DES (wszystkie z wykorzystaniem 64 bitowego szyfru). Jak zwykle w przypadku szyfrowania wiadomości pozostaje problem klucza rozszyfrowującego. W przypadku PGP, każda wiadomość zostaje wzbogacona o 128 bitowy klucz szyfrowany RSA lub Diffie-Hallman (klucz publiczny). Odbiorca posługując się swoim kluczem (klucz prywatny) rozszyfrowuje metodą RSA klucz zawarty w wiadomości. Ten rozszyfrowany klucz służy do odczytania zawartości wiadomości.

Kompresja to kolejna usługa polegająca na tym, że każda wiadomość po zaszyfrowaniu zostaje spakowana – jest to spowodowane jedynie oszczędnością pamięci.

Podstawowym problemem była zgodność zaszyfrowanej wiadomości ze standardem poczty elektronicznej, który zakłada że treść musi być w kodzie ASCII. Zaszyfrowany PGP jest natomiast w postaci 8 bitowych ciągów danych binarnych. Rozwiązano to w możliwie najprostszy sposób, podwójna konwersja (najpierw PGP z kodu binarnego na kod ASCII i następnie z kodu ASCII do kodu binarnego) zapewnia zgodność ze standardami (Kizza, 2013).

Aby sprostać wymaganiom standardów poczty elektronicznej PGP dzieli wysyłaną wiadomość na segmenty odpowiedniej wielkości. Klucz i podpis cyfrowy są wysyłane tylko raz, na początku pierwszego segmentu. Po przesłaniu wszystkich segmentów, nagłówki poczty są usuwane i przywracany jest oryginalny stan wiadomości zaszyfrowanej PGP.

3.2. Protokół S/MIME - Secure/Multipurpose Internet Mail Extension

Protokół S/MIME podobnie jak PGP służy do szyfrowania wiadomości email. Rozszerza on podstawowy protokół MIME o podpis cyfrowy i mechanizm szyfrowania. Aby dobrze zrozumieć protokół S/MIME przyjrzyjmy się najpierw jego przodkowi. MIME jest techniczną specyfikacją protokołu komunikacji przesyłania plików multimedialnych (zdjęć, wideo, muzyki). Protokół ten działa na podobnej zasadzie jak wysyłanie i odbieranie żądania HTTP. W skrócie wiadomość email składa się z nagłówka i treści. W nagłówku MIME zawarte się dwie podstawowe informacje: typ i podtyp MIME. Typ określa podstawowy rodzaj informacji jaki przesyłamy (zdjęcie, film, muzyka, tekst, aplikacja itp.). Podtyp określa rozszerzenie przesyłanego pliku (np. *.jpg). Sama treść wiadomości może być bez jakiejkolwiek struktury lub może być oparta o strukturę opisaną w specyfikacji MIME. Najważniejszym aspektem, który dotyczy tego artykułu jest to, że MIME w żaden sposób nie szyfruje przesyłanych danych. Twórcy S/MIME postanowili rozszerzyć MIME właśnie o mechanizmy szyfrowania: szyfrowanie danych (ang. *encryption*) oraz podpis cyfrowy (ang. *digital signature*). S/MIME wspiera trzy algorytmy szyfrowania kluczem publicznym – szyfrowane są klucze zawarte w treści wiadomości. Diffie-Hallman jest preferowanym algorytmem, RSA służy do podpisu i szyfrowania klucza oraz na końcu 3DES (Kizza, 2013). Do tworzenia podpisu cyfrowego S/MIME używa funkcję haszującą 160-bitowego SHA-1 lub MD5. Do rozszyfrowania używane są DSS lub RSA.

3.3. Protokół S-HTTP

Podobnie jak protokół S/MIME rozszerza MIME, tak protokół S-HTTP (Secure-HTTP) rozszerza podstawowy protokół HTTP (ang. *Hypertext Transfer Protocol*) (Kizza, 2013). Przy tworzeniu podstawowego protokołu HTTP nikt nie myślał o dynamicznych stronach internetowych, wymagających dodatkowych zabezpieczeń w postaci szyfrowania. Wraz z rozszerzaniem się możliwości aplikacji internetowych i ich rosnącej roli w biznesie, stwierdzono, że dodatkowa rozbudowa tego mechanizmu o aspekty bezpieczeństwa staje się nieunikniona. W 1994 roku Enterprise Integration Technologies stworzyło S-HTTP rozbudowując jego przodka o możliwość szyfrowania i dodano podpis cyfrowy. Jak już wcześniej wspomniano dane HTTP zawierają dwie części: nagłówek i treść. Nagłówek zawiera informacje i instrukcje w jaki sposób serwer i przeglądarka mają przetwarzać dane zawarte w treści. Dzięki nagłówkom HTTP klient (przeglądarka) i serwer komunikują się i wymieniają informacje o sposobie interpretacji danych. Protokół S-HTTP rozszerzono o możliwość zawierania w nagłówku dodatkowych informacji potrzebnych do korzystania

z podpisu cyfrowego i szyfrowania. Opis możliwości rozszerzonego nagłówka wykracza poza zakres tego dokumentu.

Protokół używa mechanizmu klucza symetrycznego. Przed rozpoczęciem wymiany danych, klient i serwer muszą uzgodnić klucz szyfrowania. Przeglądarka wysyłając żądanie strony dołącza listę schematów szyfrowania jakie obsługuje wraz z kluczem publicznym. Serwer odpowiada również listą szyfrów, które wspiera. Dodatkowo serwer może przesłać klientowi klucz prywatny, wygenerowany kluczem publicznym który otrzymał od klienta. W przypadku, gdy klucz prywatny nie jest dostarczony do klienta, zapytanie szyfrowana jest kluczem publicznym serwera. Jeśli proces wymiany sposoby szyfrowania przebiegnie pomyślnie, dane są przesyłane i rozszyfrowane. W przeciwieństwie do podstawowego protokołu HTTP klient i serwer nie rozłączają się. Połączenie trwa, tak długo jak żąda tego klient. Istotnym aspektem jest obsługa przez protokół S-HTTP wielu żądań na raz.

3.4. Protokół Kerberos

System identyfikacji użytkowników w środowiskach otwartych, opracowany w ramach Projektu Atena w Massachusetts Institute of Technology (MIT) (Słownik, 30.11.2017). Wykorzystywany do autoryzacji żądań dostępu do rozproszonych zasobów sieciowych (np. składników oprogramowania) oraz bezpiecznej wymiany informacji między użytkownikami. W systemie Kerberos każdemu użytkownikowi logującemu się do sieci przydzielany jest unikalny klucz służący jego identyfikacji na czas sesji. Unikalne klucze mogą być wydawane także serwerom (np. gdy zachodzi konieczność nawiązania między nimi bezpiecznych połączeń). Każda sieć posiada wydzielony serwer Kerberos, którego zadaniem jest zarządzanie centralną bazą kluczy (m.in. wydawanie nowych kluczy oraz odpowiadanie na pytania dotyczące ich autentyczności) (Słownik, 30.11.2017).

Nazwa ma swoje źródła w mitologii greckiej Kerberos (Cerber) był trójgłowym psem, jednym z trzech potworów strzegącym bram Hadesu. Przeznaczenie protokołu Kerberos to uwierzytelnianie użytkowników żądających dostępu do zasobów sieciowych. Skonstruowany został w oparciu o koncepcje zaufanej strony niezależnej, której zadanie polega na bezpiecznej weryfikacji użytkowników i usług. W nomenklaturze Kerberos ta zaufana strona nazywana jest centrum dystrybucji kluczy KDC (ang. *Key Distribution Center*), a czasem nazywana jest serwerem uwierzytelniającym. Podstawową metodą wykorzystania Kerberos jest sprawdzanie, czy użytkownicy usług sieciowych są tymi, za których się podają. W tym celu serwer Kerberos przydziela użytkownikom "bilety". Bilety te maja określony czas ważności i są przechowywane w buforze referencyjnym użytkownika. Mogą być one wykorzystywane w zastępstwie standardowych mechanizmów uwierzytelniających, weryfikujących nazwy i hasła użytkowników.

Na początku klient Kerberos zna tylko klucz szyfrujący, znany wyłącznie użytkownikowi i KDC. Identycznie każdy serwer współużytkuje klucz szyfrujący z KDC. Kiedy klient próbuje utworzyć związek z określonym serwerem zasobów, musi uzyskać od KDC bilet

i klucz sesji. W tym celu następuje wymiana zadania uwierzytelnienia i odpowiedzi (Kalata, 2017). Przesyłanie zadań i odpowiedzi jest wymianą, w której klient uwiarygodnia się na serwerze aplikacji, któremu znany jest klucz sesji umieszczony w bilecie Kerberos.

4. Podsumowanie

W niniejszym artykule zostało przedstawionych kilka protokołów bezpieczeństwa wykorzystywanych w warstwie aplikacji do zapewniania prywatności i poufności przesyłanych danych. Opisanie wszystkich protokołów jest zbyt obszernym tematem, aby zmieścić się w ramach tego dokumentu. Istotnym aspektem, który przewodzi tematowi tego artykułu jest znaczny wzrost w ostatnich latach znaczenia i wagi bezpieczeństwa informacji przesyłanych między korporacjami czy użytkownikami prywatnymi. Wymusiło to opracowania szeregu norm, standardów i mechanizmów wspomagających użytkowników w bezpiecznym korzystaniu z dobrodziejstw Internetu. Wydawać się może, że zapoczątkowało to całą machinę, proces, który będzie trwał tak długo jak tylko będzie istniała potrzeba wymiany informacji poprzez ogólnodostępną sieć komputerową. Wraz z każdym nowym ulepszeniem, mającym na celu usprawnienie pracy za pomocą Internetu, mechanizmy bezpieczeństwa muszą ewoluować, aby sprostać wymaganiom ich użytkowników, których głównie interesuje poufność i bezpieczeństwo danych (w większości przypadków). W przypadku gdy zostaniemy pozbawieni możliwości jakiejkolwiek ochrony przed atakami hakerów, śmiało można stwierdzić, że wykorzystanie Internetu w celach biznesowych zmniejszy się prawie do zera.

Bibliografia

1. Kizza, J.M., (2013). *A guide to computer network security.* London: Springer-Verlag.
2. Model TCP/IP - Wikipedia, wolna encyklopedia. (30.11.2017). Available online: http://pl.wikipedia.org/wiki/TCP/IP#Warstwa_aplikacji
3. Pretty Good Privacy, czyli kryptografia publiczna dla mas | IPsec.pl. (30.11.2017). Available online: http://ipsec.pl/meta/pretty-good-privacy-czyli-kryptografia-publiczna-dla-mas.html
4. Prezentacja „Check Point Application Intelligence”. (30.11.2017) http://www.checkpoint.com/products/downloads/applicationintelligence_whitepaper.pdf
5. Słownik. (30.11.2017). http://ws-webstyle.com/pl/netopedia/bezpieczenstwo_hacking/kerberos