WOJCIECH MAZURCZYK, KRZYSZTOF SZCZYPIORSKI
Instytut Telekomunikacji, Politechnika Warszawska
00-665 Warszawa, ul. Nowowiejska 15/19
E-mail: {W.Mazurczyk, K.Szczypiorski}@tele.pw.edu.pl
http://security.tele.pw.edu.pl

BEZPIECZEŃSTWO VOIP OPARTEGO NA SIP

1 Wstęp

Niniejszy artykuł poświęcony jest bezpieczeństwu usługi Voice over IP (VoIP) bazującej na protokole SIP (Session Initiation Protocol). Protokół SIP jest najbardziej obiecującym protokołem sygnalizacyjnym dla realizacji usługi VoIP w sieciach TCP/IP. W artykule przedstawiono zagadnienia związane z bezpieczeństwem wiadomości sygnalizacyjnych wymienianych pomiędzy komunikującymi się stronami, w szczególności przeanalizujemy mechanizmy bezpieczeństwa zastosowane w dwóch zaleceniach organizacji IETF (The Internet Engineering Task Force) dla SIP: RFC 2543 (dot. pierwszej wersji SIP z 1999 r.) oraz RFC 3261 (dot. drugiej wersji SIP z 2002 r.).

2 Podstawy SIP

2.1 Adresowanie

„Obiektami" adresowanymi w protokole SIP są użytkownicy, którzy należą do różnych domen (hostów). Ich identyfikacja odbywa się na podstawie analizy SIP URL postaci np. *sip://j.kowalski@tele.pw.edu.pl*. Stosowany format adresów jest zbliżony do tego stosowanego w adresach e-mailowych np. **user@host** (gdzie *user* oznacza nazwę użytkownika lub numer telefonu, a *host* jest nazwą domeny, bramy lub adresem IP) co ułatwia identyfikację i zapewnia pomoc w dotarciu, bo nawet nie znając dokładnego formatu adresu można go zgadnąć np. na podstawie maila. Ciekawostką dotyczącą adresowania jest możliwość umieszczenia odnośnika (SIP-URL) na stronie WWW umożliwiającego bezpośrednie dotarcie do danego użytkownika (click-to-call).

2.2 Architektura funkcjonalna protokołu SIP

Na architekturę funkcjonalną składają się następujące komponenty:

- **Agent Użytkownika (User Agent)** będący systemem końcowym (zazwyczaj odpowiednio inteligentnym oprogramowaniem), działającym w imieniu użytkownika, który uczestniczy w połączeniu. Składa się on z dwóch części: **klienta** (User Agent Client) i **serwera** (User Agent Server). Klient wysyła **żądania** protokołu SIP, natomiast serwer wysyła **odpowiedzi** w imieniu użytkownika oraz odbiera żądania przesyłane do niego przez innych agentów. Bezpośrednia komunikacja agentów opiera się, więc na popularnym modelu klient/serwer.
- **Serwery Sieciowe (Network Servers)**, których podstawową funkcją jest translacja adresów i pośredniczenie w procesie odnajdywania użytkownika, do którego jest skierowane żądanie.

  Istnieją dwa podstawowe rodzaje serwerów sieciowych w SIP: **proxy** i **redirect**.

- **Proxy,** po otrzymaniu żądania, odpowiedzialny jest za ustalenie adresu następnego serwera, do którego należy je skierować i przy użyciu odpowiednich mechanizmów jak np. DNS zostaje ono tam przesyłane (może być generowane nowe żądanie SIP). Odpowiedź na to żądanie będzie powracać tą samą drogą w odwrotnym kierunku. Rozróżnia się dwa rodzaje serwerów proxy: **stanowy** i **bezstanowy**. Proxy stanowy charakteryzuje się tym, że każde zgłoszenie do niego przychodzące jest zapamiętywane oraz tworzony jest dla niego osobny proces. Musi być stosowany w przypadku, gdy serwer „rozwidla" (forking proxy) wysyłanie wiadomości, gdy współpracuje z protokołem TCP oraz

przy realizacji zaawansowanych usług. Proxy bezstanowy interpretuje każdą wiadomość oddzielnie nie łącząc ich w sekwencje żądanie-odpowiedź.

- **Redirect,** po odebraniu żądania, zajmuje się wysyłaniem do agenta użytkownika odpowiedzi zawierających adres następnego serwera, z którym należy się skontaktować w celu poszukiwania właściwego serwera użytkownika końcowego (nie może inicjować swojego własnego żądania SIP). Serwer redirect jest zawsze bezstanowy.

Dodatkowo funkcjonalność architektury opisywanego protokołu jest wzbogacana serwerem **registrar** będącego serwerem akceptującym żądania REGISTER. Odbiera on zgłoszenia rejestracyjne agentów użytkownika zawierające informacje o obecnej lokalizacji użytkownika. Lokalizowany jest on zwykle wspólnie z serwerami: proxy lub redirect.

Żądania wysyłane przez klienta wywołują w serwerze odpowiednie metody. Po otrzymaniu i zinterpretowaniu żądania agent wysyła odpowiedź, która oznajmia sukces lub niepowodzenie danej operacji albo wskazuje postęp w realizacji wywoływanej metody.

2.3 Metody stosowane w protokole SIP

Zostało wyspecyfikowanych **sześć** podstawowych metod w tym protokole (dla zalecenia RFC 2543):

- **INVITE** – wskazuje, że użytkownik lub serwer został zaproszony do udziału w sesji (konferencji). Treść ciała wiadomości (message body) zawiera opis sesji (wykorzystując protokół SDP) i typ mediów, który ma być wykorzystany podczas rozmowy, jak również adres źródłowy i docelowy, lokalizację użytkownika oraz preferencje dzwoniącego. Może się tu również znaleźć propozycja wybranego przez dzwoniącego kodeka mowy. Metoda musi być zaimplementowana w serwerach proxy, redirect oraz w agencie użytkownika,
- **ACK** – potwierdza, że klient otrzymał końcową odpowiedź na żądanie INVITE. Służy do zapewnienia niezawodnej wymiany INVITE (jest używane tylko z tą metodą). Serwer zawsze retransmituje odpowiedzi finalne aż do uzyskania potwierdzenia ACK od klienta. W ciele wiadomości (message body) może zawierać informacje ostatecznego opisu sesji, z którego powinien korzystać dzwoniący w przeciwnym razie należy użyć parametrów zawartych w żądaniu INVITE. Obsługi tej metody wymaga się od agentów użytkowników oraz serwerów: proxy i redirect.
- **OPTIONS** – przekazuje informację o funkcjonalności (capabilities) – nie zestawia połączenia. Musi być obsługiwana przez serwery SIP: proxy i redirect, agenta oraz registrar.
- **BYE** – oznacza chęć zakończenia połączenia między dwoma użytkownikami sesji. Połączenie może być rozłączone zarówno przez użytkownika, który zainicjował połączenie jak i przez jego odbiorcę.
- **CANCEL** – przerywa żądanie będące w toku, które zawiera te same deskryptory połączenia, ale nie wpływa na już ukończone żądania.
- **REGISTER** – przenosi informację o lokalizacji użytkownika dla serwera SIP. Po wykonaniu rejestracji z adresem użytkownika jego lokalizacja zostanie skojarzona. Może ona się odbyć w dowolnym momencie. Protokół SIP zezwala klientowi na rejestrację z różnych lokalizacji – dla przykładu na początku agent użytkownika może wysłać IP 'mulitcast' do wszystkich aktywnych serwerów SIP w domenie (np. sip.mcast.net, 224.0.1.75). Standardowo użytkownicy posługujący się adresami SIP powinni posiadać swój serwer *registrar* w domenie, do której należą i w nim dokonywać rejestracji.

Funkcje, które nie są obsługiwane przez serwery: proxy i redirect są przez nie traktowane jako metody OPTIONS. Natomiast te, których nie zapewniają serwer agenta użytkownika oraz registrar powodują wysłanie odpowiedzi „nie zaimplementowane".

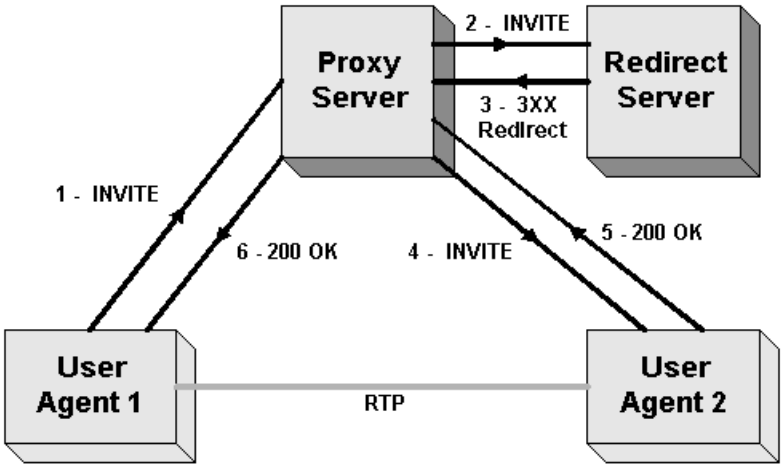
2.4 Rodzaje odpowiedzi

Kody wykorzystane do przesłania statusu (Status-Codes) obecnie wykonywanego żądania zostały podzielone na następujące klasy:

- **1xx -** klasa **Informująca** (Informational)– żądanie zostało odebrane, podjęto kroki w celu jego wykonania, nie napotkano żadnych problemów - wskazuje na postęp w wywoływaniu metody, zawsze po tego typu kodzie następują wiadomości określające status zakończonej operacji.
- **2xx –** klasa **Sukces** (Success) – akcja została zakończona sukcesem.
- **3xx –** klasa **Przekierowanie** (Redirection) – dalsze czynności muszą być podjęte w celu wypełnienia żądania.
- **4xx –** klasa **Błąd Klienta** (Client Error) – żądanie zostało błędnie sformułowane lub nie może być wypełnione przez dany serwer i zostanie odrzucone.
- **5xx –** klasa **Błąd Serwera** (Server Error) – serwer nie zdołał wypełnić pozornie poprawnie sformułowanego żądania.
- **6xx –** klasa **Błąd Globalny** (Global Failure) – żądanie nie może być wypełnione przez żaden z dostępnych serwerów.

2.5 Model inicjacji połączenia

Ogólna zasada nawiązania połączenia, w której występują wszystkie elementy architektury funkcjonalnej protokołu SIP, została przedstawiona na rysunku poniżej:



Rys. nr.4. Połączenie nawiązane za pomocą metody INVITE

Analiza przebiegu połączenia:

1. Od użytkownika (User Agent 1) zostaje wysłane żądanie INVITE (1), które ma zainicjować połączenie z odbiorcą (User Agent 2)

   W wiadomości zostają wyszczególnione: adres SIP-URL nadawcy (pole From) i odbiorcy (pole To) oraz zostaje nadany unikalny identyfikator połączenia Call-ID, który wskazuje bezpośrednio na serwer inicjującego.

2. Wiadomość INVITE zostaje odebrana przez sieciowy serwer proxy, który odpowiedzialny jest za ustalenie adresu następnego serwera, do którego należy je skierować.
3. Następnie wiadomość INVITE (2) zostaje przesłana do sieciowego serwera redirect, który zajmuje się wysyłaniem do proxy odpowiedzi zawierającej adres następnego serwera, z którym należy się skontaktować w poszukiwaniu właściwego serwera użytkownika końcowego.
4. W wyniku działania tegoż odnaleziona zostaje domena odbiorcy (3).
5. Następnie serwer proxy ponawia wysłanie żądania INVITE, tym razem używając otrzymanego ostatnio adresu (4).
6. Serwer, do którego została skierowana wiadomość odpowiada akceptując połączenie przesłane serwerowi docelowemu odpowiedzi 200 (OK).
7. Źródłowy serwer agenta użytkownika przesyła wiadomość potwierdzenia ACK do odbiorcy i następuje faza rozmowy (pakiety RTP).

3 Bezpieczeństwo połączeń VoIP

Zagwarantowanie bezpiecznych połączeń dla usługi VoIP jest sprawą złożoną. Nie ogranicza się ono tylko do zapewnienia zabezpieczonego transportu strumieni danych zawierających głos, ważniejszą sprawą jest fakt, w jakich warunkach przesyłane są wiadomości protokołu sygnalizacyjnego, na którym bazuje VoIP.

Problemy bezpieczeństwa dla usługi telefonii IP w świetle powyższych stwierdzeń można podzielić na:

a. **Bezpieczeństwo wiadomości sygnalizacyjnych wymienianych pomiędzy stronami komunikującymi się**,
b. Bezpieczeństwo pakietów przenoszących głos (pakiety RTP),
c. Problemy związane z „przechodzeniem" pakietów przez ściany przeciwogniowe (Firewalls) oraz przez mechanizmy translacji adresów wewnętrznych (np. intranetowych) na zewnętrzne (np. internetowe) NATs (Network Address Translators).

Niniejszy artykuł skupia się wyłącznie na tematyce zawartej w punkcie a; bezpieczeństwo przepływu pakietów „z głosem" (b) jest zapewniane niezależne od gwarantowania bezpiecznej sygnalizacji. Trzecia grupa problemów (c) nie jest związana bezpośrednio z bezpieczeństwem samego protokołu SIP – dotyczy raczej sieci będącej implementacją usługi VoIP opartej na protokole SIP. W takiej sieci z powodu wymienionych „przeszkód" może dojść do uniemożliwienia:

- Nawiązania połączenia z powodu blokowania pakietów zawierających sygnalizacje SIP,
- Porozumiewania się przez strony żądające komunikacji ze względu na blokowanie pakietów RTP.

4 Aspekty bezpieczeństwa w SIP

W czasie fazy sygnalizacyjnej określone parametry sesji są wymieniane pomiędzy użytkownikami końcowymi w celu poprawnej realizacji żądanego połączenia. Mogą one zawierać informacje, które użytkownik wolałby pozostawić niedostępne dla osób trzecich (np. jego lokalizacja, czy nazwisko). Ważne jest również to, aby użytkownicy, którzy nie są uwierzytelnieni nie mieli możliwości zmiany, wstawiania oraz usuwania wiadomości wysyłanych w czasie fazy sygnalizacyjnej. Aby zapewnić te oraz inne aspekty bezpieczeństwa, które są już przy tym poziomie rozwoju teleinformatyki konieczne dla każdej

nowoczesnej sieci stosuje się mechanizmy takie jak np. szyfrowanie, wymianę uwierzytelniającą, funkcje skrótu oraz podpisy cyfrowe.

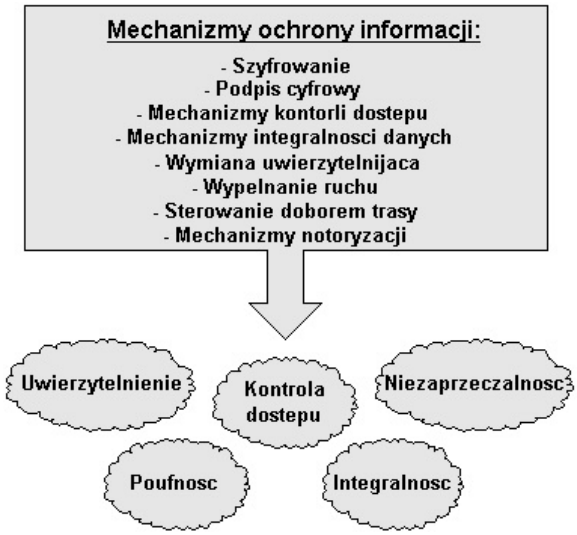
4.1 Problem doboru kryterium oceny bezpieczeństwa dla SIP

Aby usystematyzować analizę mechanizmów bezpieczeństwa protokołu SIP trzeba na początku ustalić kryterium, według którego nastąpi jego analiza. Nie jest to wbrew pozorom prosta sprawa, gdyż należy przy tym wyborze kierować się specyfiką protokołu, charakterystyką potencjalnych zagrożeń itp. tak, aby być w stanie rozważyć wszystkie aspekty zapewniania bezpieczeństwa jak najpełniej.

Przedstawiony poniżej wywód prezentuje tok rozumowania autorów, który został przyjęty przy ustalaniu niezbędnego do dalszej analizy kryterium bezpieczeństwa protokołu SIP. Wymagało to przypomnienia kilku podstawowych definicji z zakresu ochrony informacji. Zaleca się, aby czytelnik posiadający tę wiedzę ograniczył się tylko do pobieżnego zapoznania z niniejszym punktem.

Spośród dostępnych, potencjalnych kryteriów zdecydowaliśmy się wybrać rozwiązanie będące modyfikacją podziału zawartego w normie ISO 7498-2, według którego bezpieczeństwo w systemach otwartych należy rozpatrywać w kontekście możliwości zapewnienia pięciu podstawowych **usług** ochrony informacji (kontroli dostępu (access control), uwierzytelnienia (authentication), integralności danych (data integrity), poufności danych (confidentiality) oraz niezaprzeczalności (non-repudation)).

Podstawowe usługi przedstawione powyżej są budowane na bazie **mechanizmów**. Wspomniana norma ISO definiuje je również. Przedstawia je poniższy rysunek:



Rys. 5. Mechanizmy budujące podstawowe usługi ochrony informacji

Postaramy się teraz wykazać, które z wymienionych usług wpływają w sposób znaczący na bezpieczeństwo protokołu SP. Usługa integralności zawiera się w uwierzytelnieniu, a ono współtworzy niezaprzeczalność z tą różnicą, że dla niezaprzeczalności dodatkowo nie ma możliwości wyparcia się przez nadawcę (odbiorcę) faktu wysłania (odebrania) określonej wiadomości. Zapewnienie tej usługi nie jest sprawą najistotniejszą w przypadku przesyłania wiadomości sygnalizacyjnych protokołu SIP, które kontrolują lub modyfikują sesje użytkowników. Można jednak założyć, że szczątkowo niezaprzeczalność jest zrealizowana obecnie w SIP w postaci usługi uwierzytelnienia.

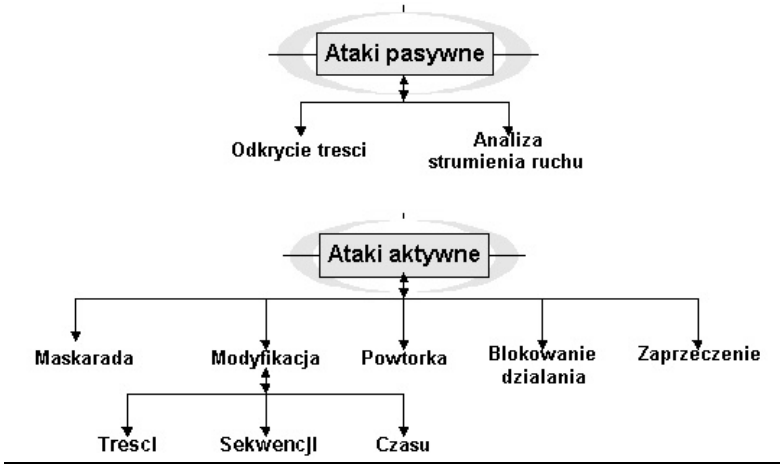
Natomiast usługa kontroli dostępu do zasobów nie odnosi się bezpośrednio do wymiany wiadomości sygnalizacyjnych SIP, gdyż dostęp np. do usługi telefonicznej powinien mieć każdy, a to czy ktoś nieupoważniony nie narusza zasobów w sieci jest poza gwarancjami bezpieczeństwa oferowanymi przez omawiany tu protokół sygnalizacyjny. Poza tym uwierzytelnienie w sieci opartej na protokole SIP odbywa

się w celu ustanowienia sesji lub uzyskania dostępu do określonych usług sieciowych, a zatem pośrednio realizowana jest usługa kontroli dostępu.

Zastosowane kryterium, aby być miarodajnym, powinno dodatkowo uwzględniać jak najwięcej klas zagrożeń oraz potencjalnych problemów bezpieczeństwa, ponieważ wymiana informacji pomiędzy uczestnikami sesji może podlegać szeregowi ataków. Zatem kryterium powinno zostać dopasowane tak, aby jak najpełniej oddać również tą składową problemu.

Zagrożenia powstałe na wskutek celowej akcji ze strony intruza mogą mieć charakter **pasywny** (istnieje jedynie możliwość podsłuchu lub stwierdzenia faktu przepływu wiadomości) lub **aktywny** (może dojść do ingerencji w przesyłanie wiadomości np. poprzez zmianę jej zawartości).

Podział na poszczególne **klasy ataków** na sieć oraz na komunikowanie się w niej z podziałem na pasywne i aktywne przedstawia poniższy rysunek. Klasyfikacja ta jest zmodyfikowaną wersją podziału wg Stallings'a.



Rys. 6. Podział ataków na sieć oraz komunikację w niej

Konkludując można przyjąć odpowiadające specyfice protokołu SIP kryterium oceny mechanizmów bezpieczeństwa tego protokołu jako umiejętność zapewnienia dwóch głównych usług ochrony informacji oraz komunikacji w sieci tzn.:

- **Poufności** – dającej ochronę przed atakami pasywnymi oraz zabezpieczającej wiadomości sygnalizacyjne wymieniane pomiędzy agentami użytkowników komunikującymi się ze sobą przed nieuprawnionym ich uzyskaniem przez strony do tego nieupoważnione;
- **Uwierzytelnienia** – gwarantującego ochronę przed większością ataków aktywnych oraz kontrolę tożsamości stron i/lub wiadomości sygnalizacyjnych wymienianych pomiędzy nimi. Podobnie jak w normie ISO 7498-2 integralność wiadomości zawiera się w uwierzytelnieniu, a ono następnie stanowi główną część niezaprzeczalności.

Dlatego przy omawianiu mechanizmów bezpieczeństwa protokołu SIP główny nacisk położono na sposób zapewnienia właśnie tych dwóch usług.

4.2 Techniki ataków na protokół SIP

W świetle przedstawionego podziału na klasy ataków na bezpieczeństwo sieci i komunikacji w niej oraz usług ochrony informacji, jakie powinny być oferowane postaramy się zidentyfikować i sklasyfikować potencjalne techniki ataków na SIP - techniki, czyli określone działania oraz narzędzia użyte do ataku, które mogą być wykorzystane w różny sposób, a zatem jedna technika może być wykorzystana do stworzenia wielu klas zagrożeń (typów ataków). Poniżej w tabeli prezentujemy najczęściej używane techniki.

Technika ataku:	Opis:	Klasy zagrożeń dla sieci i wiadomości w niej przesyłanych:
Podszycie się (Spoofing)	Technika ta polega na podszywaniu się np. pod cudze adresy IP, dzięki czemu możliwe jest np. wysyłanie sfałszowanych informacji	Odkrycie treści wiadomości, Modyfikacja, Maskarada, Blokowanie działania, Powtórka, Zaprzeczenie
Podsłuchiwanie (Sniffing)	To technika monitorowania i analizowania pakietów (pasywne nasłuchiwanie) wszystkich wiadomości przesyłanych przez sieć.	Odkrycie treści wiadomości, Powtórka
Blokowanie działania (Denial of Service - DoS)*	Polega na blokowaniu funkcjonowania elementów sieci np. za pomocą wysyłania dużej liczby wiadomości. Tak obciążony element nie wytrzymuje naporu wiadomości i zostaje zablokowany.	Blokowanie działania

* istnienie takiej techniki ataku nie wyklucza istnienia klasy ataków o takiej samej nazwie

Tab. 1. Techniki wykorzystywane do ataków na protokół SIP

Przykłady typowych ataków z pomocą scharakteryzowanych technik to:

Podszycie (*Spoofing*), które może powodować wiele klas zagrożeń i występuje w wielu postaciach. Przeprowadzenie ataku z wykorzystaniem tej techniki jest podobne do ataku na protokół SMTP. Dla protokołu SIP mogą to być sytuacje, gdy następuje *spoofing*:

- Pola *From* w metodzie REGISTER (przekierowanie połączenia) – zwane również *Registration Hijacking,*
- Pola *From* w metodzie INVITE.
- *Man in the Middle* – atakujący jest w stanie podsłuchać serwer proxy i również zmieniać kluczowe informacje w wiadomościach,
- *Impersonating a Server* – agent użytkownika kontaktuje się z serwerem proxy, w celu dostarczenia żądania, a intruz podszywa się pod serwer. Sytuacja szczególnie się komplikuje w przypadku obsługi użytkowników mobilnych,
- *Tearing down session* – wstawienie przez atakującego odpowiedzi BYE w czasie, gdy zachodzi komunikacja między użytkownikami.

Podsłuch (*Sniffing*), które też może przyjmować dla SIP następujące postacie:

- Podsłuchiwanie wiadomości sygnalizacyjnych (*Eavesdropping*),
- Penetracja ciała wiadomości sygnalizacyjnych (*Tampering with message bodies*).

Istnienie wymienionych powyżej rodzajów ataków powoduje konieczność istnienia określonych rodzajów mechanizmów bezpieczeństwa koniecznych, aby im przeciwdziałać. Przejdźmy teraz do ich analizy.

5 Architektura bezpieczeństwa SIP w zaleceniu RFC 2543

Session Initiation Protocol nie definiuje ściśle określonego typu uwierzytelnienia ani techniki stosowanej do realizacji mechanizmu szyfrowania – jest to potencjalnie łatwa rozszerzalność. Jakkolwiek proponuje, jak można wykorzystać mechanizmy uwierzytelniające zastosowane w protokole HTTP, szyfrowanie oraz generację podpisów cyfrowych z użyciem PGP (Pretty Good Privacy).

5.1 Uwierzytelnienie

Uwierzytelnienie wiadomości jest to procedura sprawdzania, czy pochodzą one z zaznaczonego w nich źródła i czy nie zostały zmienione w czasie ich transportu (czyli integralność). Dodatkowo może ona obejmować weryfikacje kolejności oraz autentyczności czasu.

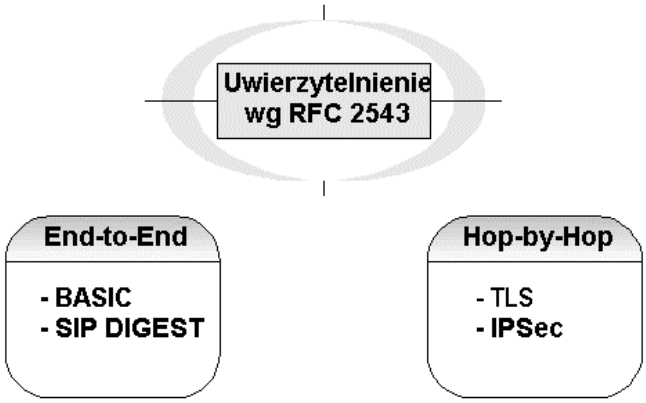
Usługa uwierzytelnienia jest możliwa z punktu widzenia kierunku jej wykonania jest dostępna w dwóch trybach: jednokierunkowym oraz dwukierunkowym (*mutual* - wzajemna).

Uwierzytelnienie w sieci opartej na protokole SIP jest procesem, w którym agent użytkownika dostarcza swoje dane uwierzytelniające serwerowi SIP (głównie proxy) lub innemu agentowi użytkownika w celu:

- Ustanowienia sesji
- Uzyskania dostępu do określonych usług sieciowych.

Mechanizmy uwierzytelniające opisane w zaleceniu RFC 2543 można podzielić ze względu na obszar realizacji omawianej usługi w odniesieniu do drogi komunikacyjnej na mechanizmy typu **End-to-End** (obsługa bezpośrednia źródło->cel) oraz **Hop-by-Hop** (obsługa tranzytowa – tylko pojedyncze połączenie warstwy transportowej). Podobny podział będzie również wprowadzony w przypadku zalecenia RFC 3261).

Wyszczególnienie konkretnych mechanizmów z uwzględnieniem tego podziału zostało umieszczone na rysunku poniżej.



Rys. 7. Realizacja usługi uwierzytelnienia w SIP wg RFC2543

W dalszej części tego rozdziału zostaną przedstawione wymienione mechanizmy z uwzględnieniem istniejącego na rysunku podziału.

5.2 Uwierzytelnienie typu End-to-End

Aby zapewnić tę usługę protokół SIP oferuje dwa różne mechanizmy: *Basic* i *Digest*. Nie były one specjalnie zaprojektowane dla tego protokołu sygnalizacyjnego – zostały one zaczerpnięte w prawie niezmienionej formie z protokołu HTTP.

Oba mechanizmy bazują na wykorzystaniu czterech pól nagłówka przesyłanej wiadomości, z których część jest charakterystyczna dla serwerów proxy, a reszta dla agenta użytkownika. Są to: ***WWW-Authenticate***, ***Authorization***, ***Proxy-Authenticate*** oraz ***Proxy-Authorization*** (oraz dla *Digest* dodatkowo może być użyte pole: ***Authentication-Info***).

5.3 Uwierzytelnienie Basic

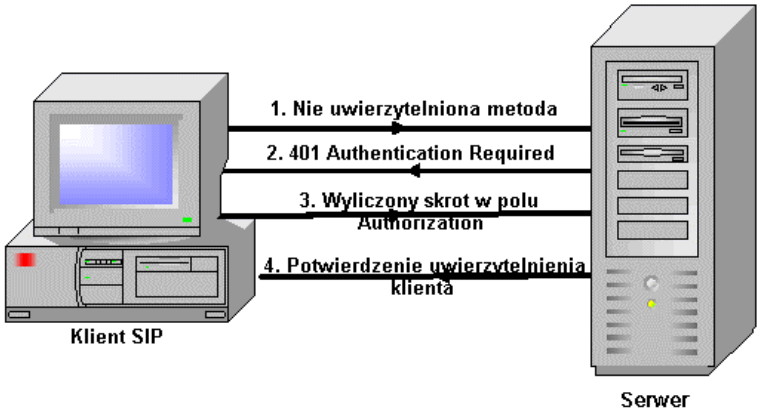
Bazuje na schemacie współdzielonego sekretu (shared secret) - użyty klucz jest znany przez obie strony (serwera i klienta). Mechanizm ten jest jednak bardzo prymitywny i nie gwarantuje wystarczającego poziomu bezpieczeństwa, ponieważ dane uwierzytelniające (hasło) użytkownika są przesyłane jako tekst jawny.

5.4 Uwierzytelnienie Digest

Ten mechanizm uwierzytelniający jest lepiej zabezpieczony od poprzednika (daje on ochronę przed większością defektów uwierzytelnienia typu *Basic)*, lecz nadal w konfrontacji z nowoczesnymi standardami kryptograficznymi jest stosunkowo słaby.

Bazuje na wykorzystaniu współdzielonego sekretu razem z uwierzytelnieniem metodą wyzwanie/odpowiedź (challenge / response) oraz zastosowaniem funkcji skrótu (hash function).

Przebieg realizacji omawianej usługi dla mechanizmu *Digest* jest następujący: gdy klient otrzymuje wyzwanie z odpowiedzią *401 Authentication Required*, która zawiera w nagłówku pole *WWW-Authenticate* wykorzystuje swoje hasło i część informacji przeniesionych z tego właśnie pola do wyliczenia skrótu. Jest on następnie dołączany do nagłówka w pole *Authorization* w ponownie wysyłanym do serwera żądaniu. Obrazuje to następujący rysunek:



Rys. 8. Przebieg uwierzytelnienia *Digest*

Do wypełnienia wspomnianego pola nagłówka *Authorization, SIP Digest* wykorzystuje się po stronie klienta charakterystyczne dla niego informacje (zwykle kombinacje nazwy użytkownika i hasła) w połączeniu z parametrem *nonce* dostarczanym przez serwer. Dopiero taki klucz jest stosowany do funkcji haszującej MD5 (Message Digest 5). Algorytm wykorzystuje właśnie ten połączony ciąg znaków i „skraca" go do postaci 128-bitowej. Następnie wysyła otrzymany skrót do serwera, który zna dane użytkownika (shared secret) łączy je z parametrem *nonce* i również stosuje tą samą funkcję skrótu. Kolejnym etapem jest porównanie obu skrótów i jeśli okazują się być takie same tym samym klient zostaje uwierzytelniony.

Ważnym aspektem w odniesieniu do zagwarantowania bezpieczeństwa uwierzytelnienia mechanizmem *Digest* jest odpowiedni dobór parametru *nonce,* który charakteryzuje bieżące wyzwanie. Powinien być on unikalny dla każdej wymiany uwierzytelniającej. Jest on tworzony zwykle, w zależności od implementacji, z wykorzystaniem charakterystycznego klucza serwera, adresu IP klienta, znaku czasowy (timestamp) itp. Poprzez umiejętny dobór tego parametru można uniemożliwić przeprowadzenie udanego ataku typu powtórka.

Podobna sytuacja następuje w przypadku potrzeby uwierzytelnienia serwera na rzecz klienta, z tą różnicą, że na nie uwierzytelnione żądanie serwera klient wysyła odpowiedź: *407 Proxy Authorization Required*.

Po uwierzytelnieniu klienta kolejne odpowiedzi serwera mogą zawierać pola nagłówka *Authentication-Info*, aby zapewniać wzajemne uwierzytelnienie (Mutual Authentication). Stosuje się je po to, aby uniknąć sytuacji, w której trzeba by było przy każdym nowym żądaniu powtarzać pełny mechanizm uwierzytelnienia. Działoby się tak, ponieważ serwer przy każdej nowej wiadomości sygnalizacyjnej generuje nową wartość parametru *nonce*, a tak przy wykorzystaniu wspomnianego wyżej pola można poinformować go o zmienionej wartości potrzebnego do obliczenia skrótu parametru.

SIP Digest może oferować prymitywną formę wsparcia usługi integralności danych poprzez dodanie do informacji poddanych funkcji skrótu ciała wiadomości lub określonych nagłówków. Jednak w takiej formie protokół SIP nie wspiera uwierzytelnienia odpowiedzi oraz nie przeprowadza się uwierzytelniania w relacji *proxy-proxy*. Obie przedstawione wersje mechanizmu *SIP Digest* zapewniają integralność wiadomości wyłącznie w szczątkowej formie, co kłóci się z definicją uwierzytelnienia założoną przez nas przy definiowaniu wzajemnych relacji pomiędzy poszczególnymi usługami (oraz samych usług). Biorąc pod uwagę to i inne „defekty" obu opisanych dotychczas mechanizmów należy wysnuć wniosek, że jakość uwierzytelnienia przez nie gwarantowana jest bardzo niska.

5.5 Uwierzytelnienie typu Hop-by-Hop

Protokoły gwarantujące uwierzytelnienie tego typu to: protokół warstwy transportowej TLS (Transport Layer Security) oraz warstwy sieciowej IPSec (Internet Protocol Security). Nie zostały one obowiązkowo narzucone w RFC 2543 – zakłada się tam tylko, że jeśli trzeba będzie zapewniać uwierzytelnienie Hop-by-Hop to można użyć jednego z tych dwóch protokołów. Należy również przypomnieć, że oba protokoły realizują pełną usługę integralności wiadomości, czego niestety nie gwarantują mechanizmy uwierzytelnienia typu End-to-End w tym zaleceniu, co dowiedziono powyżej.

5.6 Poufność

Zapewnienie poufności wiadomości gwarantuje, że tylko upoważnione do tego osoby są w stanie odczytać jej zawartość. Realizację tej usługi osiąga się w SIP poprzez zastosowanie mechanizmu szyfrowania. Nie ma narzuconych systemów kryptograficznych dla realizacji tej usługi, a zatem potencjalnie istnieje łatwa rozszerzalność praktycznie zaś brak narzuconego rozwiązania zwykle owocuje brakami implementacyjnymi.

Wiadomości SIP (żądania i odpowiedzi) mogą zawierać ważne dane dotyczące połączenia, jak również w samym ciele wiadomości znajdować mogą się klucze użyte do szyfrowania danej sesji. Zapewnienie prywatności połączenia poprzez szyfrowanie oznacza zabezpieczenie przed dostępem osób trzecich do informacji o agencie użytkownika tj. niemożliwość pozyskania adresu IP czy numeru portu. Dodatkowo maskowane są parametry zestawianego połączenia. SIP obsługuje trzy uzupełniające się formy szyfrowania:

- ***End-to-End*** – gdzie szyfrowaniu podlega całe ciało wiadomości oraz istotne z punktu widzenia bezpieczeństwa pola nagłówka. Żądania i odpowiedzi nie mogą być w ten sposób szyfrowane w całości, ponieważ wymagana jest dostępność niektórych pól np. *To* i *Via* dla serwerów proxy po to, aby wiadomość mogła zostać poprowadzona właściwą drogą.

  W RFC 2543 zaleca się wykorzystywać przy szyfrowaniu dwa pola nagłówka: *Encryption* oraz *Response-Key* – jeden do wskazania, że szyfrowanie zostało użyte w wiadomości, a drugie zawiera klucz odpowiedzi. Realizacja tego szyfrowania oparta jest na kluczach współdzielonych przez obu agentów uczestniczących w wypełnieniu żądania. Zwykle wiadomość jest zaszyfrowana z użyciem klucza publicznego odbiorcy, żeby tylko on był stanie ją przeczytać. Wszystkie implementacje bazujące na tym zaleceniu powinny obsługiwać szyfrowanie na bazie PGP (Pretty Good Privacy).

- ***Hop-by-Hop*** – zapobiegająca możliwości wytropienia, kto dzwoni, do kogo. Szyfrowana jest w tym przypadku cała wiadomość SIP. Możliwe jest również zaszyfrowanie wiadomości, na której uprzednio została przeprowadzona metoda *End-to-End*. Jednak w takiej sytuacji serwery proxy nadal muszą być w stanie zidentyfikować rozmawiających, a zatem identyfikacja stron jest łatwa do ustalenia np. poprzez przeprowadzenie analizy ruchu w sieci (co stanowi to o ograniczoności tego rodzaju zabezpieczenia). Realizacja takiego typu szyfrowania jest możliwa w warstwie sieciowej lub transportowej modelu TCP/IP. Do tego celu wykorzystuje się jeden z dwóch mechanizmów: IPSec lub TLS (podobnie jak w przypadku uwierzytelnienia).

  Inną odmianą tej wersji szyfrowania jest szyfrowanie *Hop-by-Hop pola Via* – w celu ukrycia trasy, którą podjęła dana wiadomość. Pole *Via* jak już wcześniej wspomniano jest używane po to, aby odpowiedź mogła zostać wysłana tą samą drogą, którą przyszło żądanie oraz w celu likwidacji nieskończonych pętli żądań. Szyfrowanie tego pola jest jednak możliwe bez utraty jego funkcjonalności poprzez odpowiednią współpracę serwerów proxy i wykorzystanie odpowiedniego pola nagłówka (*Hide: hop*).

Wobec powyższych faktów należy jasno stwierdzić, że szyfrowania typu *Hop-by-Hop* nie są wspierane bezpośrednio przez protokół SIP – wykorzystują do tego wspomniane protokoły warstw niższych.

W przypadku wykorzystania szyfrowania wszystkie pola nagłówka, które pozostaną niezaszyfrowane muszą poprzedzać te zaszyfrowane. Obowiązkowe w związku z tym jest wystąpienie dodatkowo dwóch pól:

- Pola *Encryption,* które wskazuje na wykorzystanie w danej wiadomości mechanizmów szyfrowania,
- Pola *Content-Length*, wskazującego długość zaszyfrowanego ciała wiadomości.

Każde pole nagłówka, które zostało zaszyfrowane w przychodzącym żądaniu powinno mieć taki sam status w odpowiedzi.

Dodatkowo:

Istnieje możliwość **szyfrowania przez serwery proxy** żądań SIP (jedyny przypadek) - gdy system końcowy sam nie może wykonać tej operacji.

6 Architektura bezpieczeństwa SIP w zaleceniu RFC 3261

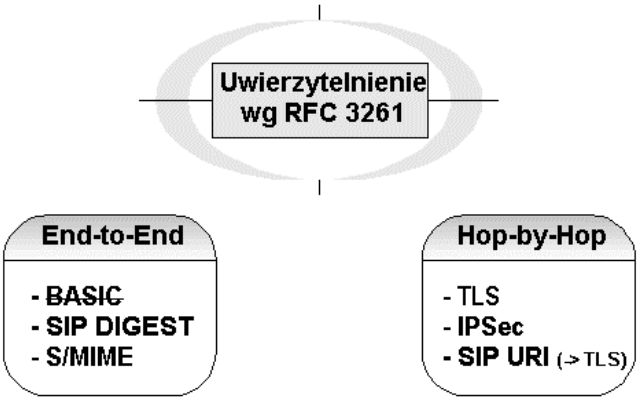
Nowe zalecenie konstytuujące drugą wersje protokołu SIP zostało opublikowane w czerwcu 2002 roku. Uwzględnia ono aspekty zapewnienia bezpieczeństwa w dużo szerszym zakresie niż pierwsza wersja zawarta w RFC 2543. Wymienia się tam potencjalne ataki na protokół SIP, a następnie charakteryzuje mechanizmy, które pozwalają im zapobiegać. Zweryfikowane zostały również stare mechanizmy i dodane nowe. Konwencja tworzenia protokołu SIP polegała właśnie na tym, aby wykorzystywać możliwie jak najwięcej istniejących już i sprawdzonych komponentów (mechanizmów) jest tak zarówno w nowym jak i starym zaleceniu. SIP wykorzystuje mechanizmy zaprojektowane oczywiście przez IETF.

Architekturę bezpieczeństwa tego zalecenia będziemy rozpatrywać zgodnie z przyjętym kryterium dla umiejętności zagwarantowania usług uwierzytelnienia oraz poufności (podobnie jak w przypadku wersji pierwszej tego protokołu) dla dwóch grup mechanizmów: End-to-End oraz Hop-by-Hop.

Na początek należy wspomnieć, iż w zaleceniu RFC 3261 mechanizmy bezpieczeństwa typu Hop-by-Hop zostały zapewnione (dla obu usług) poprzez warstwy niższe – czyli transportową (TLS) oraz sieciową (IPSec) modelu TCP/IP.

6.1 Uwierzytelnienie typu End-to-End

W stosunku do poprzedniej wersji protokołu SIP zrezygnowano ze stosowania uwierzytelnienia typu *Basic*, natomiast konieczne staje się implementowanie *SIP Digest*, które zostało nieznacznie zmodyfikowane. Dodatkowo dodano możliwość skorzystania z nowego rozwiązania zwanego *S/MIME (Secure / Multipurpose Internet Mail Extension)*. Zatem realizacja usługi uwierzytelnienia w tym zaleceniu przedstawia się następująco:



Rys. 9. Realizacja usługi uwierzytelnienia wg RFC 3261

Przejdziemy teraz do opisu i analizy nowych mechanizmów tutaj zastosowanych.

6.2 Uwierzytelnienie wzajemne z wykorzystaniem S/MIME

S/MIME jest to protokół zapewniający bezpieczeństwo protokołowi MIME, który jest z kolei głównym standardem wykorzystywanym do (de)kodowania plików o różnych formatach oraz znaków narodowych.

Wiadomości SIP zawierają ciała MIME, a ten standard definiuje protokół S/MIME do zapewnienia usługi uwierzytelnienia i poufności.

S/MIME oferuje możliwość podpisywania oraz szyfrowania jednostek MIME (czyli zapewnienia obu usług ochrony informacji i komunikacji w sieci: uwierzytelnienia oraz poufności). Dla zagwarantowania pierwszej z nich S/MIME wykorzystuje tunelowanie. Osiąga się to poprzez wykonanie pełnej lub częściowej kopii nagłówków wiadomości SIP i umieszczenie jej wraz z oryginalnym ciałem w wiadomości „wewnętrznej". To właśnie ta „wewnętrzna" wiadomość jest enkapsulowana w jednostce MIME, która reprezentuje ciało nowej wiadomości. Może ona zostać również podpisana cyfrowo z wykorzystaniem S/MIME. W takim przypadku oryginalne nagłówki są umieszczane w wiadomości „zewnętrznej" i mogą być modyfikowane podczas transmisji. Po osiągnięciu celu odbiorca wiadomości weryfikuje dostarczony podpis cyfrowy. Jeśli podpis jest prawidłowy to dodatkowo należy dokonać porównania nagłówków wiadomości „wewnętrznej" i „zewnętrznej" oczywiście tych, które nie były wykorzystywane przez urządzenia znajdujące się na drodze komunikacyjnej, ponieważ one mogły być modyfikowane. Jest to po prostu zapewnienie podstawowej integralności wiadomości. Jeśli taka procedura przebiegnie prawidłowo oznacza to, iż wiadomość nie została zmodyfikowana w trakcie transmisji.

6.3 Uwierzytelnienie typu Hop-by-Hop

W przypadku tego rodzaju uwierzytelnień pozostały sprawdzone z pierwszej wersji SIP rozwiązania wykorzystujące protokoły warstw niższych modelu TCP/IP: transportowej - TLS oraz sieciowej - IPsec. Nowością dodaną w RFC 3261 jest mechanizm o nazwie SIPS URI.

6.4 Uwierzytelnienie SIPS URI

Jeśli zwykły URI (Uniform Resource Identifier) postaci **sip://JKowalski@pw.edu.pl** zamienimy na **sips://JKowalski@pw.edu.pl** będzie to oznaczać fakt, iż cel wyszczególniony w adresie ma zostać osiągnięty w sposób bezpieczny, co oznacza, iż wszystkie urządzenia oraz cała droga powinna być zabezpieczone z wykorzystaniem protokołu TLS. Jeśli taki warunek nie może zostać spełniony to nie dochodzi do nawiązania połączenia. Innymi słowy, jeśli jako mechanizm gwarantujący uwierzytelnienie został wybrany SIPS URI oznacza to, że musi zostać zaimplementowany w takiej sieci protokół TLS, gdyż jest on niezbędny do prawidłowego jego funkcjonowania.

W przypadku nie stosowania mechanizmu SIPS URI, jeśli usługa poufności jest gwarantowana w „zwykły" sposób z wykorzystaniem TLS oznacza to jedynie, że odcinek pomiędzy agentem użytkownika wysyłającym żądanie, a pierwszym serwerem jest zabezpieczony za pomocą tego protokołu.

6.5 Poufność

W przypadku gwarantowania tej usługi ochrony informacji w porównaniu z pierwszą wersją protokołu SIP nastąpiła całkowita rezygnacja z szyfrowania typu End-to-End za pomocą PGP.

Została ona zastąpiona opcjonalnym wykorzystaniem do tego celu protokołu S/MIME, który już na pierwszy rzut oka wydaje się być lepszym, bardziej intuicyjnym rozwiązaniem niż PGP. Niestety nie wiadomo, czemu organizacja IETF nie zastosowała protokołu S/MIME już w pierwszej wersji SIP.

Szyfrowanie typu Hop-by-Hop – patrz początek rozdziału (dla przypomnienia: TLS oraz IPSec).

6.6 Poufność z wykorzystaniem protokołu S/MIME

Protokół S/MIME jest stosowany (jak wspomniano już wcześniej) do zagwarantowania bezpieczeństwa jednostkom MIME. W poprzednim punkcie opisano, w jaki sposób zagwarantować usługę uwierzytelnienia z wykorzystaniem tego protokołu teraz kolej na usługę poufności.

Poufność ciała wiadomości SIP jest realizowana z wykorzystaniem szyfrowania, a nagłówków z wykorzystaniem tunelowania oraz szyfrowania. Tunelowanie nagłówków wiadomości polega na ukryciu ich w „wewnętrznej" wiadomości, podczas, gdy w nagłówkach „zewnętrznych" znajduje się tylko część zawartych w nich informacji. Szyfrowanie bazuje na technice klucza publicznego i powinno wykorzystywać jako funkcji skrótu: SHA-1, a jako mechanizmu szyfrującego 3DES (inne są również akceptowane).

7 Słabe punkty w architekturze bezpieczeństwa SIP w zaleceniu RFC 2543

Znane jest powiedzenie: „bezpieczeństwo systemu jest tak mocne jak mocne jest jego najsłabsze ogniwo". Nikt nie będzie wykorzystywał protokołu SIP do realizacji usługi telefonii, jeśli nie będzie on gwarantował satysfakcjonującego poziomu bezpieczeństwa. Jasną sprawą jest fakt, że skoro powstała druga wersja protokołu SIP to pierwsza łagodnie ujmując nie spełniała oczekiwań twórców oraz użytkowników. Głównie chodzi tu właśnie o aspekty bezpieczeństwa (wykazano wiele słabych punktów i niedociągnięć SIP z zalecenia RFC 2543 rozważanego w aspekcie zapewnienia dwóch podstawowych usług ochrony informacji).

Zastosowane tu mechanizmy wywodzące się z samego protokołu SIP (czyli bez grupy mechanizmów Hop-by-Hop) są niewystarczające, a zastosowane systemy kryptograficzne w dużej mierze przestarzałe w porównaniu z tymi uważanymi za należące do nowoczesnej kryptografii. Głównymi „grzechami" SIP w zaleceniu RFC 2543 jest:

- Zbyt duża opcjonalność w wyborze mechanizmów gwarantujących wymagane usługi oraz brak jasnego narzucenia i przypisania konkretnych mechanizmów bezpieczeństwa poszczególnym częściom drogi komunikacyjnej,
- Brak gwarancji integralności wiadomości przy uwierzytelnieniu typu End-to-End: ani *Basic* ani *D*igest, co automatycznie uniemożliwia spełnienia usługi uwierzytelnienia,
- Zastosowanie do uwierzytelnienia mechanizmów *Basic* (całkowita nie odporność na atak typu powtórka),
- Niewystarczające bezpieczeństwo z *SIP Digest* - działanie na zasadzie współdzielonego sekretu (tak samo *Basic*),
- Dla realizacji usługi poufności wykorzystanie głównie szyfrowania PGP. Obecnie nie jest to najbezpieczniejszy system kryptograficzny szczególnie w starszych wersjach głównie (brak systemu certyfikacji, niekiedy zbyt krótkie długości kluczy kryptograficznych). Kolejnym minusem jest fakt wykorzystania w PGP funkcji skrótu MD5 (przeprowadzono kompromitujący ją atak oraz długość jej skrótu – 128 bitów – jest na obecną chwilę zbyt mała).

8 Słabe punkty w architekturze bezpieczeństwa SIP w zaleceniu RFC 3261

Zalecenie RFC 3261 jest jeszcze zbyt nowe i dotychczas nie opublikowano udanych ataków na mechanizmy bezpieczeństwa w nim zawarte, ponieważ nie zaimplementowano dotychczas zbyt wiele komercyjnych systemów bazujących na protokole SIP w wersji drugiej. Wydaje się, że architektura bezpieczeństwa oferowana w tym zaleceniu jest poprawna, aczkolwiek nie wystrzeżono się kilku słabości (z których większość nie jest bezpośrednio związana z samym protokołem SIP). Przejdźmy jednak do ich analizy.

Przypomnijmy: w RFC 3261 nie ma uwierzytelnienia typu *Basic*, ale za to *Digest* jest obowiązkowe. Jest dodatkowo opcjonalne S/MIME realizujące uwierzytelnienie wzajemne.

W przypadku mechanizmów realizujących usługi *Hop-by-Hop* protokół TLS musi być obowiązkowo implementowany (ale działa on niestety tylko na TCP) lub IP Sec (wybór opcjonalny).

Pomimo, że mechanizmy bezpieczeństwa dostarczone w SIP w omawianym zaleceniu redukują ryzyko ataku posiada on klika słabości. Są one następujące:

- **SIP Digest** – pozostawione odkryte niektóre nagłówki (inaczej się nie da!), które muszą być wykorzystywane przez pośrednie urządzenia. Nie ma gwarancji integralności – można jedynie dodać kilka nagłówków, które następnie zostaną użyte do obliczenia skrótu.
- **S/MIME** – brak infrastruktury wymiany kluczy publicznych - powinien być jakiś system wymiany kluczy – ten zdefiniowany w wersji drugiej SIP jest nieodporny na atak typu *man-in-the-middle* (podobnie jak w innych systemach np. w SSH). W sytuacji, gdy atakujący przechwyci pierwszą wymianę kluczy pomiędzy komunikującymi się i będzie miał szansę przechwytywania całego dialogu między stronami komunikującymi się przeprowadzony atak można będzie uznać za udany.

  Druga rzecz – wykorzystanie S/MIME może owocować dużymi (w sensie objętości) wiadomościami.

  Kolejnym problemem jest fakt, iż jeśli na drodze wiadomości znajdzie się jakiś rzadki typ serwera sieciowego (nie typowy proxy), którego prawidłowe działanie zależy od możliwości dostępu i modyfikowaniu ciała wiadomości SIP to wtedy protokół S/MIME uniemożliwi prawidłowe funkcjonowanie takiego elementu sieciowego.

- **TLS (mechanizm nie wynikający bezpośrednio z SIP)** – nie funkcjonuje na UDP – tylko TCP – ciągłe, długo trwające połączenia TLS-over-TCP oraz uwierzytelnienie tylko serwerów, do których przylegają jednostki SIP (niedogodność ta występuje również w przypadku zalecenia RFC 2543).

Dla obu wersji protokołu SIP typem ataku, przed którym nie ma całkowitej ochrony jest atak typu blokowanie działania (Denial of Service). Bez względu na rodzaj zaimplementowanych mechanizmów bezpieczeństwa zawsze możliwe jest „zalanie" serwera (głównie chodzi tu o serwery proxy) poprzez wysyłanie nadmiernej ilości zwykle niepoprawnych wiadomości, w ten sposób powodując odmowę świadczenia usług, dla których dana jednostka została stworzona. W efekcie może to doprowadzić do stanu, gdy określony ruch pomiędzy stronami komunikującymi się nie będzie mógł poprawnie osiągnąć celu.

Niestety tego typu ataku nie da się wyeliminować całkowicie, ponieważ wiązałoby się to z ograniczeniem podstawowych funkcji serwerów sieciowych, dla których zostały one stworzone (trudno sobie wyobrazić np. serwer proxy nie obsługujący zgłoszeń połączeń).

Jedynym rozwiązaniem, które może w sposób satysfakcjonujący ograniczyć prawdopodobieństwo wystąpienia takiego ataku jest przeprowadzanie wzajemnego uwierzytelnienie serwerów proxy poprzez wykorzystanie protokołu TLS, który powinien być obowiązkowo uwzględniany przy aplikacjach bazujących na drugiej wersji protokołu SIP.

9 Doświadczenia praktyczne

W ramach potwierdzenia wyników przeprowadzonych analiz mechanizmów bezpieczeństwa dla protokołu SIP w dwóch wersjach przeprowadzono badania praktyczne. Ich przebieg, wykorzystane aplikacje oraz testy zostaną zaprezentowane poniżej.

9.1 Przebieg badań

Całość przeprowadzonych działań praktycznych polegała na wykonaniu szeregu testów na aplikacjach będących implementacjami Agentów Użytkownika SIP. Pierwotnym celem było zbadanie odporności na różnego rodzaju próby ataków mechanizmów bezpieczeństwa w nich zawartych. W zamierzeniu miała być to po prostu symulacja potencjalnych, celowych prób działań atakującego i na tej podstawie ocena umiejętności radzenia sobie w przypadku ich wystąpienia.

Niestety ilość i różnorodność ataków na protokół SIP jest znaczna i testując samą aplikację Agenta Użytkownika SIP nie odda się całego spektrum potencjalnych zagrożeń. Jednakże uznając, że w praktyce użytkownik nie ma zbyt wielkiego wpływu na bezpieczeństwo innych komponentów architektury funkcjonalnej SIP tylko na SIP UA, właśnie na testowanie tego elementu funkcjonalnego położono nacisk.

Jeśli chodzi o same testy, zostały one dobrane tak, aby oddawały jak najwięcej klas zagrożeń – kładąc szczególny nacisk na ataki najłatwiejsze do wykonania a tym samym najbardziej prawdopodobne. Wyszliśmy z założenia, iż intruz ma możliwość (co najmniej) przeprowadzenia podsłuchania pakietów (co w obecnych czasach nie sprawia większych trudności) i w ten sposób generowania fałszywych wiadomości.

Niestety po zapoznaniu się z dostępnymi nam aplikacjami będącymi implementacjami Agenta Użytkownika pierwotny cel testowania musiał zostać zmodyfikowany. Stało się tak, ponieważ:

- Żadna z testowanych aplikacji (prócz jednej) nie miała zaimplementowanego, choć najprostszego mechanizmu bezpieczeństwa.
- Wszystkie aplikacje bazowały wyłącznie na starym zaleceniu SIP (RFC 2543).

W związku z faktami, które przytoczono powyżej testowanie Agentów Użytkownika ograniczyło się do osiągnięcia dwóch celów:

- Zbadania zgodności implementacji wybranych aplikacji z zaleceniem, na którym bazuje (w przypadku testowanych aplikacji jest to RFC 2543).
- Wykazanie wyższości programu, w którym został zaimplementowany, choć prosty mechanizm bezpieczeństwa opisany w zaleceniu nad aplikacją nie posiadającą żadnych wspomnianych mechanizmów (zatem jedynym mechanizmem, którego skuteczność w praktyce przetestowano był SIP Digest).

Postacie wykorzystanych w doświadczeniach testów zostały umieszczone w plikach tekstowych tak, aby umożliwić ich łatwe tworzenie, dodawanie oraz modyfikację. Ich postać zostanie przedstawiona i opisana w dalszej części pracy (jest także zawarta w [14]).

Do przeprowadzenia doświadczeń wykorzystana została autorska, pomocnicza aplikacja S2C (SIP Security Call Checker), która pozwoli na uwidocznienie niektórych słabości w architekturze bezpieczeństwa badanych aplikacji oraz zbadanie zgodności implementacji tychże z zaleceniem. Jej głównym zadaniem jest wysłanie określonego testu do wskazanego Agenta Użytkownika SIP, a następnie zbadanie jego reakcji.

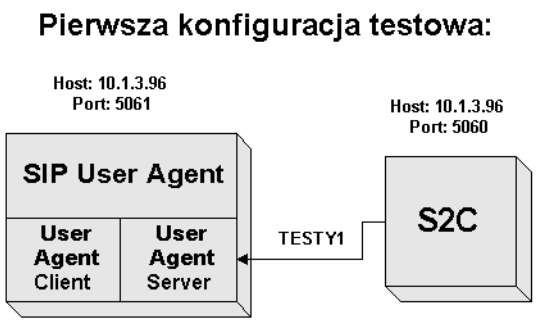
9.2 Opis doświadczeń i wykorzystanych testów

Postać testów została napisana specjalnie w taki sposób, aby oddać specyfikę wymiany wiadomości sygnalizacyjnych pomiędzy dwoma Agentami Użytkownika SIP. Przy ich opracowaniu bazowano na testach dla SIP, które zostały stworzone przez twórców SIP'a: Neila Deasona, Andersa Kristensena, Jonathana Rosenberga oraz Henninga Schulzrinna.

Całość testów została przeprowadzona na jednym komputerze z zainstalowanym systemem operacyjnym Windows XP oraz kartą sieciową (jednak ma przeciwwskazań ani żadnych problemów, aby przeprowadzić je w sieci np. LAN). Zainstalowano na nim wszystkie potrzebne do doświadczeń aplikacje do testowanie (przedstawione w następnej części pracy) jak i swoje narzędzie testujące (S2C). W celu symulacji ataków na sygnalizację SIP zastosowano odpowiednie konfiguracje doświadczeń, które przedstawiamy poniżej.

9.2.1 Konfiguracja doświadczeń

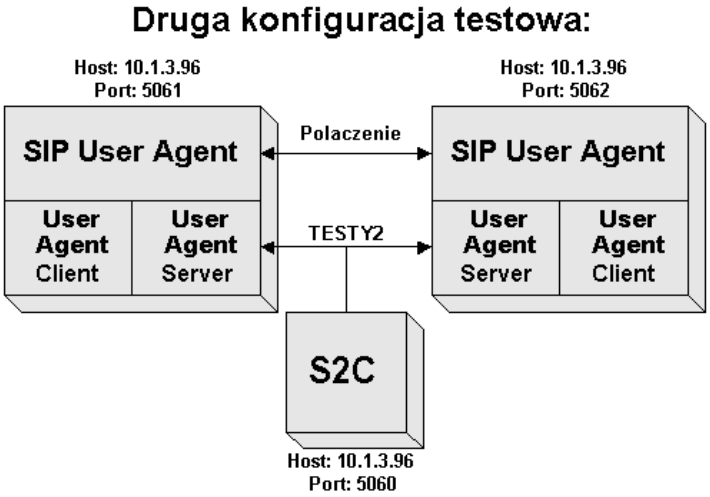
a) Konfiguracja1 - w tej konfiguracji badane było zachowanie testowanej aplikacji, do której przesyła się odpowiednio sporządzone wiadomości-testy. Ma to na celu zarówno sprawdzenie zgodności zachowań z zaleceniem RFC 2543 oraz symulację ataku, w którym intruz może się podszyć pod czyjąś tożsamość.



Rys. 10. Konfiguracja testowa pierwsza

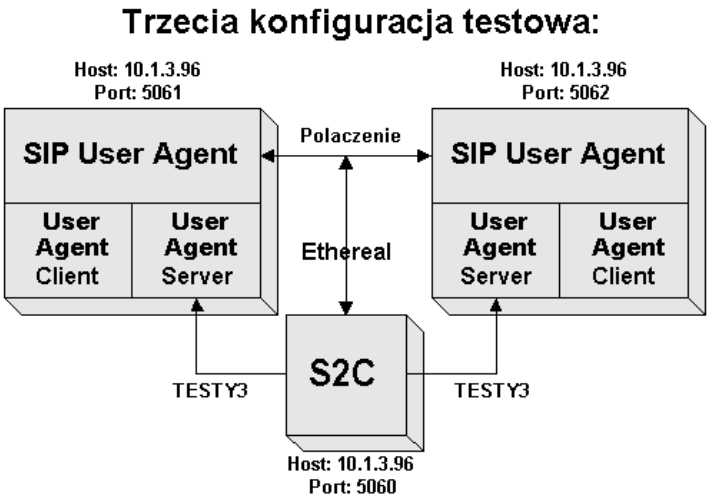
b) Konfiguracja2 - w tym przypadku nawiązywane było połączenie pomiędzy dwoma wybranymi aplikacjami Agenta Użytkownika SIP, a następnie za pomocą aplikacji S2C oraz poprzez określone testy

próbowano wpłynąć w negatywny sposób na wymianę wiadomości sygnalizacyjnych/pakietów RTP (wiadomości-testy były raz wysyłane do jednej, raz do drugiej testowanej aplikacji). Była to próba symulacji ataku aktywnego, w którym intruz nie podsłuchuje komunikujących się między sobą agentów, jednak może wysyłać odpowiednio zaadresowane i sporządzone wiadomości w celu uniemożliwienia nawiązania lub przerwania trwającego połączenia.



Rys. 11. Konfiguracja testowa druga

c) Konfiguracja3 – Sytuacja tutaj jest podobna do przypadku opisanego w poprzedniej konfiguracji. Badana była tu również odporność połączenia na przeprowadzane na nim prób ataków. Główną różnicą pomiędzy doświadczeniami w punktach b i c było założenie, iż atakujący ma możliwość podsłuchiwania wiadomości sygnalizacyjnych (do tego celu wykorzystano popularną aplikację *Ethereal*) oraz odpowiednią ich modyfikację – głównie chodziło w tym przypadku o pokazanie jak łatwo wpłynąć na połączenie w przypadku nie wykorzystywania żadnych mechanizmów bezpieczeństwa (poprzez znajomość nagłówka: *Call-Id*).



Rys. 12. Konfiguracja testowa trzecia

9.2.2 Opis treści testów

Dla każdej konfiguracji przeprowadzane testy nie były identyczne. Odpowiedni ich dobór został dokonany w zależności od rodzaju wybranej konfiguracji.

Dodatkowo należy w tym miejscu koniecznie jeszcze raz pokreślić, iż postać testów dla podanych powyżej konfiguracji nie skupia się na badaniu opisanych wcześniej mechanizmów bezpieczeństwa (jest jeden wyjątek w przypadku aplikacji firmy Helmsman), lecz głównie na:

- Próbach przerwania lub zakłócenia w jak najprostszy sposób trwającego lub zestawianego połączenia. Celem takiego podejścia jest wykazanie potrzeby implementacji opisanych w zaleceniach mechanizmów bezpieczeństwa – nawet tych prostych, gdyż finalnie mogą one utrudnić, choć trochę możliwość ingerencji w komunikację intruza.
- Analizie zgodności implementacji Agenta Użytkownika z zaleceniem na tych obszarach, które mogą mieć krytyczne znaczenie dla bezpieczeństwa zarówno działania samej aplikacji jak i możliwej wymiany wiadomości sygnalizacyjnych.

9.2.3 Grupa testów do Konfiguracji1

Głównym celem tej grupy testów jest zbadanie zgodności zachowania badanej aplikacji (pod kątem odbioru celowo źle sformowanych wiadomości sygnalizacyjnych) z zaleceniem, na którym bazuje (RFC 2543). Testy te mają wykazać ewentualne braki implementacyjne mogące wspomagać działania potencjalnego intruza, który znając takie „dziury" może utrudniać prawidłowe funkcjonowanie Agenta Użytkownika, a w rezultacie obniżyć *QoS (Quality of Service)* całej usługi. Wszystkie treści wykorzystanych w tej konfiguracji testów zostały zaprezentowane w [14]

9.2.4 Grupa testów do Konfiguracji2

Celem przeprowadzania tej grupy testów jest chęć sprawdzenia szans intruza, który będzie próbował przerwać lub zakłócić istniejące połączenie. Zakładamy, iż będzie on działał „na ślepo" tzn. zna tylko adres SIP jednej z komunikujących się stron. Wszystkie treści wykorzystanych w tej konfiguracji testów zostały zaprezentowane w [14].

9.2.5 Grupa testów do Konfiguracji3

Celem przeprowadzania tej grupy testów jest chęć sprawdzenia szans intruza, który będzie próbował przerwać lub zakłócić istniejące połączenie. Zakładamy, że atakujący za pomocą aplikacji podsłuchującej pakiety jest wstanie odkryć zawartość przesyłanych wiadomości. Wszystkie treści wykorzystanych w tej konfiguracji testów zostały zaprezentowane w [14] (są to prawie te same testy, co w przypadku Konfiguracji2).

9.2.6 Testy mechanizmu SIP Digest

Do przeprowadzenia doświadczeń w tym punkcie wykorzystamy konfigurację trzecią (tam gdzie wykorzystuje się aplikację *Ethereal*) oraz odpowiednio ułożone testy, które będą odpowiadały sytuacji:

a) Podobnej jak w Konfiguracji2, gdy intruz przeprowadza ataki „na ślepo" – znając tylko adres SIP Agenta Użytkownika.

b) Podobnej jak w Konfiguracji3, gdy atakujący za pomocą aplikacji podsłuchującej pakiety jest wstanie odkryć zawartość przesyłanych wiadomości.

Zastosowane testy:

a) Dowolne wybrane z grupy testów dla Konfiguracji1 oraz próba nawiązania prawidłowego połączenia w czyimś imieniu,

b) Próba podszycia się pod innego użytkownika z wykorzystaniem informacji uzyskanych przy pomocy aplikacji *Ethereal* – zastosowanie poprawnie sformułowanej wiadomości ze zmodyfikowanymi polami pobranymi z przechwyconych wiadomości.

9.3 Testowane aplikacje SIP UA

W Internecie dostępnych było około dziesięciu darmowych aplikacji będących implementacjami Agenta Użytkownika SIP. Przy wyborze programów do testowania kierowaliśmy się głównie ich dostępnością oraz tym, by każdy z nich był firmowany przez innych twórców (by uzyskać jak najszerszy przegląd dostępnych UA). Wszystkie dostępne nam aplikacje SIP UA zostały zaimplementowane zgodnie z RFC2543. **Nie mieliśmy możliwości przetestowania żadnych aplikacji bazujących na RFC3261.**

Do celów badawczych pokazujących możliwości zrealizowanego przez nas narzędzia wybrano następujące dostępne darmowe programy:

- **Helmsman User Agent 3.0.6** firmy Helmsman,
- **eStara SoftPHONE 3.0** – firmy eStara,
- **Siemens Communication System Client v.1.0** firmy Siemens,
- **Magellan 4.0** opracowany w Instytucie Telekomunikacji PW,
- **Hughes SIP User Agent (E-Z Phone)** firmy Hughes Software Systems,
- **Vovida SIP UA 1.0.2 -** Columbia University.

Należy w tym miejscu po raz kolejny podkreślić, iż z pośród wszystkich aplikacji wymienionych powyżej tylko Helmsman User Agent 3.0.6 posiada celowo zaimplementowany jakikolwiek mechanizm bezpieczeństwa (w tym przypadku jest to możliwość uwierzytelnienia *SIP Digest*). Reszta programów została zubożona on ten krytyczny obszar stanowiący o jakości oraz funkcjonalności zarówno samego Agenta Użytkownika jak i całej usługi Voice over Internet Protocol.

9.4 Analiza wyników przeprowadzonych testów z grup 1,2 oraz 3

Wyboru najlepszej pod względem zapewnienia bezpieczeństwa aplikacji będącej implementacją Agenta Użytkownika SIP dokonano po analizie ocen wszystkich testów dla poszczególnych aplikacji (ilości zaliczonych testów). W trakcie doświadczeń przeprowadzono na każdej aplikacji 25 różnych testów.

Ocenianie testów:

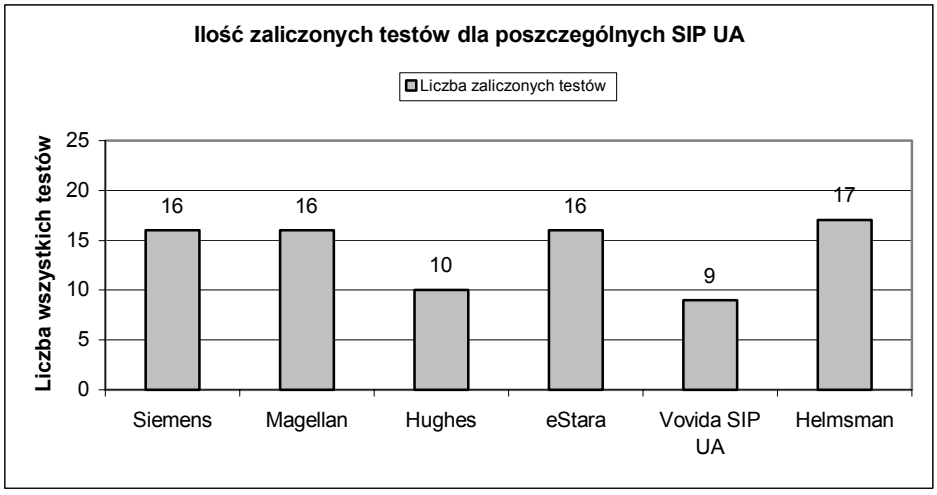
„**+**" – test zaliczony – zachowanie zgodne z oczekiwaniami;

„**+/-**" – test częściowo zaliczony (sytuacja, gdy np. nie nastąpiło przerwanie połączenia, ale nie ma sygnalizacji błędu);

„**-**" – test nie zaliczony.

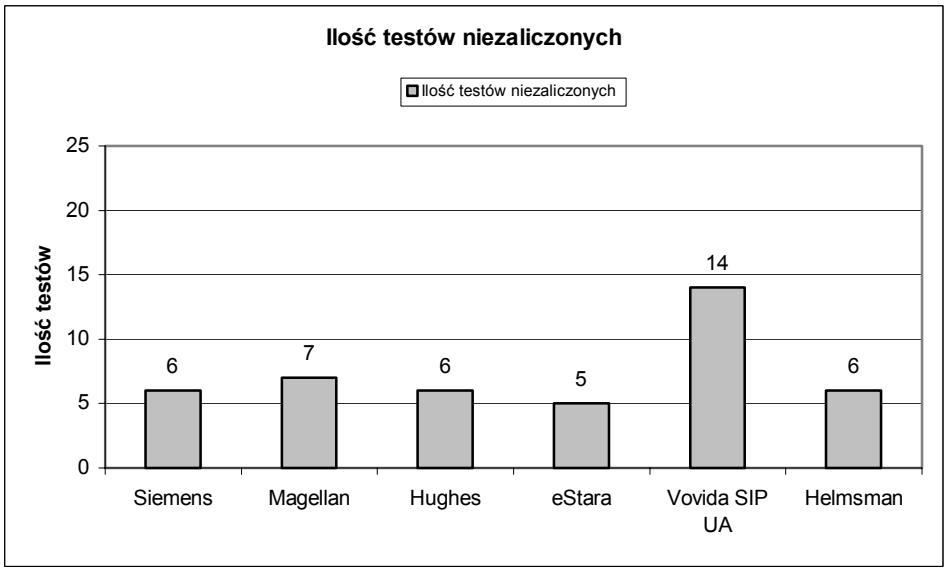
9.4.1 Uzyskane wyniki

Suma zaliczonych testów dla poszczególnych aplikacji przedstawia się następująco:



Wykres 1. Ilość zaliczonych testów dla poszczególnych SIP UA

Natomiast sumę testów nie zaliczonych dla poszczególnych aplikacji obrazuje wykres:



Wykres 2. Ilość testów nie zaliczonych dla poszczególnych SIP UA

Analiza powyższych wykresów prowadzi do wniosku, iż spośród sześciu testowanych aplikacji SIP UA na wyróżnienie zasługuje aplikacja firmy Helmsman. Sprawdziła się ona zarówno pod względem zaliczonych testów, jak pod względem interfejsu i komunikacji z użytkownikiem - jest on godny uwagi – prosty, intuicyjny i funkcjonalny.

Żadna z aplikacji nie zaliczyła wszystkich testów, co dowodzi jak łatwo wcale nie dużymi środkami osiągnąć udany atak na tego rodzaju program. Wielu Agentów Użytkownika w ogóle nie reagowało na wiadomość testującą, mimo tego, że działając zgodnie z zaleceniem, na którym bazują powinny zasygnalizować wystąpienie błędu.

9.4.2 Wyniki i analiza testów mechanizmu SIP Digest

Jak już podkreślano jedyną aplikacją ze wszystkich dostępnych, która miała zaimplementowany jakiekolwiek mechanizm bezpieczeństwa była aplikacja Helmsman User Agent 3.0.6, co uwzględniając wyniki testów przeprowadzone w poprzednim punkcie stawiają tą aplikację w bardzo korzystnym świetle i pozwalają, tym bardziej, na wyróżnienie jej pośród innych testowanych w tej pracy.

Zastosowanym w niej mechanizmem bezpieczeństwa jest *SIP Digest*. Opcja wykorzystania tego mechanizmu nie jest włączona domyślnie. Dlatego też testy przeprowadzone wcześniej na tej aplikacji nie uwzględniały takiego przypadku.

W przypadku wysłania poprawnej wiadomości bez zawartych w niej danych uwierzytelniających głównie otrzymano odpowiedzi: 401 Unauthorized. Jest to zgodne z oczekiwaniami oraz zasadą działania SIP Digest.

Tą samą odpowiedź otrzymano również w przypadku próby nawiązania połączenia poprawną wiadomością (poszycie się).

Aby odnaleźć wartości potrzebne do obliczenia skrótu przy uwierzytelnieniu SIP Digest dla aplikacji Helmsman UA możliwe są dwa rozwiązania:

- Odczytanie ich z odpowiedzi 401 Unauthorized, która przychodzi do nas zwrotnie po wysłaniu np. metody INVITE (nagłówek WWW-Authenticate),
- Zdobycie szukanych wartości za pomocą podsłuchania wiadomości (aplikacja *Ethereal*).

Niestety, dla atakującego, aby wymiana uwierzytelniająca zakończyła się sukcesem to skrót wyliczony z tych wartości jak również z nazwy użytkownika oraz jego hasła musi zostać wysłany w „poprawionej" wiadomości w nagłówku *Authorization*. Mimo podsłuchania tego nagłówka intruz nie ma wystarczających danych ku temu by wygenerować oczekiwany skrót.

Jedynym sposobem złamania takiego mechanizmu uwierzytelnienia jest zdobycie brakujących danych użytkownika. Dodatkowo wartość pola *nonce* jest różna dla każdej wymiany uwierzytelniającej.

Podsumowując nie jest to rozwiązanie idealne, ale w wydatny sposób poprawia bezpieczeństwo badanego Agenta Użytkownika. Widać teraz jak na dłoni, iż atakujący zmuszony jest do dużo większego wysiłku przy zastosowaniu nawet tak nieidealnego mechanizmu.

W innym przypadku naprawdę nie jest trudno wpłynąć negatywnie na samą aplikację lub połączenie – atakujący potrzebuje naprawdę minimalnej wiedzy w postaci odpowiedniej treści wiadomości oraz narzędzia umożliwiającego jej wysłanie.

9.5 Słabości implementacji komercyjnych SIP UA ujawnione przez organizację CERT

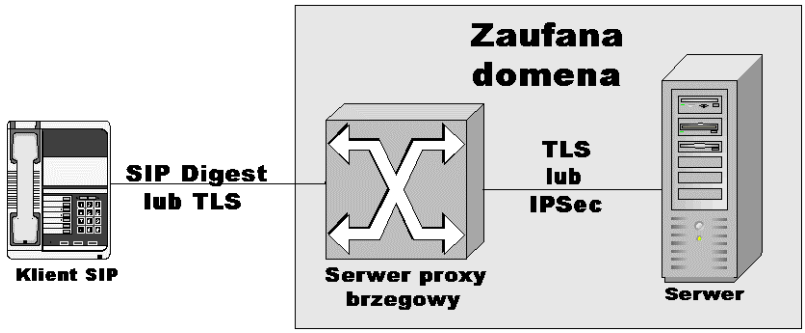
Po opracowaniu metody testowania oraz przeprowadzeniu zaprezentowanych testów bezpieczeństwa na dostępnych aplikacjach SIP UA w ramach pracy dyplomowej - 21 lutego 2003 roku znana organizacja CERT w artykule *„CA-2003-06 Multiple vulnerabilities in implementations of the Session Initiation Protocol (SIP)"* opublikowała niezależne wyniki specjalnie opracowanych testów PROTOS, których cel oraz natura były podobne do przeprowadzonych przez autorów. Jako, że testy odbywały się były w laboratoriach ich zasięg oraz możliwości były większe (m.in. testowane również inne elementy architektury funkcjonalnej SIP tj. serwery proxy, redirect oraz registrar). Przeprowadzone tam doświadczenia wykazały głównie podobne błędy implementacyjne w testowanych SIP UA opartym na zaleceniu RFC 3261. Stąd należy wnioskować, iż nawet najbezpieczniejszy protokół, jeśli zostanie źle zaimplementowany nie gwarantuje żadnego poziomu bezpieczeństwa. Powtórzyła się, zatem sytuacja, która została opisana w tym artykule.

Należy, więc przed zakupem komercyjnego oprogramowania dla VoIP bazującym na protokole SIP dowiedzieć się dokładnie, czy i jakie mechanizmy bezpieczeństwa zostały tam zaimplementowane. Nie dopilnowanie tego może mieć krytyczne znaczenie dla bezpieczeństwa całej komunikacji przeprowadzanej za pomocą takiej aplikacji.

10 Przykład bezpiecznej konfiguracji usługi VoIP bazującej na SIP

Przy projektowaniu bezpiecznej konfiguracji dla usługi Voice over Internet Protocol wykorzystującej jako protokół sygnalizacyjny protokół SIP należy umiejętnie dobierać dostępne mechanizmy bezpieczeństwa oferowane przez ten protokół, jak również niezbędne jest wykorzystanie z mechanizmów oferowanych przez warstwy niższe (patrz opis teoretyczny).

Przykładem bezpiecznej konfiguracji zgodnej z zaleceniem RFC3261 może być system przedstawiony na rysunku poniżej.



Rys. 13. Przykład bezpiecznej konfiguracji usługi VoIP opartej na SIP z zalecenia RFC3261

Mechanizm wykorzystany do komunikacji klient SIP - serwer proxy: **TLS** lub **SIP Digest**.

Mechanizm wykorzystany do komunikacji brzegowy serwer proxy – serwer docelowy: **TLS** lub **IPSec**.

Niezbędne założenia: brzegowy serwer proxy „dopuszcza" tylko żądania od uwierzytelnionych klientów (firewall). Serwer docelowy akceptuje tylko ruch przenoszony przez brzegowy serwer proxy.

Niepodważalnymi **zaletami** przedstawionej w tym punkcie konfiguracji jest (oprócz bezpiecznej wymiany wiadomości sygnalizacyjnych):

- Umożliwienie przeprowadzenia brzegowemu serwerowi proxy uwierzytelnienia klienta typu Hop-by-Hop,
- Oddzielenie logiki realizowania usługi uwierzytelnienia od logiki danej usługi.

Wadą takiego rozwiązania może okazać się fakt, iż wszystkie elementy usługowe muszą ufać serwerowi proxy oraz to, że serwer docelowy może nie znać prawdziwej tożsamości klienta.

11 Podsumowanie i wnioski

W niniejszym artykule w części teoretycznej zostały zebrane, przeanalizowane oraz ocenione mechanizmy bezpieczeństwa tworzące architekturę bezpieczeństwa SIP dla dwóch zaleceń: RFC 2543 oraz RFC 3261. Wykazano potencjalne zagrożenia, które mogą wystąpić dla SIP w wersji pierwszej (RFC 2543). Pokazano również jak rozwiązano te problemy zastępując pierwszą wersję SIP drugą (RFC 3261). Głównym wnioskiem z tej części jest możliwość utworzenia potencjalnie bezpiecznej architektury SIP – nowe zalecenie określa mechanizmy, które powinny umożliwić wydajną i bezpieczną wymianę wiadomości sygnalizacyjnych. Wszystko pozostaje teraz w rękach implementujących, ponieważ nawet najlepsze plany są niczym w przypadku, gdy nie zostaną one poprawnie zrealizowane. Przykład takiej architektury został podany w rozdziale 10.

Sprawdzenie za pomocą odpowiednio dobranych testów-wiadomości sześciu dostępnych, darmowych Agentów Użytkownika SIP bazujących na RFC 2543 nie dostarczyło niestety optymistycznych wyników. Z sześciu aplikacji tylko jedna (Helmsman User Agent) miała zaimplementowany jakikolwiek mechanizm bezpieczeństwa (SIP Digest). To wystarczyło, aby odpowiednio utrudnić „domowy" atak na sygnalizację SIP. Reszta aplikacji niestety nie została zrealizowana w sposób należyty.

Część winy za taki stan rzeczy ponoszą sami twórcy zalecenia RFC 2543, gdyż mechanizm SIP Digest (podobnie jak i Basic) nie jest tam narzucony (wymagany) – jest to tylko opcja. Drugą przyczyną takiego stanu rzeczy jest fakt, iż większą część z tych aplikacji stanowią wersje testowe programów lub takie, które mają zachęcić do kupna jego pełnej wersji. Możliwe, że kupując pełną wersję produktu otrzymuje się program, razem z zaimplementowanymi mechanizmami bezpieczeństwa należy, więc to bezwzględnie sprawdzić przed zakupem.

Jak pokazaliśmy w swoich doświadczeniach istnieje wiele prostych metod, za pomocą których można zakłócić wymianę sygnalizacji, przerwać połączenie lub wpłynąć negatywnie na samą aplikację.

Dowodzi to konieczności uwzględnienia w trakcie implementacji Agenta Użytkownika odpowiednich (nawet najprostszych) mechanizmów gwarantujących bezpieczeństwo. Dlatego też m.in. w drugiej wersji protokołu SIP zapisano obowiązek realizacji mechanizmu SIP Digest.

Jeśli chodzi o bezpieczeństwo sygnalizacji systemu VoIP bazującego na protokole SIP to z pewnością druga jego wersja (bazująca na RFC 3261) gwarantuje więcej trafnych i nowoczesnych mechanizmów kryptograficznych limitujących ilość przeprowadzenia udanych ataków na taki system. Nie jest to dziwne, gdyż właśnie jedną z głównych przesłanek jej stworzenia były luki bezpieczeństwa wersji pierwszej.

Z kolei, aby prawidłowo zabezpieczyć wymianę wiadomości sygnalizacyjnych z wykorzystaniem protokołu SIP należy umiejętnie dobierać dostępne mechanizmy bezpieczeństwa. A uściślając należy zdefiniować jak jednostki funkcjonalne SIP mogą dokonywać wyboru pomiędzy odpowiednimi mechanizmami podczas komunikacji tak, aby być w stanie zagwarantować podstawowe usługi ochrony informacji i komunikacji zdefiniowane w przyjętym kryterium.

Architektura bezpieczeństwa protokołu SIP przedstawiona w RFC 3261 powinna gwarantować wystarczający poziom bezpieczeństwa tzn. taki, który zminimalizuje prawdopodobieństwo przeprowadzenia udanego ataku na protokół sygnalizacyjny SIP. Na razie jednak zalecenie to jest zbyt nowe i jeszcze nie ma żadnych opublikowanych możliwych ataków na mechanizmy bezpieczeństwa tam opisane. Ale powód braku potencjalnych ataków może być również inny – niewykluczone, że doświadczenia zebrane przy tworzeniu SIP w wersji pierwszej jak i testowaniu pierwszych aplikacji SIP zaowocowały właściwym doborem mechanizmów bezpieczeństwa dla drugiej wersji tegoż protokołu.

Literatura

[1] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg – „SIP: Session Initiation Protocol" – Request for Comments nr. 3261 lipiec 2002

[2] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg – „SIP: Session Initiation Protocol" – Request for Comments nr. 2543 marzec 1999

[3] S. Salsano, L. Veltri, D. Papalilo – "SIP Security Issues: The SIP Authentication Procedure and its Processing Load" – IEEE Network, vol. 16, vol. 6, November 2002

[4] W. Stallings - "Cryptography and Network Security : Principles and Practice, Second Edition". Prentice-Hall, June 1998.

[5] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart. "HTTP Authentication : Basic and Digest Access Authentication". Request For Comments 2617. Internet Engineering Task Force, June 1999.

[6] T. Dierks, C. Allen. "The TLS protocol version 1.0". Request For Comments 2246. Internet Engineering Task Force, January 1999.

[7] R. Rivest. "The MD5 Message-Digest Algorithm". Request For Comments 1321. Internet Engineering Task Force, April 1992.
[8] T. Berners-Lee, R. Fielding, H. Frystyk. "Hypertext Transfer Protocol -- HTTP/1.0". Request For Comments 1945. Internet Engineering Task Force, May 1996.
[9] N. Borenstein, N. Freed. "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies ". Request For Comments 1341. Internet Engineering Task Force, June 1992.
[10] B. Ramsdell. "S/MIME Version 3 Message Specification". Request For Comments 2633. Internet Engineering Task Force, June 1999.
[11] J. Callas, L. Donnerhacke, H. Finney, R. Thayer, "Open PGP Message Format". Request For Comments 2440. Internet Engineering Task Force, November 1998.
[12] P. Gajowniczek, M. Średniawa - „Voice over IP – Wykorzystanie techniki IP do przesyłania głosu" - CITCOM-PW październik 1999
[13] H. Sinnreich, A. Johnston – „Internet Communications Using SIP" – Wiley Computer Publishing
[14] W. Mazurczyk – „Bezpieczeństwo Voice over IP opartego na SIP " – VII Krajowa Konferencja Zastosowań Kryptografii Enigma'2003, Warszawa, maj 2003