Bezpieczeństwo i anonimowość w internecie

autor: Jerzy Wawro, styczeń 2019

W opracowaniu wykorzystano fragmenty pracy autora (wkrótce pojawi się na stronach verbu.pl) pod tytułem „Identyfikacja i autoryzacja. Poradnik programisty i administratora”

I. Szkoła wobec zagrożeń płynących z internetu

Co chroni tradycyjne społeczeństwo?
- Moralność
- Reguły życia w społeczeństwie
- System prawa + państwo
- Środki bezpieczeństwa

Internet zmniejsza skuteczność ochrony
- Moralność: złudne poczucie bezkarności, obce kultury (globalna wioska)
- Zamiast społeczeństwa – społeczności bez silnych więzi i relacji osobowych
- Niska skuteczność państwa (rządzą ponadnarodowe korporacje)

Tradycyjne zabezpieczenia nie są skuteczne

Najważniejsze zagrożenia
- Katastrofy
- Łamanie zasad, przestępczość
- Nieostrożność (wypadki)

Nowe zagrożenia
- Demoralizacja (pornografia / sexting, obce wzorce kulturowe, fałszywe autorytety)
- Stalking (nękanie) – czasem prowadzi do samobójstwa
- Utrata prywatności, utrata danych (w tym danych osobowych, danych wrażliwych) – może być źródłem szantażu
- Kradzieże (na przykład pieniędzy z konta)
- Masowe powielanie kłamstwa (postprawda)

Skąd się biorą te zagrożenia?

- Internetowi chuligani zwani „hakerami”
- Biznes
- Cele militarne i makroekonomiczne
- Cele polityczne

Żyjemy w globalnej wiosce – także zagrożenia są globalne

Internetowe Chuligaństwo

Słowo „haker” oznaczało kiedyś człowieka zgłębiającego tajniki rozwiązań technicznych dla ich ulepszenia, rozbudowy, czy zastosowania w nietypowych rozwiązaniach.

Obecnie – synonim chuligana. Przykład: haker przejął kontrolę nad kamerą w komputerze Miss Nastolatek USA, Cassidy Wolf . Przez rok czasu ją podglądał bez jej wiedzy, podsłuchiwał jej

intymne rozmowy z rodziną i robił zdjęcia, gdy w sypialni zmieniała ubrania
Obecnie do głównych zagrożeń należą (http://www.egospodarka.pl/art/galeria/151082,TOP-10-zagrozen-w-sieci,3,12,1.html):

1. Cryptojacking – zmuszanie do udziału w kopaniu kryptowaluty.
2. Włamania pod pozorem fałszywej pomocy technicznej
3. Ransomware - szyfrowanie dysku i żądanie okupu https://pl.wikipedia.org/wiki/Ransomware

Ataki internetowe jako część biznesu

Najprostszym przykładem jest rozsyłanie spamu. To jest usługa płatna. Serwery poczty są obecnie głównym celem ataków. Dlatego - jeszcze raz **musisz stosować silne hasła**.

Wywiad gospodarczy

W Polsce trwa wdrażanie systemu sprawozdawczości podatkowej JPK, Krajowa Izba Rozliczeniowa (KIR) realizuje system STIR gromadzący dane o podmiotach gospodarczych dostępne w bankach. BIK – biuro informacji kredytowej.
Gromadzone w jednym miejscu informacje biznesowe o wielu podmiotach to duże niebezpieczeństwo wykorzystania ich niezgodnego z prawem. W USA oficjalnie telekomy mogą spieniężać dane o klientach. U nas jest to szara strefa – ale wszyscy doświadczamy nękania telefonami w ramach telemarketingu – bez względu na to, czy kiedykolwiek wyraziliśmy zgodę na takie kontakty.

Dokumenty ujawnione przez Snowdena pokazują, że NSA podsłuchiwała nie tylko polityków, ale także organizacje międzynarodowe i liderów biznesu. Szpiegowane były (są?) transakcje finansowe, w szczególności przy pomocy kart kredytowych. Monitorowano system Visa oraz system płatności międzynarodowych SWIFT (projekt "Follow the Money"). Agencja tworzy bazę danych finansowych o nazwie Tracfin do przechowywania informacji zebranych od instytucji finansowych. W 2011 r. baza danych miała 180 milionów wpisów, z których 84 procent to transakcje kartami kredytowymi..

Cybernetyczna wojna ekonomiczna

Celem hakerskich ataków NSA były firmy i instytucje Chin. Stany Zjednoczone ukradły ogromne ilości ważnych informacji wywiadowczych z Chin i innych krajów. Włamano się między innymi do serwerów z Tsinghua University, najbardziej prestiżowej uczelni Chin.

Przykłady danych gromadzonych przez USA
- dane prawie 5 miliardów połączeń telefonii komórkowej na całym świecie każdego dnia;
- komórki kanclerz Niemiec Angeli Merkel przez co najmniej 10 lat;
- podłączenie do głównych łączy centrów danych Yahoo oraz Google i kradzież w ten sposób danych setek milionów klientów;
- aplikacje do telefonów komórkowych w celu kradzieży danych osobowych;
- operacjach szpiegowskich na dużą skalę zarówno przeciw chińskim przywódcom i chińskim instytutom badawczym i chińskim firmom – w tym gigantowi Huawei.

Stany Zjednoczone również pobiera informacje z gier komputerowych. Gry, takie jak "World of

Warcraft" i "Second Life" są używane w celu szpiegowania graczy. Powszechnie wiadomo, że większość graczy w tych grach to Chińczycy. Także chińskie komunikatory i poczta (China Mobile), były inwigilowane przez NSA.

Tymczasem jako główne źródło zagrożeń przedstawia się osławionych „rosyjskich hakerów". Według WSJ: *Rosyjskim cyberwłamywaczom udało się wykraść nie tylko dane dotyczące cyberobrony USA, ale także informacje o metodach stosowanych przez NSA w inwigilacji systemów informatycznych obcych państw.*
Kradzież tych informacji pozwoliła Moskwie na penetrację amerykańskich systemów informatycznych oraz na wzmocnienie obrony swoich własnych systemów przed analogicznymi próbami amerykańskiego wywiadu.
Zdaniem ekspertów cytowanych przez portal "WSJ" rosyjskim hakerom udało się wykraść cybernetyczne tajemnice władz amerykańskich poprzez wykorzystanie programu antywirusowego rosyjskiej firmy Kaspersky Lab w charakterze "tylnych drzwi" do komputera prywatnej firmy pracującej na zlecenie NSA.

Kłamstwa i polityka

„**post-prawda**" - pojęcie to zrobiło zawrotną karierę po zwycięstwie Donalda Trumpa w USA. Oskarżono go, że zwycięstwo to zapewnił sobie, posługując się kłamstwami.

Dlaczego Trump łamie wyborcze obietnice? Można postawić dwie hipotezy: „głębokie państwo" jest zbyt silne i Trump musi się mu podporządkować (bo na przykład mają na niego „haki") lub w czasie kampanii wyborczej obecny prezydent rzeczywiście kierował się jedynie strategią, mówiąc to co wyborcy chcieli usłyszeć.

Amerykański naukowiec Michał Kosiński stworzył model analizy danych („big data") który „na podstawie dziesięciu polubień z Faceboka jest w stanie ocenić daną osobę lepiej, aniżeli przeciętny kolega z pracy. 70 polubień wystarcza, aby przewidzieć znajomość charakteru danej osoby lepiej, niż jej przyjaciel, 150 lepiej, niż rodzice, przy 300 polubieniach maszyna jest w stanie przewidzieć zachowanie danej osoby lepiej, aniżeli jej partner. Przy jeszcze większej ilości polubień udaje się nawet przebić to, co ludzie sami myślą, że wiedzą o sobie. W dniu w którym Kosiński publikuje te odkrycia otrzymuje dwa telefony. Jeden z groźba kroków prawnych oraz jeden z ofertą pracy. Oba od Facebooka".

Firma SCL (Strategic Communications Laboratories), która nawiązała kontakt z Kosińskim jest firmą-matką dla Cambridge Analytica. To jest ta firma, która zorganizowała wyborczą kampanię internetową dla Trumpa i dla Brexitu.

Głównym polem walki jest praktycznie bezbronna część każdego społeczeństwa - 20% „naiwnych" zidentyfikowanych przez Huxley'a. To oni są bardziej podatni na propagandę, otwierają bezmyślnie załączniki w mailach, wchodzą na fałszywe strony zmiany haseł etc…
20% wystarczy by rządzić?
Tak – jeśli większość uzna, że polityka to draństwo

Cybernetyczna wojna

Kiedy w 2006 roku ujawniono opisane wyżej działania, Unia Europejska zażądała od Stanów Zjednoczonych poszanowania prywatności i prawa europejskiego. Porozumienie zawarto w 2010. Jednak Snowden ujawnił, że Stany Zjednoczone nigdy nie zaprzestał monitorowania przelewów SWIFT.

W dniu 29 grudnia 2013 r., Der Spiegel informował, że NSA złamała niemal wszystkie procedury zabezpieczeń największych firm, w tym Cisco, Huawei, Juniper i Dell. Włamano się m.in. do sieci komputerowej, brazylijskiego koncernu naftowego Petrobras.

Jak twierdzi Der Spiegel[1] „*Rząd USA szkoli ludzi nie tylko do paraliżowania sieci wroga, ale także do przejmowania kontroli nad kluczowymi elementami infrastruktury, takimi jak elektrownie, fabryki, lotniska czy oczyszczalnie. Z wojskowego punktu widzenia prowadzona przez NSA inwigilacja jest tylko rozpoznaniem*".

Teza ta została postawiona na podstawie dokumentów ujawnionych przez Edwarda Snowdena.

„Z przeanalizowanych przez Der Spiegel dokumentów wynika, że w razie potrzeby NSA, zapoznawszy się uprzednio z systemami, z jakich korzystają wrogie kraje, może przejść do kolejnej fazy prowadzenia działań ofensywnych w cyberprzestrzeni. Według tajnych danych Stany Zjednoczone przygotowują się do wojen przyszłości, w których będą miały możliwość przez Internet wyłączać komputery, a za nimi - potencjalnie całą infrastrukturę, w tym system energii elektrycznej, wodny, fabryki, lotniska i przepływy pieniężne potencjalnego wroga".
Z NSA współpracuje (lub współpracowała) „banda zwykłych hackerów", której dewiza brzmiała: „twoje dane to nasze dane, a twój sprzęt to nasz sprzęt".

Dyskusyjna jest postawiona przez niemieckich dziennikarz teza, że chodzi o jakieś bliżej nieokreślone „wojny przyszłości". Równie dobrze możemy przyjąć, że już trwa nie wypowiedziana „trzecia wojna światowa"[2]

Trwa III Wojna Światowa - toczy się w internecie

Wszyscy jesteśmy atakowani i wszyscy musimy podjąć wysiłek obrony

Na przykład: **stosuj bezpieczne hasła i nie otwieraj załączników do poczty od nieznanych osób**. Hasła muszą być długie i nie podlegać atakom słownikowym. Na przykład złożenie dwóch krótkich słów to słabe hasło.

Moje hasło – to nie tylko mój problem? Pokazuje to scenariusz ataku na MSZ'2013:

1. Złamanie hasła do poczty jednego z pracowników (zaczęło się od słabego hasła)
2. Wysłanie do jego kolegów załącznika – arkusza ze złośliwym oprogramowaniem (**nie był od obcej osoby – więc otworzono**)
3. Uzyskanie dostępu do sieci

Nawiasem mówiąc równolegle próby takiego samego ataku na MON i KPRP nie powiodły się.

Czytamy o tym we wnioskach po tym ataku opisane na stronie

:https://niebezpiecznik.pl/post/kancelaria-premiera-msz-mon-i-kancelaria-prezydenta-zhackowan

---

[1] *Cytat za* giznet.pl http://giznet.pl/der-spiegel-inwigilacja-nsa-to-tylko-rozpoznanie/
Zobacz tekst w angielskim wydaniu Der Spiegel.
http://www.spiegel.de/international/world/new-snowden-docs-indicate-scope-of-nsa-preparations-for-cyber-battle-a-1013409.html

[2] Tak twierdzi na przykład Papież Franciszek. Zobacz:
https://www.deon.pl/religia/serwis-papieski/aktualnosci-papieskie/art,3917,franciszek-trwa-trzecia-wojna-swiatowa.html

e-wiemy-kto-stoi-za-tymi-wlamaniami/

Nigdy nie jesteśmy w stanie w 100% obronić się przed atakami, dlatego nie powinniśmy zakładać, że nigdy nas nie "zhackują" ale przygotować się zawczasu na taką ewentualność. Z tego powodu, w drugiej linii obrony funkcjonować powinien sprawny monitoring "anomalii", który jak widać w przypadku MON-u i KPRM popłacił — szybkość działania MON-u i KPRP docenił nawet sam Alladyn2 [włamywacz].

Szkoła – szczególna odpowiedzialność ochrony dzieci.

W ramach programu "Bezpieczne dziecko" przeprowadzono ankietę, z której płyną następujące wnioski:
Wprawdzie gros respondentów badania „Bezpieczne dziecko 2018" jest świadomych nowych niebezpieczeństw, jakie czyhają na ich pociechy, to jednak okazuje się, że aktywność najmłodszych niejednokrotnie wymyka się kontroli rodzicielskiej. Przykładem zagrożenia, z którym rodzice nie mieli do czynienia jeszcze kilkanaście lat temu, jest Internet i rozwój nowoczesnych technologii. Prawie 80% przedszkolaków posiada własny telefon komórkowy lub tablet. Ten odsetek wśród nastolatków jest jeszcze wyższy, sięga aż 97% (93% w 2016).
[http://www.egospodarka.pl/151083,Bezpieczne-dziecko-2018,1,39,1.html]

Nie możemy zabronić dzieciom dostępu do internetu. Zamiast tego pozostaje nam je edukować.
Pomocne strony:

- https://sieciaki.pl/
- https://www.edukacja.fdds.pl/scenariusze-zajec
- http://www.necio.pl/
- https://www.saferinternet.pl/
- https://cybernauci.edu.pl/

Nauczycielom potrzebna wiedza

- Jakie są zagrożenia?
- Jakie zachowania są bezpieczne?
- Jak się przed nimi bronić?
- Jak reagować w razie stwierdzenia problemów?

Pierwsze trzy punkty wymagają kompetencji technicznych: wiedzy o funkcjach programów związanych z bezpieczeństwem oraz środkach techniczne jakie mamy do dyspozycji.

II Kompetencje techniczne

Szyfrowanie danych

Można założyć, że wszystkie informacje jakie są przesyłane w sieci mogą zostać podsłuchane. Dlatego ważne informacje muszą być szyfrowane. W szczególności dotyczy to danych służących do identyfikacji i autoryzacji. Jeśli w procesie logowania się do konta posługujemy się nie zaszyfrowanym hasłem, to może ono zostać przechwycone i posłużyć do podszywania się pod

właściciela konta.
Szyfrowanie polega na przekształceniu funkcyjnym wiadomości przy pomocy klucza. Rozważmy na przykład szyfr polegający na dodaniu do każdej cyfry klucza modulo 10. Dla klucza równego 2 otrzymamy: szyfr(1289) = 3401
Odszyfrowanie wiadomości polega na wykonaniu operacji odwrotnej. W tym wypadku odejmowanie klucza modulo 10.
Użycie tego samego klucza przy kodowaniu i odkodowaniu nazywamy **szyfrowaniem symetrycznym**. Algorytmy takiego szyfrowania konstruuje się w taki sposób, aby utrudnić lub uniemożliwić złamanie szyfru. Czyli należy wziąć pod uwagę wszystkie znane metody łamania szyfrów. Kryptografom w łamaniu szyfrów bardzo pomaga możliwość pozyskania (odgadnięcia) tekstu, którego zaszyfrowaną wersję posiadamy. Załóżmy, że budowaliśmy bardzo złożoną funkcję przekształcającą każdą literę na kod. Złamanie szyfru wcale nie wymaga odgadnięcia tej funkcji - wystarczy zestawienie wszystkich liter i odpowiadających im kodów. Dlatego takie kodowanie nie może odbywać się znak po znaku. Stosujemy dłuższe bloki informacji. Takie, by zbudowanie słownika: (tekst , kod) było niemożliwe - nawet przy dostępie do dużej ilości zaszyfrowanych tekstów. Tak powstała idea szyfrów blokowych (https://pl.wikipedia.org/wiki/Szyfr_blokowy).
Ważną cechą kodu jest to, że wyniki kodowania wyglądają na losowy ciąg cyfr (każdy ciąg występuje z podobnym prawdopodobieństwem). Taki kod z dostatecznie dużym kluczem jest praktycznie nie do złamania.

Szyfrowanie asymetryczne

Szyfrowanie symetryczne nie jest dobre, gdy chcemy chronić wiadomość przed fałszowaniem. Fałszerz po odkodowaniu wiadomości może ją zmienić i ponownie zaszyfrować.

Szyfrowanie asymetryczne polega na tym, że odbiorca dysponuje jedynie kluczem do odkodowania, a klucz kodujący jest w posiadaniu nadawcy. Przykład prostego kodowania asymetrycznego to mnożenie. Klucz kodujący (oznaczany **e** - od "encryption") i dekodujący (oznaczany **d** - od decryption) po pomnożeniu przez siebie dają 1. Przykład dla e=4 i d=0.25. Dla wiadomości 12 mamy:

e*12 = 48; 48*d = 12

W tym wypadku uzyskanie klucza kodującego na podstawie klucza dekodującego jest proste - wystarczy policzyć odwrotność. Istnieją jednak metody takiej konstrukcji pary kluczy, że algorytm zgadywania klucza kodującego jest zbyt złożony, by w praktyce był skuteczny. Jedna z takich metod (RSA) zostanie zaprezentowana w dalszej części tego rozdziału.

Klucz deszyfrujący musi być przekazany od nadawcy do odbiorcy. Jak zrobić to w sposób bezpieczny? Można rozróżnić trzy sposoby:

1. Klucze nie są przekazywane ale konstruowane (wyliczane) u nadawcy i odbiorcy w im tylko znany sposób.
2. Do przekazania klucza służą wcześniej zabezpieczone kanały transmisji (zakodowane inną metodą).
3. Skoro nie można czegoś chronić w 100% - uznajemy, że jest to informacja publicznie dostępna.

Jeśli chcemy jedynie chronić wiadomość przed fałszowaniem - asymetryczny klucz dekodujący może być jawny. Nazywa się to metodą **klucza publicznego**.

Szyfry asymetryczne i certyfikaty

Jak podano na wstępie - w szyfrach asymetrycznych używa się pary kluczy oznaczanych literami **e** oraz **d**. Funkcja szyfrująca musi być tak skonstruowana, że znajomość klucza deszyfrującego umożliwia tylko odszyfrowanie (odkodowanie) wiadomości, ale nie umożliwia odgadnięcia klucza szyfrującego i ponownego zaszyfrowania (zakodowania) sfałszowanej wiadomości.

Jest to ważne szczególnie wówczas, gdy zależy nam bardziej na gwarancji autentyczności (**niezaprzeczalności**) niż tajności przekazu. Klucz publiczny (deszyfrujący) może być jawny, ale sposób jego wyliczenia powinien wykluczyć odgadnięcie klucza szyfrującego (**prywatnego**).

Prawidłowe skonstruowana para kluczy (e, d) musi zapewniać praktyczną awykonalnść wyliczenia **e** na podstawie **d**.

Przykładem poprawnego szyfru asymetrycznego jest szyfr wykładniczy. Polega na szyfrowaniu poprzez potęgowanie modulo n (^ oznacza potęgowanie):

C = M^e **mod** n

Mając daną wiadomość zaszyfrowaną (C) oraz klucz (e,n) nie potrafimy w łatwy sposób uzyskać wiadomości pierwotnej. Odszyfrowanie można wykonać poprzez potęgowanie używając innego wykładnika (klucza):

M=C^d **mod** n

Jest to możliwe jeśli powyższe równania są swoją odwrotnością (czyli pomnożenie ich daje w wyniku 1), albo inaczej mówiąc:

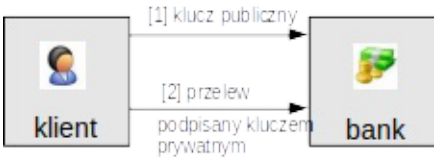
(M^e **mod** n) ^d **mod** n = M

Okazuje się, że znalezienie pary kluczy (e,n) i (d,n) spełniającej ten warunek jest stosunkowo proste (zobacz algorytm RSA).

Transmisja z wykorzystaniem szyfrowania asymetrycznego

Użycie pary kluczy wystarczy do zapewnienia bezpiecznej transmisji. Jeśli wygenerujemy parę kluczy i przekażemy odbiorcy klucz publiczny, to możemy mu wysyłać informacje zaszyfrowane kluczem prywatnym. Mamy przy tym gwarancję, że informacja nie będzie zafałszowana, gdyż jedyne co może zrobić odbiorca, to odszyfrować informację. W ten sposób działał tradycyjny home banking. Klient podpisywał umowę z bankiem i generował parę kluczy. Klucz publiczny dostarczał do banku. Kluczem prywatnym zaś szyfrowana była informacja o przelewach.

Uwaga! Nie chodzi tu o bankowość internetową (w niej stosuje się inne zabezpieczenia), ale o home banking oparty o oprogramowanie instalowane u klienta.



Na przykładzie prostej wymiany informacji z bankiem możemy zauważyć trzy istotne kwestie dotyczące transmisji z kluczem publicznym :

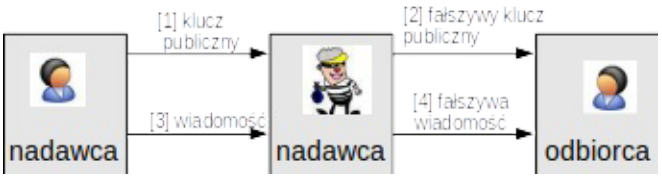
1. Poza szyfrowaniem mamy rozwiązany problem uwierzytelnienia. Dla banku ważniejsza jest identyfikacja klienta i potwierdzenie jego tożsamości (niemożność wyparcia się wykonanej transakcji), niż samo szyfrowanie (bezpieczeństwo jest dodatkowo zapewnione ochroną kanału transmisji). Stąd konieczność obustronnej zgody (umowy) na takie transakcje. Gdyby w powszechnym użyciu były podpisy elektroniczne - taka umowa nie byłaby niezbędna.

2. Samo używanie bezpiecznego szyfru nie zabezpiecza w pełni transakcji. Istnieją bowiem ataki na samą strukturę (system) szyfrowania. Należy zdawać sobie sprawę z tego, że nawet najmniejsza luka z pewnością zostanie wykorzystana. W jednym z polskich banków system informatyczny zapamiętywał jedynie informację o aktualnym kluczu publicznym klienta (bez historii zmian). Zostało to wykorzystane przez pracownika banku, który zamiast wgrać klucz publiczny klienta, wygenerował własną parę kluczy, wgrał fałszywy klucz publiczny do systemu i po ogołoceniu konta wgrał klucz właściwy. Zawsze najsłabszym ogniwem w systemie bezpieczeństwa jest człowiek!

3. Szyfrowanie asymetryczne jest operacją długotrwałą. Dlatego do do szyfrowania wiadomości można stosować prostsze metody, a jedynie jednoznaczny skrót wiadomości zaszyfrować kluczem prywatnym. To daje gwarancję, że wiadomość nie jest zafałszowana. Taki sposób kodowania nazywa się podpisywaniem (elektronicznym).

Certyfikaty

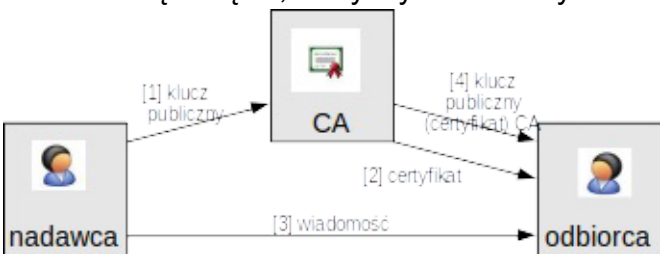
W opisywanym wyżej systemie home bankingu klucze były przekazywane do banku na dyskietce. Obecnie częściej mamy do czynienia z sytuacjami, gdy chcemy dokonać bezpiecznej transmisji, gdy strony nie mają kontaktu bezpośredniego (fizycznego).
Taka transmisja jest narażona na kradzież danych. Ilustruje to poniższy rysunek:



Po wygenerowaniu pary kluczy nadawca przekazuje klucz publiczny [1]. Jednak zostaje on przechwycony przez złodzieja, który generuje swoją parę kluczy i przekazuje nadawcy fałszywy klucz publiczny. Następnie przechwytuje transmisję a dalej przekazuje fałszywą wiadomość zaszyfrowaną fałszywym kluczem. Odbiorca nie ma żadnych szans na stwierdzenie fałszerstwa!

Aby uniknąć takiej sytuacji, wprowadza się zaufaną osobę trzecią, która sprawdza tożsamość nadawcy i informację o nim, zawierającą klucz publiczny podpisuje swoim kluczem prywatnym. Taka wiadomość zawierająca informacje o nadawcy i podmiocie weryfikującym tożsamość nazywa się certyfikatem. Podmiot wystawiający certyfikat (nasza zaufana osoba trzecia) nazywa się urzędem certyfikacji (Certification Authority, CA). Certyfikat jest strukturą danych zawierającą:

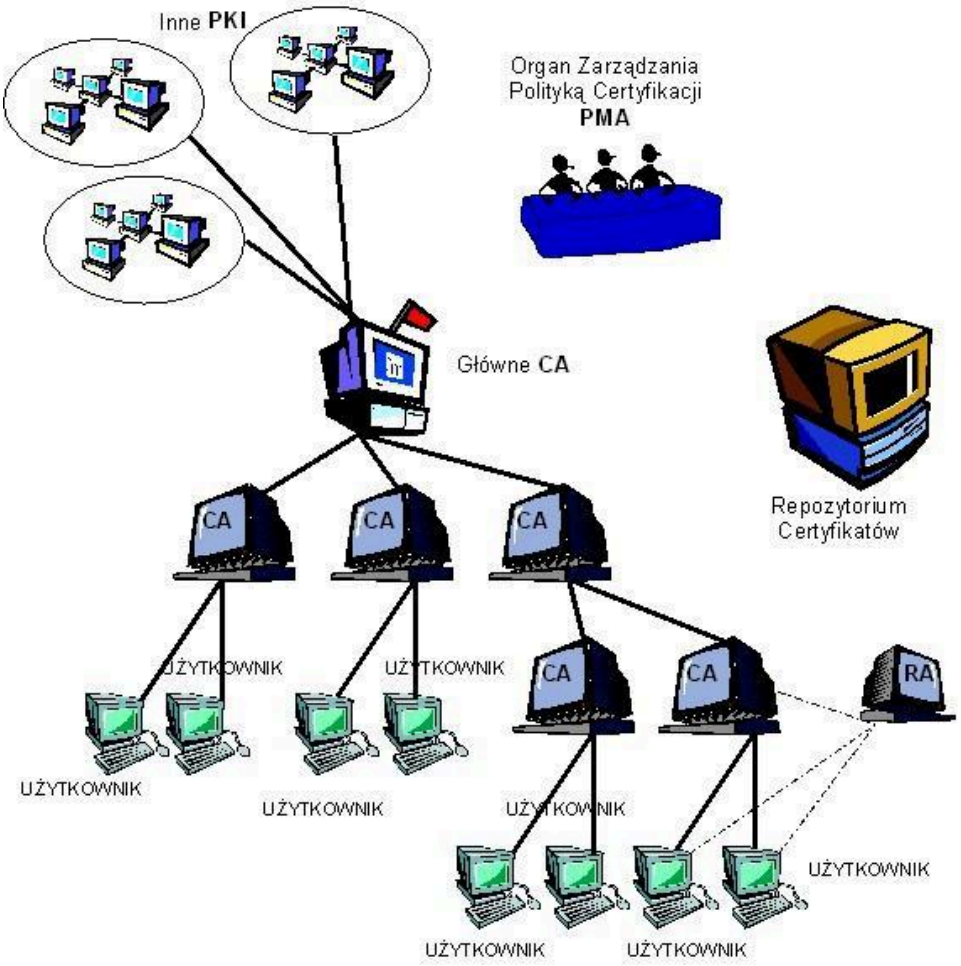
- nazwę podmiotu,
- jego klucz publiczny,
- termin ważności certyfikatu,
- numer seryjny,
- nazwę urzędu, który wystawił certyfikat.



Odbiorca znając klucz publiczny CA może odszyfrować certyfikat i uzyskuje nie tylko klucz publiczny nadawcy, ale również pewność co do jego tożsamości.

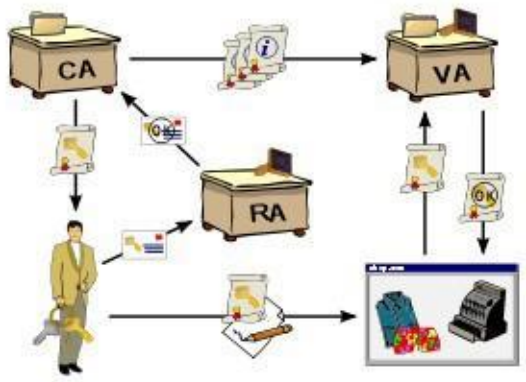
Struktura klucza publicznego

Aby mieć pewność także co do tożsamości centrum certyfikacji, norma X.509 (według której tworzy się podpisy elektroniczne) całą hierarchię certyfikatów, nazywaną Infrastrukturą Klucza Publicznego (PKI - Public Key Infrastructure).



źródło rysunku: http://edu.pjwstk.edu.pl/wyklady/bsi/scb/index51.html

Certyfikat jest wiarygodny, jeśli można zweryfikować wszystkie certyfikaty nadrzędne - aż do głównego CA, który jest nadzorowany przez Organ Zarządzania Polityką Certyfikacji (PMA - Policy Management Authority).
Ponadto można wprowadzić urząd rejestrujący (RA, Registration Authority). Jest to jednostka, któremu CA zleca zdania np. weryfikacji podmiotu.



źródło: http://en.wikipedia.org/wiki/Public_key_infrastructure

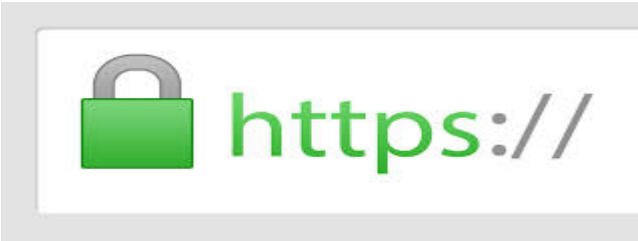
Na rysunku VA oznacza weryfikację (validation).
Strukturę uzupełniają listy unieważnionych certyfikatów CRL (Certificate Revocation List).
Weryfikacja musi być wykonywana z uwzględnieniem aktualnej listy unieważnionych certyfikatów.
Listy te są publikowane przez centrum certyfikacji (CA).

TLS

Protokół SSL (Secure Socket Layer) został opracowany przez firmę Netscape w drugiej połowie lat 90-tych XX wieku. Po ustandardyzowaniu przyjął on nazwę TLS (Transport Layer Security).
Specyfikacja i biblioteki (OpenSSL) dla TLS są ogólnie dostępne.
Obecna wersja 1.3 standardu to znaczące jego uproszczenie i ulepszenie. Zrezygnowano z tych fragmentów standardu, które uznano za niebezpieczne.

SSL/TLS - praktyczne przykłady

HTTPS – bezpieczna odmiana http



Czy moja przeglądarka obsługuje TLS 1.3?

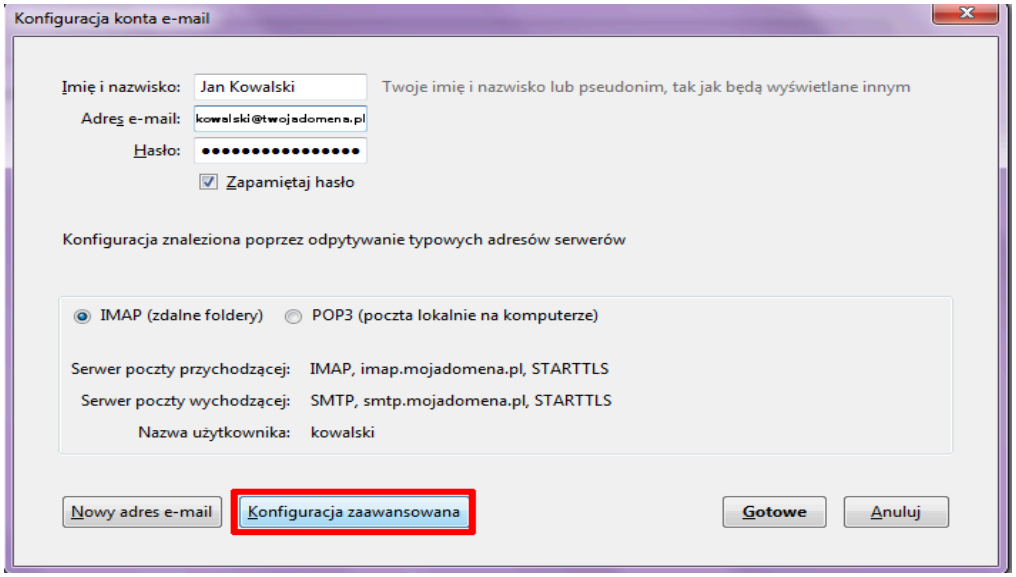
W przypadku Firefox wpisz na pasku adresu "about:config". Odszukaj parametr security.tls.version.max. Powinien mieć wartość 4 (można ustawić).
W przypadku Google Chrome, na pasku adresu wpisujemy: "chrome://flags/". Odszukujemy i ewentualnie włączamy parametr TLS 1.3.

Czy nasz serwer www jest bezpieczny?
Poprawność konfiguracji możemy sprawdzić przy użyciu jednego z serwisów:

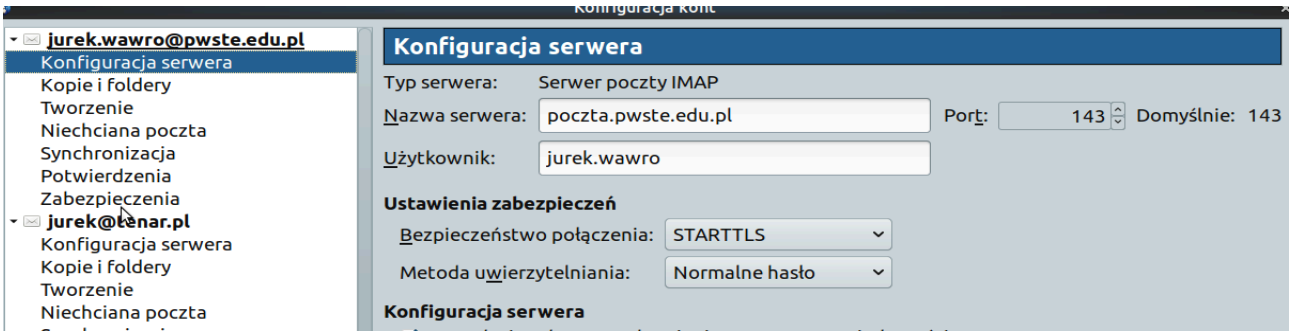
- https://www.ssllabs.com/ssltest/
- https://cservices.certum.pl/muc-customer/ssl/certificate/diagnostics

Poczta

Poczta wychodząca - SMTP - srtartls



Poczta przychodząca IDMAP

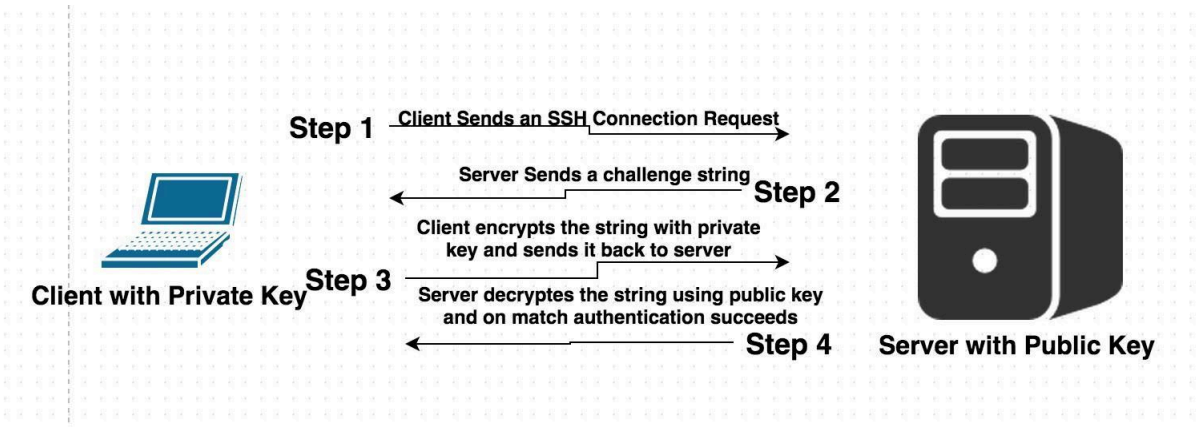


SSH - Dostęp zdalny do konsoli

Serwery (i nie tylko) są zarządzane poprzez komendy (programy) uruchamiane z konsoli. Jednak obecnie rzadko mamy dostęp do komputera do którego jest podłączona klawiatura i monitor. Dlatego niezbędne jest oprogramowanie pozwalające stworzyć konsolę dostępną zdalnie. Służy do tego SSH (skrót od Secure Shell - zob. https://en.wikibooks.org/wiki/OpenSSH/SSH_Protocols, http://itfocus.pl/dzial-it/sieci/ssh-secure- shell-podstawy-bezpiecznej-komunikacji/\).
Pod Windows służą do tego programy Putty i Winscp.
Najczęściej wykorzystujemy SSH w ten sposób, że przy pierwszym logowaniu ściągany jest klucz publiczny z serwera, który następnie zapisuje się w lokalnej bazie kluczy. Użytkownikowi wyświetlany jest skrót - "odcisk palca", który powinien on weryfikować z danymi uzyskanymi wcześniej lub innym kanałem.
Jest to ważne, aby uniknąć ataku z pośrednikiem podszywającym się pod serwer (Man-in-the-middle). Mógłby on nawiązać z klientem łączność swoją parą kluczy, wykorzystując dane podane przez użytkownika do łączności z serwerem. Taki pośrednik może przechwycić i odszyfrować całą transmisję.

Zdalne logowanie z kluczem

Możemy wygenerować parę kluczy i zarejestrować na serwerze klucz publiczny. Wówczas nie będziemy musieli podawać hasła - wystarczy, że posiadamy odpowiednie klucze.
Komunikacja bardzo się upraszcza:



źródło rysunku: https://vmcentral.zendesk.com/hc/en-us/articles/205576449-How-to-Configure-SSH-to-Accept-Only-Key-Based-Authentication

Hasła i funkcje skrótu

W dobrze napisanych systemach hasła nigdy nie są przechowywane. Zamiast nich przechowuje się tak zwany skrót (hash). Jeśli mamy w systemie zapamiętaną wartość hash=skrót(hasło), to w chwili logowania sprawdzamy po prostu, czy hash==skrót(wprowadzone hasło). Przykładowy skrót od słowa “hasło”:
:

```
$1$XpzsBCCNUh6DzchcomXRD0
```

Standardowo gdy skrót zaczyna się od znaku $, to jego początek zawiera dane na temat tego, jak klucz został wygenerowany. Cyfra 1 oznacza MD5. Po drugim znaku $ następuje właściwy skrót.

Dane przetwarzane przez funkcje skrótu mogą być dowolnie długie. Dlatego zastosowanie tych funkcji nie ogranicza się do pamiętania haseł. Można wyliczać skrót (hash) dla plików. Na przykład w systemie Linux poleceniem md5sum. Jeśli wraz z plikiem jest publikowana ta suma, po ściągnięciu z internetu pliku możemy poleceniem md5sum sprawdzić, czy nie był ten plik zmieniany

Jak działa internet?

W przesyłaniu informacji w internecie wykorzystywany jest standard HTTP.

HTTP

Protokół HTTP opisuje wymianę danych między klientem i serwerem poprzez przesyłanie zapytania i odbieranie odpowiedzi. Protokół jest bezstanowy - czyli sposób obróbki zapytania jest zawsze taki sam, a zapytanie musi zawierać wszystkie potrzebne informacje (nie "przechodzą" one z poprzednich zapytań). Typowa para zapytanie / odpowiedź wygląda następująco (http://threats.pl/bezpieczenstwo-aplikacji-internetowych/absolutne-podstawy):

Przykładowe zapytanie:

```
GET /lesson01/index.php HTTP/1.1
Accept: image/gif, image/jpeg, (...)
Accept-Language: en-US,pl;q=0.5
User-Agent: Mozilla/4.0 (compatible; ...)
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
Host: bootcamp.threats.pl
```

Pierwszy parametr zawiera metodę przesyłania danych (https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods\. Drugie pole wskazuje na to jakiego typu danych oczekuje klient (https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Accept\.

Przykładowa odpowiedź:

```
HTTP/1.1 200 OK
Server: IdeaWebServer/v0.60
Date: Mon, 30 Mar 2009 16:16:38 GMT
X-Powered-By: PHP/5.2.6
Set-Cookie: SampleCookie1=CookieValue
Set-Cookie: SampleCookie2=CookieValue; expires=Mon, 30-Mar-2009 17:16:38
GMT
Content-Type: text/html
```

```
Connection: Keep-Alive
Content-Length: 2001
```

Zapytania są wysyłane na adres URL. Adres zawiera wskazanie programu na serwerze, który odpowie na zapytanie oraz dodatkowe parametry - umieszczane po znaku zapytania.

Na przykład:
`https://serwer.com/program?client_id=demo&scope=email`

Algorytm analizy zapytania http:

https://github.com/for-GET/http-decision-diagram/blob/master/doc/README.md

API

Do zaimplementowania transmisji nie wystarczy wybór formatu danych i sposobu ich przesyłania. Potrzebna jest pełna specyfikacja interfejsów łączących programy - czyli tak zwane API (ang. Application Programming Interface).
Protokół HTTP / HTTPS przewiduje on kilka sposobów przesyłania danych. Mogą to być dane zawarte w adresach (URI), formularzach HTML, lub struktury przygotowywane przez komunikujące się programy.
Dane te mogą być przesłane jedną z kilku metod. Standard REST (Representational State Transfer) wykorzystuje te możliwości, przyporządkowując metody do rodzaju operacji jakie mają zostać zrealizowane:

- GET– pobieranie danych,
- POS – dodawanie danych,
- PUT – zmiana (poprzez edycję) danych,
- DELETE – usuwanie danych.

Identyfikacja, autoryzacja, uwierzytelnienie w internecie

Podstawowe pojęcia.

Identyfikacja: ustalenie danych użytkownika (kto to jest.

Autoryzacja / upoważnienie: kto co komu pozwala. Autoryzacja nie ma na celu ustalenia tożsamości podmiotu, ale wyłącznie to, czy ma on uprawnienia (autoryzację) do korzystania z chronionych zasobów. Często uprawnienia przysługują wyłącznie zidentyfikowanym osobom (są przydzielane po ustaleniu tożsamości). Stąd trudność w rozróżnieniu autoryzacji i identyfikacji. Za stosowaniem tego rozróżnienia przemawia przede wszystkim to, że w wielu sytuacjach tożsamość użytkownika systemu nie ma znaczenia. Na przykład na platformie blogerskiej wystarczy informacja, czy użytkownik ma prawo pisać teksty i/lub komentować. Do tego wystarczy znajomość identyfikatora (loginu) i hasła. Nie ma znaczenie kim jest osoba kryjąca się za użytym pseudonimem i czy pisze teksty osobiście.

Uwierzytelnianie - proces identyfikacji – prowadzący do weryfikacji tożsamości w stopniu wystarczającym dla ustalenia uprawnień (autoryzacji) i ewentualnego zapewnienia

niezaprzeczalności.

Proces uwierzytelniania w systemie informatycznym przebiega według następującego schematu:

1. Podmiot się przedstawia.
2. System prosi o potwierdzenie tożsamości.
3. Podmiot potwierdza swoją tożsamość.
4. System weryfikuje poprawność przesłanego potwierdzenia tożsamości.

Przykład (logowanie do systemu bankowego):

1. Użytkownik przedstawia się podając login i hasło.
2. System prosi o potwierdzenie tożsamości - wysyłając kod SMS'em.
3. Podmiot potwierdza swoją tożsamość - wpisując otrzymany kod.
4. System weryfikuje poprawność wpisanego kodu i zakłada, że ma do czynienia z właścicielem urządzenia na które wysłano kd, który na dodatek zna login i hasło. Zakładamy, że tożsamość została potwierdzona.

Mocne (z dużym stopniem ufności) potwierdzenie tożsamości (takie jak w tym przykładzie) polega na tym, że podmiot podlegający uwierzytelnieniu **coś musi wiedzieć** (login i hasło) i **coś posiadać** (urządzenie mobilne odbierające SMS, o numerze zapisanym w systemie). To posiadanie może być **posiadaniem pewnej cechy** - w systemach biometrycznej autoryzacji.

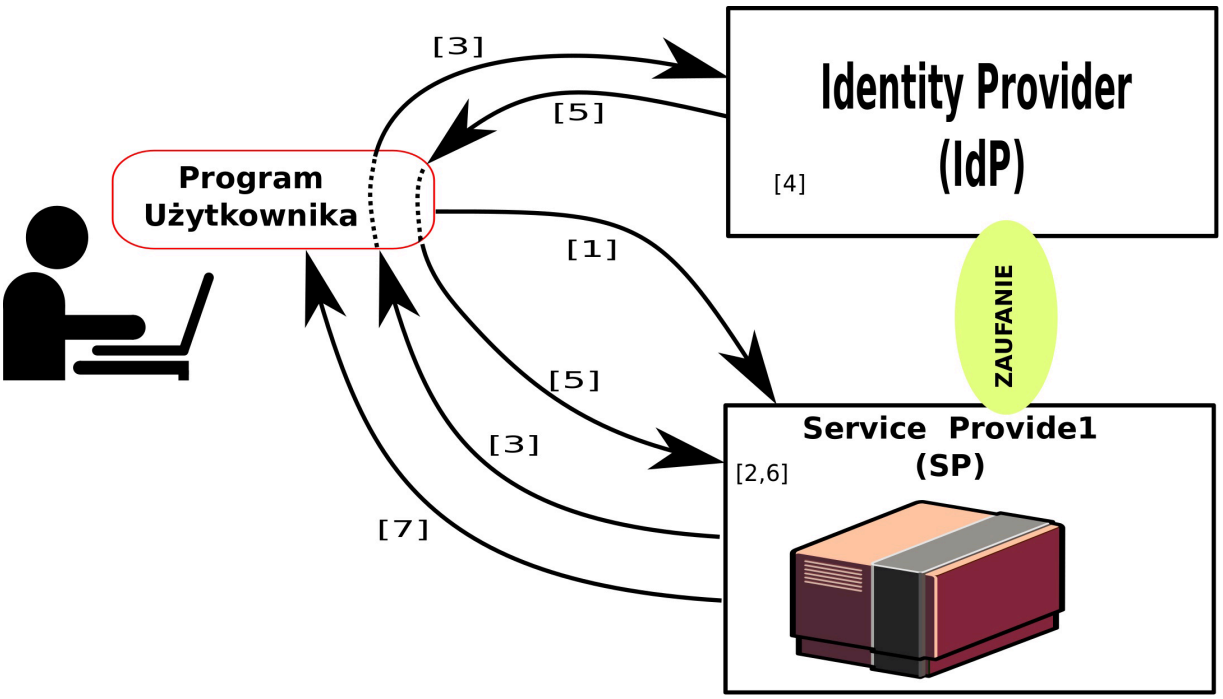
Uwierzytelnianie to proces upewniania się, że identyfikacja jest prawidłowa, a użytkownik jest tym, za kogo się podaje.

Ochrona tożsamości

Często dane związane z tożsamością podmiotu (ujawniane w trakcie identyfikacji) podlegają prawnej ochronie. Wielu dostawców usług nie chce się tym zajmować. Dla nich problemem jest jedynie zapewnienie ochrony zasobów przed nie autoryzowanym dostępem. Wygodnym rozwiązaniem jest wydzielenie ("**delegowanie**") zadania ochrony i udostępniania danych o kliencie do innego systemu, który może być utrzymywany przez zaufany podmiot zewnętrzny lub wewnętrzny (na przykład w obrębie przedsiębiorstwa). Nazywamy go "Identity provider" (IdP).

Rolę Identity Provider (IdP) może pełnić np. ePUAP, Facebook lub Google. Model zarządzania tożsamością z wykorzystaniem **IdP** został opisany jako Federated Identity Management (FIM). Ponieważ model ten obejmuje możliwości weryfikacji praw dostępu do wielu systemów - naturalną jest idea **pojedynczego logowania** (Single Sign On, SSO): te same dane uwierzytelniające (np. nazwa użytkownika i hasło) umożliwiają dostęp do różnych zasobów (systemów, usług, informacji). Najkrócej rzecz ujmując: IdP (podmiot) realizuje FiM (system, schemat działania) dostarczając usługę SSO.

Przykładowy schemat FIM:



1. Użytkownik chce uzyskać dostęp do serwisu.
2. SP (Service Provider) generuje żądanie potwierdzenia tożsamości do IdP.
3. Użytkownik zostaje przekierowany do IdP
4. IdP identyfikuje użytkownika.
5. IdP przesyła poświadczenie tożsamości, które zostaje przekierowane do SP.
6. SP weryfikuje potwierdzenie tożsamości.
7. Użytkownik uzyskuje dostęp do zasobów (serwisu).

Problem ze zrozumieniem działania standardów komunikacyjnych na podstawie takich schematów jak przedstawiony powyżej związany jest z tym, że one nie pokazują celu tej wymiany informacji. W powyższym schemacie celem jest identyfikacja użytkownika.

Schemat ten nie zawiera także informacji na temat tego w jaki sposób weryfikowane jest potwierdzenie tożsamości (krok 6). Jest to schemat idei, której realizację opisują konkretne standardy komunikacyjne (takie jak Oauth2 czy CAS)Ważne jest to, że wymiana informacji odbywa się między systemami (programami), a rola użytkownika sprowadza się do przekazania danych identyfikujących oraz wyrażenia zgody na nawiązanie połączenia.

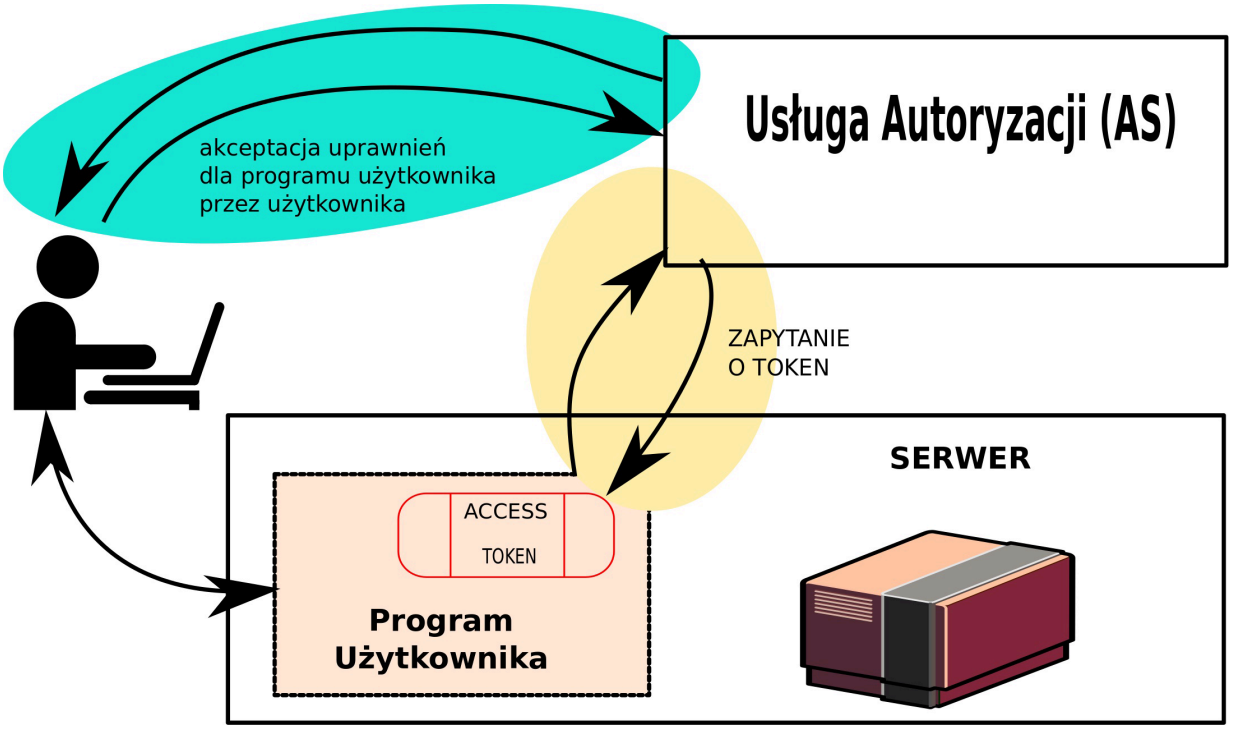
Z punktu widzenia programu (programisty) celem procesu uwierzytelniania w tym modelu jest uzyskanie **tokenu**, którego "posiadanie" upoważnia do realizacji zadania.

Tokeny

Token to informacją, której posiadanie upoważnia do wykonania jakiejś operacji. Autoryzację uzyskuje posiadacz tokenu, który może ale nie musi identyfikować tego posiadacza. Samo posiadanie tokenu pozwala rozróżnić / identyfikować użytkowników systemu. Jednak nie musi się z

tym wiązać dostęp do jakiejkolwiek dodatkowej informacji na temat tożsamości posiadacza tokenu. Sytuacja taka została przewidziana w standardzie OAuth2.

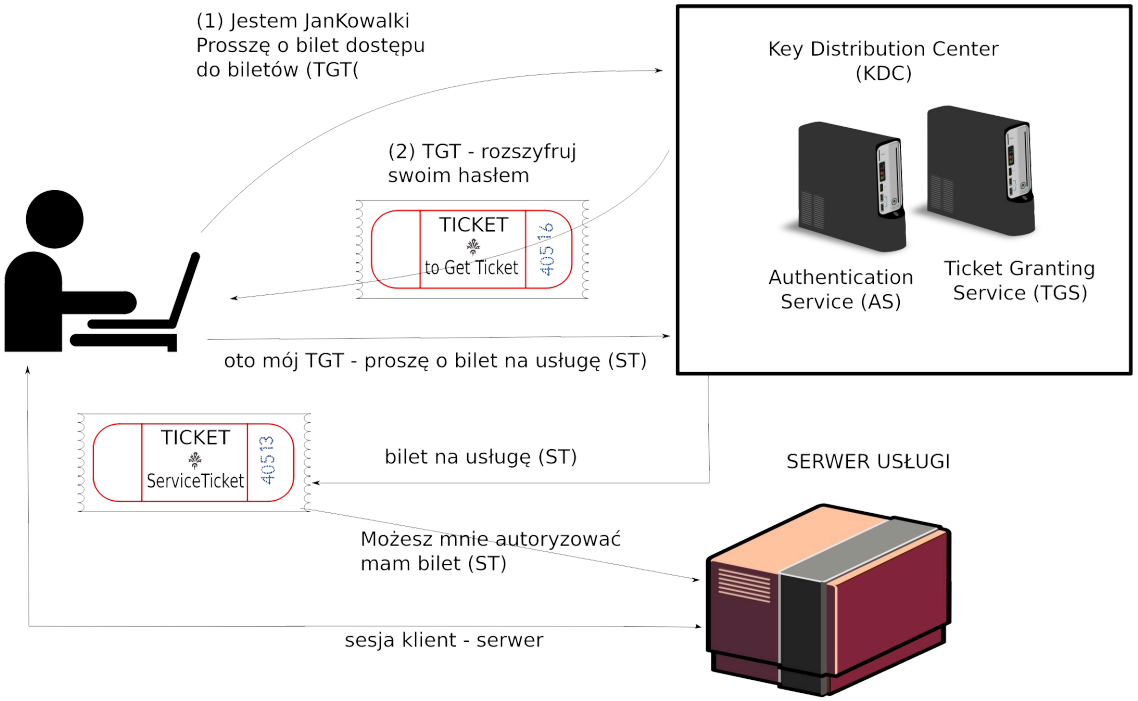
Schemat (bez uwzględnienia SP) wymiany informacji zgodnie z tym standardem przedstawiono na poniższym rysunku (na podstawie: https://medium.com/@darutk/the-simplest-guide-to-oauth-2-0-8c71bd9a15bb\):



Aplikacja internetowa (Program Użytkownika, Client Application) chce uzyskać token autoryzujący dostępu do informacji. Serwer autoryzacyjny zwraca się do użytkownika z zapytaniem o to, czy zezwala na autoryzację aplikacji. Po uzyskaniu potwierdzenia następuje przekazanie tokenu.

W podobny sposób rozwiązano logowanie się do domen **Windows**.

Logowanie do Windows z biletem (“ticket”)



Standardyzacja

Następuje podział na **identyfikację** i **autoryzację** przy uwzględnieniu zaangażowania strony trzeciej (**IdP**), realizującej ideę **SSO** (single sign-on).

W aplikacjach internetowych można rozróżnić dwie strategie ochrony:

1. Podstawą jest autoryzacja (prawa dostępu). Identyfikacja (ustalenie tożsamości) jest dodatkową możliwością.
2. Najważniejsze jest zidentyfikowanie użytkownika (ustalenie tożsamości) na tej podstawie ustalamy prawa dostępu.

W pierwszej opcji dominującym standardem jest **OAuth2** (autoryzacja), który może być uzupełniony przez **OpenId Connect** (identyfikacja).

W opcji drugiej często stosowany jest standard **SAML.** Spotyka się też inne rozwiązanie (na przykład CAS - używany w systemach korporacyjnych i na wyższych uczelniach).

W tym rozdziale znajdziesz opis standardów tych, wykonany z myślą o ich zastosowaniu w programach. Wspomniano też o innych spotykanych w praktyce standardach, jednak w ich przypadku nie zajmujemy się szczegółowym opisem implementacji.

Bardziej szczegółowe informacje znajdziesz w publikacji “Identyfikacja i autoryzacja”. (dostępna wkrótce na stronie wydawnictwa https://verbu.pl/).

III Najważniejszy jest zdrowy rozsądek

1) Stosuj bezpieczne hasła

Po wejściu w życie RODO sprawa wymagań wobec haseł stała się powszechnie znaną. Wydaje się nawet, że została przekroczona jakaś granica zbytniej komplikacji. Jeśli nie musisz stosować RODO, dobrym punktem odniesienia jest Publikacja amerykańskiej agencji NIST (https://pages.nist.gov/800-63-3/sp800-63-3.html) SP 800-63-3. Nosi ona tytuł "Digital Authentication Guideline" i definiuje wymagania, konieczne do uzyskania czterech poziomów pewności (LOA), że podmiot jest tym, za kogo się podaje.

Dokument SP 600-63-3B , nosi tytuł "Uwierzytelnianie i zarządzanie cyklem życia" i koncentruje się na prowadzeniu haseł.

Kilka prostych skazań:

1. Nie stosuj haseł w postaci jednego słowa, albo słabych modyfikacji typu "Passw0rd!"
2. Stosuj znaki specjalne (&%^ ….).
3. Nie zapisuj haseł i nie stosuj przewidywalnej metody modyfikacji (<starehasło> 1, <starehasło> 2 itd.).

Dobrą regułą jest zapamiętanie długiego zdania (może być przysłowie) i zbudowania hasła z pierwszych dwóch liter każdego słowa.

Przykład: Ala ma kota Mruczka AlmakoMr2018

Hasła typu 1234 to zaproszenie do włamania.

Niestety ludzie nie zachowują się rozsądnie. Na przykład testy klientów Swedbanku wykryły, że hasło “123456” miało ponad 5 500 kont klientów banku… [

http://www.egospodarka.pl/151639,5-zasad-cyberbezpieczenstwa-ktore-powinien-znac-kazdy-pracownik,1,12,1.html]

2) Zachowuj ostrożność przy otwieraniu i czytaniu maili

Bank nigdy nie poprosi nas o PIN. Nigdy nie klikajmy w linki czy załączniki od nieznanych nadawców. Każda nieprawidłowość jest podejrzana: błędy językowe, odwołanie do nie istniejących faktur etc…

3) Szykuj się na najgorsze

Pomimo tego, że zachowujesz ostrożność - zawsze licz się z atakiem i utratą danych. Musisz mieć plan awaryjny.

4) Kopia, kopia kopia

Pamiętaj, że pendrive nie jest odpowiednim nośnikiem dla ważnych danych. Nie tylko dlatego, że łatwo go zgubić, ale też ma ograniczoną trwałość. Kopie okresowe powinno się robić na serwerze, lub dysk zewnętrzny. Można też robić kopie w chmurze - ale pamiętaj, że to nie daje gwarancji 100% prywatności.

5) TLS

Transmisja TLS, algorytm klucza publicznego RSA i Certyfikaty dają nam 100% bezpieczeństwa transmisji. Jednak pamiętajmy, że zawsze na początku lub końcu kanału transmisji czyha niebezpieczeństwo.

Strony internetowe: zwróćmy uwagę na **https** – bezpieczna odmiana http

Poczta: protokół TLS lub Srtartls

6) Zachowaj czujność

Reagować na przypadki nietypowej pracy urządzeń i zgłaszać incydenty naruszenia bezpieczeństwa (spam, wirusy, konie trojańskie)

7) Nie dajmy się zwariować.

Istnieją urządzenia “fałszywej bazy” pozwalające przechwycić transmisję z/do naszego smartfona. Jednak wyciąganie baterii, to już przesada. Czasem fałszywa panika jest tworzona celowo. Na przykład NSA nie potrafiąc złamać algorytmu RSA rozsiewało plotki o jego zagrożeniach. Udało im się sprzedać alternatywne rozwiązania - oczywiście z tajnym kluczem dodatkowym.