Proxy Kerberos

Curs

Universitatea “Transilvania” din Brasov

Lect. Costel ALDEA

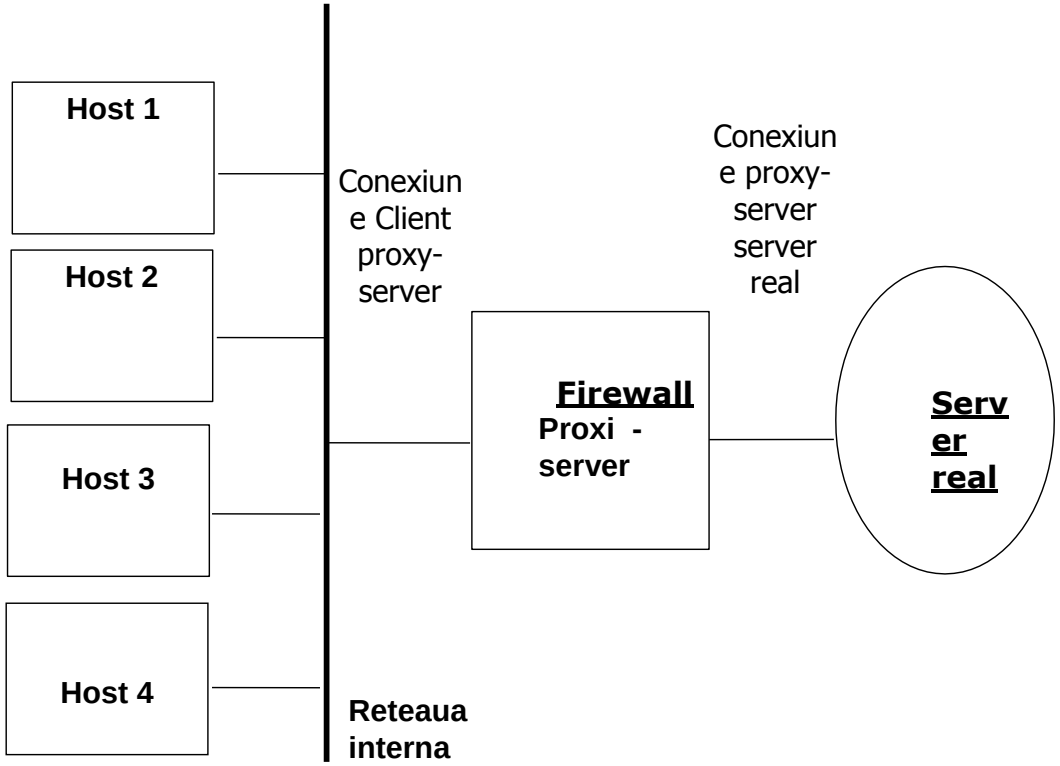
Servicii proxy

- • Un proxy-server este un program care rulează pe o maşină, componentă a firewall-ului numita **gazdă bastion,** datorită rolului de "fortăreaţă" pe care îl îndeplineşte în apărarea reţelei(figura urm).
- Ca urmare a acestui rol, trebuie luate o serie de măsuri pentru asigurarea securităţii sale:
- • • Pe gazda bastion nu vor exista, pe cât posibil, conturi de utilizatori
- - deoarece acestea expun gazda unor atacuri de tip "password guessing"; de asemenea, utilizatorii pot instala şi rula programe proprii care pot constitui ameninţări pentru securitate. Este de preferat să existe un singur cont (al super-user-ului).
- • • Pe gazda bastion nu trebuie să se găsească instrumente (de exemplu, compilatoare) care ar putea ajuta un eventual atacator; în general, pe gazda bastion trebuie să existe doar acele programe (proxy-servere), strict necesare pentru îndeplinirea rolului său.

Servicii proxy

- • • Sistemul de operare trebuie să fie cât mai simplu, având activate doar opţiunile strict necesare.
- • • Trebuie să existe proceduri de backup periodic, inclusiv al fişierelor de log. În cazul unui atac, fişierele de log pot furniza informaţii cu privire la natura acestuia.

Servicii proxy



Arhitectură cu gazdă dual

Servicii proxy

- De obicei, gazda bastion este o poartă (gateway). Din aceasta cauză, un proxy server mai este denumit şi poartă la nivel aplicaţie (application gateway). De obicei, o sesiune de comunicaţie în Internet se desfăşoară după modelul client-server: clientul emite o cerere către un server (de exemplu, transferul unei pagini Web) iar acesta răspunde cererii. În cazul serviciilor proxy, comunicaţia se desfasoară astfel:
- 1. Clientul emite cererea către proxy server (nu către serverul real), comunicându-i acestuia destinaţia dorită (serverul real care va răspunde cererii);
- 2. Proxy-serverul analizează cererea şi, dacă aceasta respectă politica de securitate a organizaţiei, este trimisă serverului real.
- 3. Serverul răspunde cererii, răspuns care se întoarce la proxy-server.
- 4. Proxy-serverul analizează răspunsul şi dacă aceasta respectă politica de securitate a organizaţiei, este trimis clientului.

Servicii proxy

- Ca urmare, nu mai este creată o singură conexiune client-/server, ci două: client/proxy-server şi proxy-server/server-real. Serverul nu este conştient de existenţa clientului; el comunică doar cu proxy-serverul. Clientul nu comunică direct cu serverul, ci doar cu proxy-serverul. Pentru ca aceasta schemă să funcţioneze, trebuie ca între client şi server să nu existe conectivitate directă (la nivel IP); în caz contrar, clientul poate trimite cererea direct către serverul real, eludând astfel proxy-serverul. Din această cauză, opţiunea de routare pe gateway (transfer automat al pachetelor dintr-o reţea în cealaltă) va fi dezactivată.
  - Fiecare proxy-server va avea un set de reguli (analoge celor de la ruterele protectoare), care implementează politica de securitate a organizaţiei relativ la protocolul respectiv.

Servicii proxy

- • Avantajul proxy-serverelor, este că ele înţeleg protocolul pentru care au fost proiectate, făcând astfel posibil controlul la nivel aplicaţie, care constă din:
- • • **Autentificare** - aceasta permite selectarea utilizatorilor (interni sau externi) care au dreptul să acceseze un anumit serviciu.
- • • **Filtrarea individuală** a operaţiilor protocolului - de exemplu, cu ajutorul unui proxy-server pentru FTP poate fi implementatăo politică de securitate care permite aducerea dar interzice exportarea de fişiere;
- • • **Monitorizare**a - înregistrarea în fişierele de log a sesiunilor de reţea.
- • Principalul dezavantaj al proxy-serverelor îl constituie faptul că, prin existenţa a două conexiuni în loc de una, se modifică procedura de utilizare a serviciilor Internet. Acest lucru impune modificarea fie a programelor client, fie a modalităţii de apel al acestora de catre utilizatori.

Servicii proxy

- • Rezultă astfel două abordări posibile:
- • • Utilizarea de clienţi normali şi modificarea comportamentului utilizatorilor: aceştia nu se mai pot conecta direct la serverele dorite, ci trebuie să se conecteze mai intâi la firewall(proxy-server) si să-i "spună" acestuia destinaţia dorită, într-un limbaj specific fiecărui proxy-server. Această metodă impune utilizatorilor un mod de lucru mai greoi. În particular, utilizatorii trebuie să folosească proceduri diferite pentru accesarea serverelor din reţeaua internă a organizaţiei (pentru aceasta nu trebuie sa treacă prin firewall), faţă de cele externe (din Internet).
- • • Utilizarea de clienţi modificaţi sau configuraţi corespunzător pentru "conlucrarea" cu proxy-serverul. În acest caz, utilizatorul nu va sesiza prezenţa proxy-serverului (exceptând cazul când este necesară autentificarea de către firewall): clientul va efectua automat toate procedurile privind conectarea prin firewall. Astfel de clienţi există în prezent pentru o serie de protocoale, ca FTP sau HTTP. De exemplu, Netscape Navigator sau Internet Explorer pot fi configurate să se conecteze prin intermediul unui proxy-server.

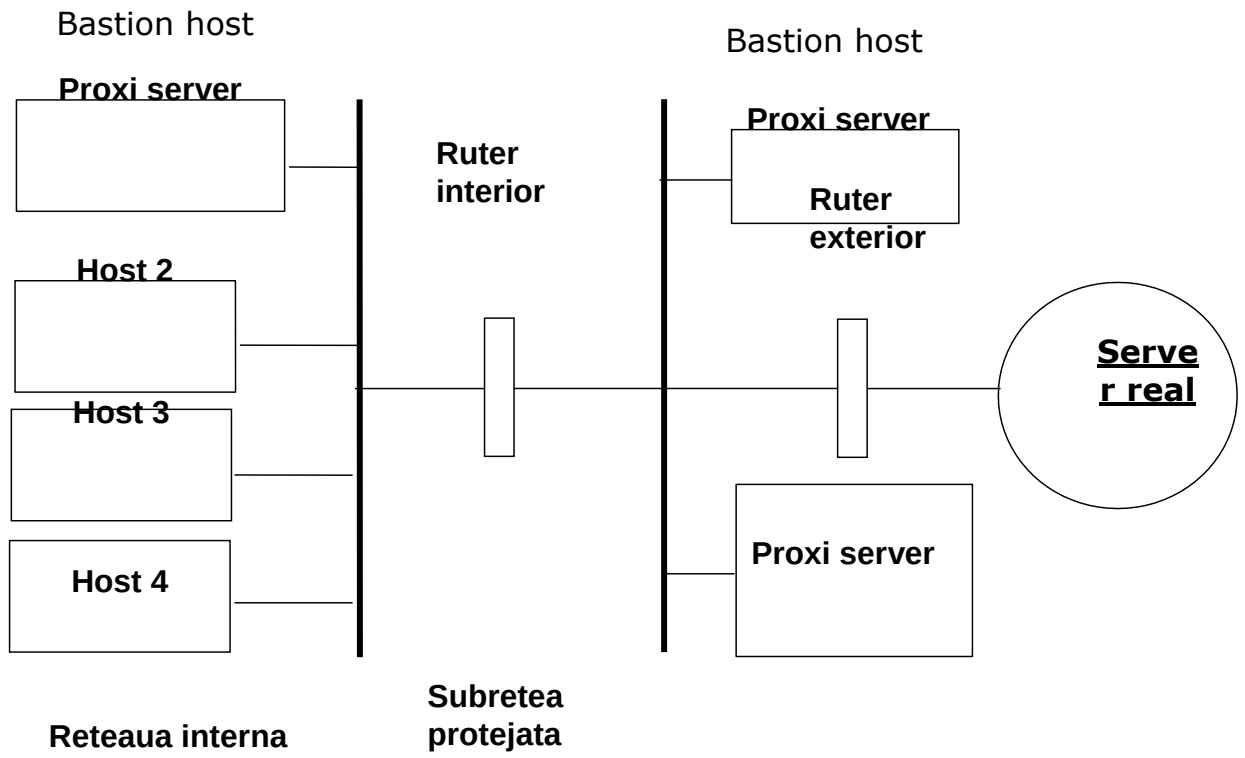
Arhitectura cu subreţea protejată (screened subnet)

- În această arhitectură, bastionul nu se mai află în reţeaua internă, ci este despărţit de aceasta printr-un al doilea ruter protector, numit router interior. Cele două routere delimitează o reţea intermediară, care separă reţeaua internă de cea externă (Internet). Pe această reţea (numită şi DMZ - DeMilitarized Zone) se află gazda (sau gazdele) bastion (vezi figura urm).
- • Astfel, dacă un atacator reuşeşte să compromită fie routerul exterior, fie un gazda bastion, el nu are acces imediat 1a reţeaua internă. Datorită routerului interior, prin reţeaua intermediară nu circulă informaţie internă, deci interceptarea traficului în această reţea nu va divulga informaţie
- confidenţială. Routerul exterior separă reţeaua intermediară de cea externă. Permite traficul doar între reţeaua intermediară (bastion host-uri) şi reţeaua externă, traficul direct între reţeaua internă şi cea externă este interzis. Principalul său rol este de a proteja reţeaua intermediară (bastion host-urile) împotriva atacurilor provenite din reţeaua externă.

Arhitectura cu subreţea protejată (screened subnet)

- • Ruterul interior separă reţeaua internă de reţeaua intermediară, permiţând traficul doar între cele două; traficul direct între reţeaua internă şi cea externă este interzis. Se observă că traficul direct intre reţeaua internă şi cea externă este interzis atât de routerul interior, cât şi de cel exterior.
- • Această arhitectură oferă cel mai înalt grad de protecţie, însă are costul cel mai mare, datorită prezenţei a două routere. În reţeaua intermediară pot exista mai multe gazde bastion (fiecare rulând proxy-servere diferite), dar această configuraţie este foarte costisitoare.
-

Arhitectura cu subreţea protejată (screened subnet)



Arhitectura cu subreţea protejată

Kerberos

- • Autentificarea (authentication) este procesul prin care se verifică identitatea unei instanţe (de exemplu un utilizator) care a produs nişte date.
- • Să observăm că pentru a putea vorbi despre autentificare trebuie să avem o noţiune de *identitate* a utilizatorilor. Calculatorul trebuie să poată distinge diferiţii indivizi care îl folosesc. Trebuie să existe un *spaţiu de nume* pentru utilizatori; autentificarea va pune în corespondenţă o entitate activă cu un astfel de nume.
- • În Unix numele utilizatorilor autorizaţi sunt trecute într-o mică bază de date/etc/passwd (sistemele mai moderne au scheme mai complicate, dar înrudite); pentru un sistem Unix a autentifica un utilizator constă în a asigna unul dintre aceste nume cunoscute unui set de procese executate de acel utilizator. Dacă un individ nu are un nume în acel fişier atunci el nu există pentru calculator.

Kerberos

- • Cu alte cuvinte, pentru a putea vorbi despre autentificare trebuie ca părţile implicate în comunicare să aibă un spaţiu de nume comun pentru utilizatori.
- • Protocolul Kerberos a fost proiectat la Universitatea MIT (Massachusetts Institute of Techonology) în cadrul proiectului Athena, în jurul anului 1984. Scopul protocolului Kerberos este de a permite unui client să-şi demonstreze identitatea unui server aflat la distanţă,undeva dincolo de o reţea complet nesigură.
- • Protocolul garantează, de asemenea că clientul nu poate conversa cu un calculator care se dă drept server; autentificarea se face în ambele direcţii.

Kerberos

- Protocolul în sine constă dintr-un schimb de mesaje între un client şi o serie de servere, fiecare cu o altă misiune. Idea de bază aparţine lui Needham şi Schroeder care au publicat-o în 1978 în Comm. of the ACM. Descrierea detaliată a protocolului se găseşte în documentul numit RFC 1510, disponibil de pe Internet.
- • Cei care administrează Kerberos pot pune întrebări pe grupul de
- News **comp.protocols.kerberos.com**
- • Protocolul Kerberos indică de fapt o serie de mesaje care trebuie schimbate între părţile care doresc să comunice; unele din mesaje sunt criptate. Ca funcţie de criptare/decriptare se foloseşte DES (Data Encryption Standard).
- • Kerberos este un protocol; pentru a fi util anumitor aplicaţii (cele care au nevoie de comunicaţie client-server),acestea trebuie modificate pentru a folosi autentificarea oferită de Kerberos. (Aplicaţiile modificate sunt atunci numite ``kerberized'').

Kerberos

- • În mod normal într-un domeniu administrativ în care se foloseşte Kerberos programe ca telnet, rlogin, POP (post-office protocol), fişiere la distanţă (AFS), etc. trebuie rescrise în aşa fel încît clientul să se autentifice serverului folosind noua metodă (toate aceste aplicaţii sunt de tip client-server).

Principii

- Premiza iniţială este că există un server central, numit server de autentificare (AS), care cunoaşte identităţile tuturor clienţilor posibili.
- • Fiecare client a stabilit o parolă secretă pe care şi acest server o ştie.
- Parola ajunge la server printr-o cale sigură; aceasta parola nu este cunoscută de nimeni altcineva, nici măcar de celelalte servere cu care
- clientul va comunica (de la care va obţine serviciile care-l interesează).
- • Toate celelalte servere din domeniul administrativ sunt şi ele clienţi ai AS: au o parolă unică ştiută numai de AS şi de acel server.
- • Niciodată parola nu va circula în clar pe reţea; atunci oricine ar putea să o citească şi să o folosească în locul clientului de drept.
- • Pentru a se proteja împotriva intruşilor care pot înregistra sau şterge mesaje, acestea vor contine informaţii ca **ora la care au fost trimise** şi **un număr de ordine** (pentru a detecta omisiunile).

Principii

- • Cheile folosite în comunicarea dintre client şi servere se schimbă frecvent.
- Implementarea standard expiră o cheie după 25 de ore. În felul acesta un atac criptanalitic nu va avea prea mult succes: dacă durează mai mult de 25 de ore atunci cheia descoperită este deja inutilă (eventualele mesaje deja interceptate şi stocate vor putea fi citite, dar stricăciunile sunt limitate).
- Cheia pe care un client şi un server o folosesc în comun se numeşte *cheie de sesiune* şi este generată aleator.

Mesajele (protocolul Needham-Schroeder)

- Să presupunem că un client vrea să dialogheze cu un server de fisiere.

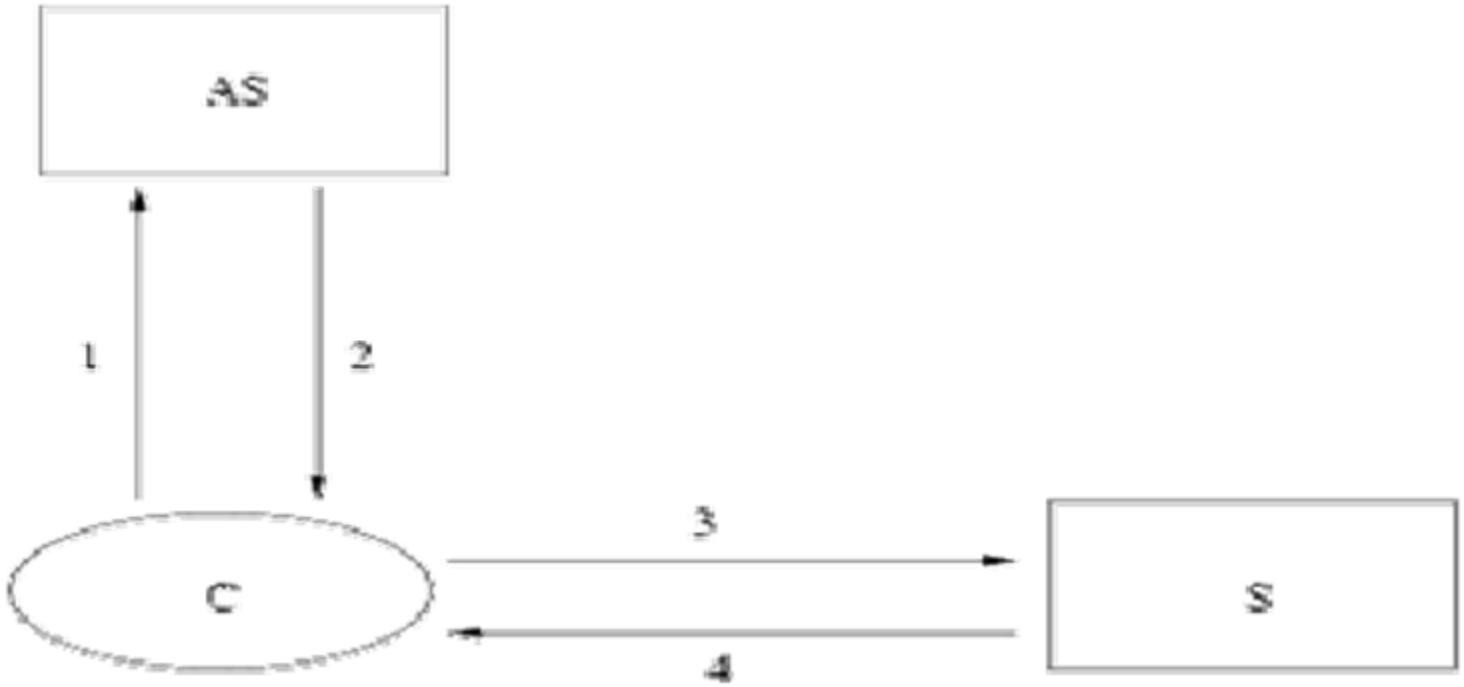
Protocolul esenţial este compus din 4 mesaje (protocolul real este o simplă extensie a ideii prezentate):

- Un mesaj de la client spre AS, prin care se indică intenţia de a comunica cu serverul de fisiere;
- Un mesaj de răspuns de la AS pentru client, prin care AS îi trimite clientului noua cheie de sesiune şi un pachet pentru serverul de fisiere;
- Un mesaj de la client spre serverul de fisiere, în care este inclus pacheţelul de mai sus, pentru a garanta faptul că clientul a discutat cu AS (pacheţelul poate veni numai de la AS);
- Un mesaj de răspuns de la serverul de fisiere prin care clientul este convins că serverul a putut deschide pacheţelul, deci că discută într-adevăr cu acest server.

Mesajele (protocolul Needham-Schroeder)

- • După acest schimb iniţial de mesaje, clientul şi serverul de fisiere vor folosi cheia de sesiune generată de AS pentru a cripta cu DES toată comunicaţia dintre ei.
- • Vom vedea că mesajele sunt relativ încîlcite pentru a preveni toate atacurile de mai sus. Dacă veţi încerca să simplificaţi protocolul aproape sigur îl veţi face vulnerabil la anumite tipuri de atacuri.
- • Să notăm cele 3 entităţi care colaborează astfel: C clientul, AS serverul de autentificare şi S serverul de fisiere(vedeţi şi figura urm.).

Mesajele (protocolul Needham-Schroeder)



Mesajele (protocolul Needham-Schroeder)

- • Vom mai introduce următoarele notaţii:
- - Ka,b este cheia de sesiune pe care o folosesc pentru conversaţie a şi b; de exemplu KC,S va fi cheia folosită pentru criptare/decriptare între client şi serverul de fis.; cum am spus mai sus, AS este serverul de autentif., care ştie parola fiecărei alte entităţi din sistem.
- - Kz este cheia pe care clientul z şi AS o folosesc în comun (parola clientului z); de pildă Ks este parola serv. de fis. (cunoscută numai de el şi de AS);
- - {mesaj1, mesaj2}K indica faptul că mesajele 1 şi 2 sunt puse într-un pachet care este apoi criptat cu cheia K.
- - *tichet* (ticket): Ta,b = { Ka,b, a, ora curenta };
- un mesaj în care sunt împachetate 3 informaţii: o cheie de sesiune
- între a şi b, numele lui a şi ora curentă.

Mesajele (protocolul Needham-Schroeder)

- Cu aceste notaţii putem scrie complet ne-ambiguu toate mesajele schimbate între C, S şi AS. Săgeata indică traseul fiecărui mesaj: C ---> AS: C, S, ora expirare, N (aleator).
- Clientul afirmă propria identitate, identitatea serverului cu care vrea să discute, trimite un număr aleator şi cît timp ar dori să converseze cu S.
- • AS va răspunde astfel: AS ---> C$: {KC,S, S, ora expirare, N}KC,
- {TC,S}KS .
  Acest mesaj complicat are două părţi mari: prima este destinata clientului, şi este criptată cu cheia clientului KC, iar a doua este un tichet pentru S, criptat cu cheia lui S. Să vedem la ce folosesc părţile din mesaj:

Mesajele (protocolul Needham-Schroeder)

- Prima parte a mesajului este criptată cu cheia KC a lui C. Pentru că această cheie este secretă, nimeni altcineva nu poate decripta acest mesaj decît C; la fel stau lucrurile şi pentru partea a doua a mesajului, care fiind criptată cu KS este inteligibilă numai pentru S.
- KC,S este un număr aleator generat de AS, care va fi folosit ca cheie de sesiune între C şi S; observaţi că cheia de sesiune apare în ambele părţi ale mesajului, în aşa fel încît va fi cunoscută atît de către C cît şi de S.
-
- AS îi confirmă lui C identitatea serverului de disc S şi indică durata de validitate a lui KC,S (care poate fi diferită decît a dorit C în primul mesaj).

Mesajele (protocolul Needham-Schroeder)

- • Faptul că N apare în mesajul către C garantează că acest mesaj a fost criptat de AS; de asemenea, acest mesaj nu putea fi un mesaj mai vechi dintr-o comunicaţie anterioară, pentru că N diferă.
- • Partea a doua a mesajului este opacă pentru C; tot ce poate face C cu ea este să o înainteze lui S; C (sau oricine altcineva) nu o poate citi; dacă cineva ar modifica partea a doua, S nu ar mai putea-o decodifica şi obţine un mesaj corect, deci tichetul nu poate fi contrafăcut sau modificat.
- • C ---> S: {C, ora curenta, suma de control}KC,S, {TC,S}KS.
- • Mesajul are din nou două părţi.
- -Partea a doua este exact partea a doua a mesajului 2, aşa cum a fost primită de la AS.

Mesajele (protocolul Needham-Schroeder)

- - Prima parte a mesajului are scopul de a demonstra că acest mesaj este
- ``proaspăt'': S va extrage KC,S din partea a doua a mesajului şi o va folosi pentru a citi prima parte a mesajului. De acolo află ora la care a fost trimis mesajul, şi îl rejectează dacă ora diferă prea tare de ora locală. (Asta ar putea să însemne că mesajul a fost capturat de un inamic şi re-lansat). Suma de control ne asigură că mesajul nu a fost modificat de nimeni; este
- practic imposibil să modifici un mesaj criptat şi să repari şi suma de control dacă nu ştii cheia.
- • S ---> C: {ora curenta+1}KC,S.
- • Prin acest mesaj S în convinge pe C că a ajuns la destinaţia dorită: mesajul are ora curentă trimisă anterior de C plus 1. Dar ora putea fi extrasă numai de cel care avea KC,S, care fiind o cheie de sesiune era ştiută numai de S. Nu era suficient să returneze aceeaşi valoare, pentru că atunci acest mesaj putea fi o copie a (unei părţi a) mesajului anterior.

Mesajele (protocolul Needham-Schroeder)

- La sfîrşitul acestei comunicaţii atît C cît şi S sunt siguri de identitatea celuilalt şi în plus au la dispoziţie o cheie de sesiune KC,S cu care pot cripta toate mesajele pe care le schimbă. Autentificarea a fost făcută.

Implementarea Kerberos

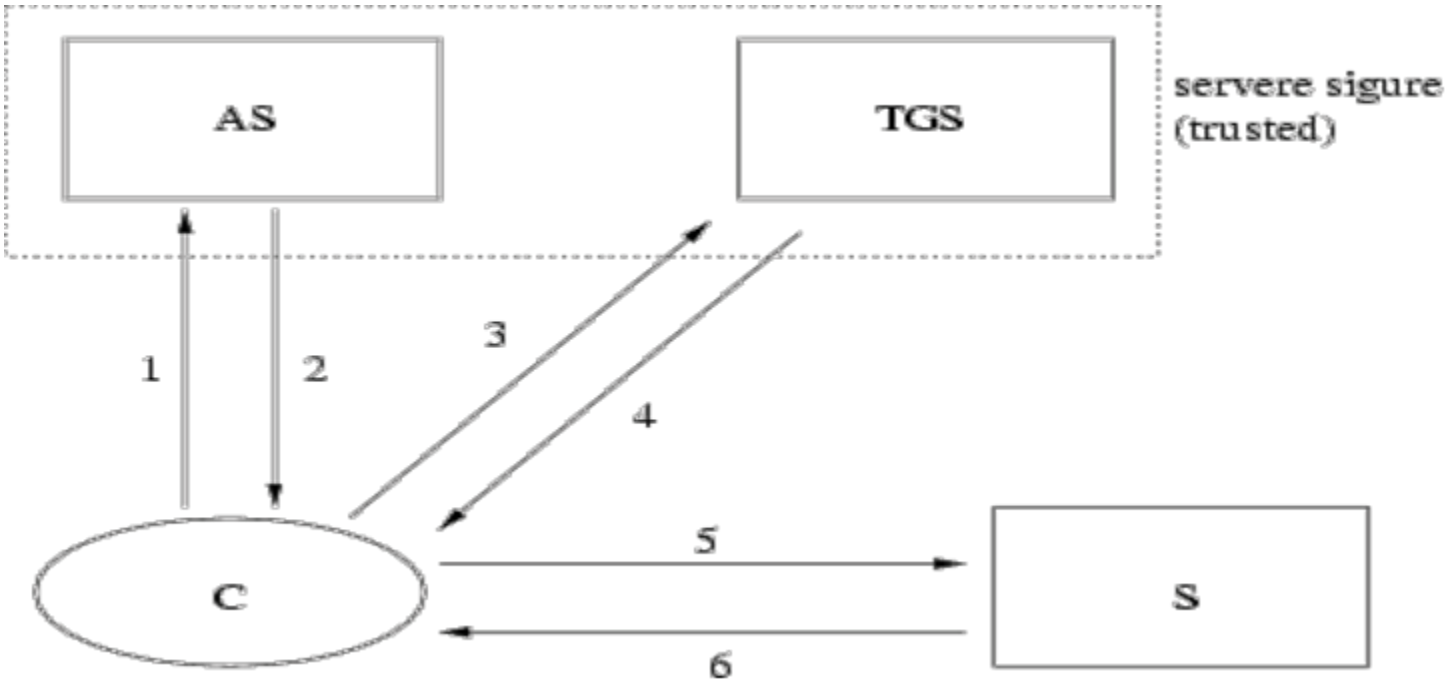
- • Implementarea lui Kerberos încearcă să ţină cont de anumite particularităţi ale lumii reale care fac realizarea protocolului mai dificilă.
- • Prima întrebare: unde este stocată parola fiecărui client KC, KS, etc. AS trebuie să fie o maşină foarte sigură; dacă parola unui client este stocată pe disc, atunci este mai putin protejata. Deci Kerberos nu memorează parola nicăieri! Utilizatorul este obligat să o tasteze de fiecare dată cînd vrea să fie autentificat.
- • Observaţi că KC este folosită în mesajele de mai sus numai pentru a decripta mesajul 2; în rest este inutilă.
- • Deci procedura este următoarea: programul kerberizat al clientului va lua legătura cu AS, iar cînd răspunsul soseşte utilizatorul trebuie să tasteze parola KC. Parola este imediat folosită pentru a decripta mesajul de la AS după care este complet şterasă din memorie. În acest fel fereastra de vulnerabilitate este redusă la maximum.

Implementarea Kerberos

- • Pe de altă parte, utilizatorul va trebui să tasteze parola pentru fiecare nou server S pe care vrea să-l folosească (adică pentru fiecare nouă sesiune). Situatie care mareste efortul. Kerberos a introdus atunci un al doilea server central numit ``Serverul care dă tichete''(TGS- Ticket Granting Server). Ideea este TGS are de fapt toate parolele KS. Clientul C se autentifică la TGS exact în acelaşi fel ca la oricare server S, obţinînd un tichet de la AS.
- Odată autentificat la TGS şi folosind cheia de sesiune KC, TGS clientul poate solicita oricîte chei pentru alte servere S, S1, etc.
- • Figura urm. arată o situaţie reală: mesajele 1,2 sunt schimbate numai cînd utilizatorul se logineaza. Mesajele 3,4 sunt schimbate de fiecare dată cînd utilizatorul vrea să contacteze un nou server (adică să deschidă o nouă sesiune).

Implementarea Kerberos

- Mesajul 5 este folosit de client pentru a se autentifica fiecărui nou server, iar mesajul 6 este opţional, autentificînd serverul pentru client. În mesajul 5 apare un element nou, numit Ksubsesiune: clientul poate alege aici o nouă cheie de sesiune care să fie folosită în locul cheii oferite de TGS, KC,S.



Implementarea Kerberos

- • Schimbul complet de mesaje în protocolul Kerberos, este prezentat in tab. urm.
- • Kerberos este implementat sub forma unor procese server (AS, TGS) şi a unor biblioteci care se pot lega în programele clienţilor şi serverelor. Funcţionează sub o mare varietate de sisteme de operare(Unix,Windows). Mai are si alte aspecte(de autentificarea între domenii administrative diferite, transmiterea tichetelor, etc.)

Implementarea Kerberos

Nr	Între	Conţinutul mesajului
1	C ---> AS	C, TGS, ora de expirare, N
2	AS ---> C	$K_{C,\ TGS}$, ora de expirare, N}K_C, $\{T_{C,TGS}\}K_{TGS}$
3	C ---> TGS	{ora locala}$K_{C,TGS}$, $\{T_{C,TGS}\}K_{TGS}$, S, ora de expirare, N_1
4	TGS ---> C	$K_{C,S}$, S, ora de expirare, $N_1\}K_{C,\ TGS}$, $\{T_{C,S}\}K_S$
5	C ---> S	{ora locala, suma de control,$K_{subsesiune}\}K_{C,S}$, $\{T_{C,S}\}K_S$
6	S ---> C	{ora locala}$K_{C,S}$

Slăbiciunile lui Kerberos

- • Chiar dacă în teorie Kerberos este minunat, implementarea lui în practică este cel puţin dificilă. Condiţiile ideale existente pe hîrtie sunt greu de obţinut într-o reţea de calculatoare reale.
- • La ora actuală nu există nici un fel de metodă complet riguroasă pentru a arăta ca un protocol criptografic nu scapă informaţii; există metode pentru a testa dacă un protocol rezistă la atacurile cunoscute, dar foarte adesea se publică algoritmi care mai tîrziu se dovedesc greşiţi. În general, raţionamentele cu astfel de protocoale sunt foarte complicate. Cercetarea în domeniu este în plină desfăşurare şi foloseşte tehnici ca teoria informaţiei, teoria complexităţii, logici speciale (ex. knowledge theory), etc.

Slăbiciunile lui Kerberos

- • Voi ilustra aici numai unele dintre deficienţe, pentru a da o idee despre natura lor.
- • Să observăm că clientul trebuie să păstreze undeva cheile de sesiune pentru a putea conversa cu serverele: fiecare mesaj după cele de autentificare va fi criptat cu aceste chei. Clientul trebuie să posede deci practic permanent KC, TGS şi KC,S. E adevărat că aceste chei expiră în 25 de ore, deci sunt mai puţin importante decît o parolă care teoretic este folosită luni întregi. Întrebarea este însă: unde sunt ţinute pe calculatorul clientului aceste chei?

Slăbiciunile lui Kerberos

- • Pe o staţie obişnuită Unix lucrează în mod normal mai mulţi utilizatori. Tichetele utiliz. Trebuie sa fie diferite. Pe un sistem Unix nimic nu poate fi ascuns de administrator (root). Administratorul unui sistem poate citi orice fişier, şi poate inspecta memoria fizică a oricărui proces. Acesta este un călcîi al lui Ahile al lui Kerberos;
- toate metodele cunoscute pentru a penetra un sistem Unix ameninţă
- siguranţa întregului protocol. Ori securitatea unui sistem Unix, care este foarte complicat, este extrem de greu de controlat; există o sumedenie de breşe de care un atacator ar putea profita.
- • O altă mare problemă este cu staţiile de lucru fără disc (diskless); aceste staţii de obicei importă discuri prin reţea. Deci de îndată ce o astfel de staţie stochează un tichet pe disc, tichetul va călători prin reţea, care am stabilit că este expusă la tot felul de atacuri!
-

Slăbiciunile lui Kerberos

- Nici păstrarea tichetului în memorie nu este neapărat mai sigură: algoritmii de paginare stochează paginile pe disc (în partiţia de swap) atunci cînd calculatorul nu are destulă memorie, deci am revenit la aceeaşi problemă.
-
  - Exemplu:
- Într-o reţea mare de calculatoare sincronizarea ceasurilor se face automat, folosind un protocol numit NTP Network: ora locală a serverului cu ora din tichet. Pentru un interval de 5 minute serverul memorează toate tichetele primite, pentru a depista duplicate, eventual rezultate dintr-un atac care re- transmite pachete vechi capturate. Un tichet mai vechi de 5 minute este considerat expirat şi ignorat. În felul acesta un server nu va primi niciodată acelaşi pachet de două ori. Asta presupune că serverul şi clientul au ceasuri Time Protocol.

Slăbiciunile lui Kerberos

- • Un atac foarte spectaculos este următorul: un atacator înregistrează o serie de mesaje de la un client care ştie că reprezintă o tranzacţie importantă. Peste o săptămînă atacatorul infiltrează în reţea mesaje false NTP prin care setează ceasul unui server cu o săptămînă în urmă. După asta atacatorul retransmite mesajele capturate, care vor fi re-executate, pentru că serverului îi par proaspete.
- • Asta face securitatea în calculatoare o problemă foarte spinoasă: adesea protocoalele propuse sunt eronate, dar nu există nici o metodă formală pentru a depista şi verifica asta. Chiar dacă un protocol este corect formal, se poate baza pe presupuneri nerezonabile asupra mediului în care operează, cum ar fi ceasurile sincronizate. Şi chiar dacă se bazează pe presupuneri rezonabile, implementarea scrisă de un programator uman poate să fie vulnerabilă.

Kerberos pentru un utilizator

- • Domeniul administrativ de lucru este complet kerberizat. Programul de login a fost modificat ca imediat ce se tasteaza parola să converseze cu AS şi să obţină un tichet pentru serverul care dă tichete, TGS. Directorul gazda se afla pe un server de fis.pe un disc din reţea, aflat la distanta. Atunci cînd se comunica prima oară cu serverul de fis. demonul local de pe maşina pe care s-a loginat clientul, acesta foloseşte tichetul obţinut de la AS pentru a obţine de la TGS un tichet pentru serverul de fis.. După aceea se autentifică serverului de fis., exact ca în schema de mai sus.
- • După autentificarea cu serverul de fis. toată comunicaţia între maşina client şi serverul de fis. se va face folosind un alt protocol, numit Secure RPC (Remote Procedure Call): apel sigur de procedură la distanţă, care foloseşte iniţial cheia de sesiune oferită de TGS.

Kerberos pentru un utilizator

- Pentru utilizatorul final totul este aproape complet transparent.
- Singura neplăcere este că la fiecare 25 de ore tichetele pentru TGS expiră, şi atunci trebuie să tasteze din nou parola pentru a obţine tichete noi.
- Utilizatorul poate vedea în orice clipă tichetele pe care le poseda cu comanda klist:

```
$ klist
Ticket file: /tkt/7108-0401-35ae552f
Principal:   me@CS.CMU.EDU
Issued           Expires          Principal
Jul 20 10:50:34 Jul 21 12:16:55 krbtgt.CS.CMU.EDU@CS.CMU.EDU
Jul 20 10:50:34 Jul 21 12:16:55 afs@CS.CMU.EDU
Jul 20 10:51:30 Jul 21 12:17:51 zephyr.zephyr@CS.CMU.EDU
```

Kerberos pentru un utilizator

- • Tichetele sunt ţinute în fişierul /tkt/7108-0401-35ae552f pe discul local.
- • Utilizatorul poseda 3 tichete: unul pentru TGS, unul pentru serverul de fis. şi unul pentru sistemul de mesagerie zephyr (care nu ne interesează prea tare acum).

Kerberos pentru un utilizator

- Mai sunt dispoziţie următoarele comenzi:
- **kinit** prin care se poate schimba identit. Kerberos (de pildă dacă un utiliz. vrea să lucreze pe calculatorul altui utiliz. va tasta kinit numele-lui şi apoi parola lui personală;
- **kdestroy** prin care toate tichetele de pe maşina locală sunt distruse;
- foarte utila dacă un utiliz. nu doreste ca un altul să lucreze în numele sau;
- **kpasswd** prin care se poate schimba cheia (parola) stocată pe AS.
- Schimbarea parolei este sigură, pentru că parola va fi trimisă criptat la un server special care face managementul parolelor şi modifică baza de date
- din care citeşte AS. Autentificarea la managerul de parole se face tot
- folosind Kerberos. Asta înseamnă că odată ce utiliz. are o parolă Kerberos (care trebuie introdusă manual în baza de date a lui AS) schimbarea o poate face fără să se mai adreseze la vreun administrator;

Kerberos pentru un utilizator

- • **kdb_init** este o comandă folosită de administrator pentru a crea o nouă
- bază de date Kerberos atunci cînd porneşte serviciul;
- • **kdb_admin** este comanda prin care administratorul adaugă un nou utilizator în baza de date;
- • **kdb_edit** este comanda prin care se pot adăuga noi administratori în baza de date AS: persoane care au dreptul să modifice baza de date.
- • În plus, majoritatea comenzilor care operează în reţea au fost kerberizate.
- De pildă demonul şi clientul de telnet (terminal virtual): cînd se face telnet pe o maşină la distanţă clientul cere parola după care obţine un tichet de la
- TGS pentru demonul telnet de pe maşina de la distanţă; în acest fel parola
- nu circulă niciodată prin reţea (cum ar fi fost cazul dacă telnet nu era kerberizat).